Catallaxy-based Grid markets

Torsten Eymann[a,*], Michael Reinicke[a], Werner Streitberger[a], Omer Rana[b], Liviu Joita[b], Dirk Neumann[c], Björn Schnizler[c], Daniel Veit[c], Oscar Ardaiz[d], Pablo Chacin[d], Isaac Chao[d], Felix Freitag[d], Leandro Navarro[d], Michele Catalano[e], Mauro Gallegati[e], Gianfranco Giulioni[e], Ruben Carvajal Schiaffino[f] and Floriano Zini[f]

[a]*Chair of Information Systems, University of Bayreuth, 95440 Bayreuth, Germany*

[b]*School of Computer Science and the Welsh eScience Centre, University of Cardiff, CF24 3AA, UK*

[c]*Information Management and Systems, University of Karlsruhe, 76131 Karlsruhe, Germany*

[d]*Department of Computer Architecture, Technical University of Catalonia, Jordi Girona, 1-3, 08034 Barcelona, Spain*

[e]*Department of Economics, Universita Politecnica delle Marche, Piazzale Martelli 8, 60121 Ancona, Italy*

[f]*Automated Reasoning Systems Division, ITC-irst Trento, Via Sommarive, 18, 38050 Povo/Trento, Italy*

Abstract. Grid computing has recently become an important paradigm for managing computationally demanding applications, composed of a collection of services. The dynamic discovery of services, and the selection of a particular service instance providing the best value out of the discovered alternatives, poses a complex multi-attribute n:m allocation decision problem, which is often solved using a centralized resource broker. To manage complexity, this article proposes a two-layer architecture for service discovery in such Application Layer Networks (ALN). The first layer consists of a service market in which complex services are translated to a set of basic services, which are distinguished by price and availability. The second layer provides an allocation of services to appropriate resources in order to enact the specified services. This framework comprises the foundations for a later comparison of centralized and decentralized market mechanisms for allocation of services and resources in ALNs and Grids.

Keywords: Grid computing, market mechanisms, agent-enabled service oriented architecture

1. Introduction

This article presents an investigation in implementing an electronic Grid market based on the "Catallaxy" concept of F. A. von Hayek. Catallaxy describes a "free market" economic self-organization approach for brokering electronic services, which can be implemented for realizing resource allocation in Application Layer Networks (ALNs). The term ALN comprises network concepts, such as Grid and Peer-2-Peer (P2P) systems, which overlay the existing physical Internet topology. In ALNs, participants offer and request application services and computing resources of different complexity and value – creating interdependent markets.

In this article, the complex interdependencies are broken down into two types of interrelated markets: (1) a service market – which involves trading of application services, and (2) a resource market - which involves trading of computational and data resources, such as processors, memory, etc. This distinction between resource and 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

*Corresponding author: Prof. Dr. Torsten Eymann, Chair of Information Systems, University of Bayreuth, Universitatsstrasse 30, 95440, Bayreuth, Germany. Tel.: +49 921 55-2807; Fax: +49 921 55-2216; E-mail: eymann@uni-bayreuth.de.

based on the particular resource capabilities that are being made available by some hosting environment.

In such interrelated markets, allocating resources and services on one market inevitably influences the outcome on the other markets. A common approach of many other Grid market concepts is to allocate resources and services by relying on the presence of centralized resource/service brokers. However, the complex reality could turn such approaches useless, as the underlying problem is computationally demanding and the number of participants in a worldwide ALN can be huge.

The research question taken up in this article is to develop a Grid realization of an economic concept, which describes the ability to trade (electronic) services in a decentralized fashion, a free-market economy to adjudicate and satisfy the needs of participants who are self-organised and follow their own interest. The Catallaxy concept is a coordination approach for systems consisting of such autonomous decentralized agents, and is based on constant negotiation and price signalling between agents [3]. Every individual (agent) assigns a value to service access information, and by exchanging bids for service access, the price signals carry information between individuals (agents) about the knowledge of others [5]. This exchange of information applies even across markets, as changing availability on the resource market will be reflected by cascading price changes for those basic services which rely on the same resources. Hayek called this feature a “tele-communication” system in its literal sense. The huge size of Grids to be controlled, and the availability of software agent technology, makes implementing Hayek’s Catallaxy an alternative to a centralized allocation approach, using the ensuing “spontaneous order” as a concrete proposal for both the design and coordination of information systems. The resulting multiagent system will be highly dynamic, thereby leading to Grid networks which behave in a P2P fashion.

The term P2P should be interpreted not as a 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 [11]. P2P systems exhibit a set of characteristics that are relevant from the architectural point of view [17]:

– 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 applications can differ in many properties such as resource characteristics, performance or 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 a priori 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.

2. Principles of the Catallaxy

Friedrich August von Hayek [6] and other Neo-Austrian economists understood markets as decentralized coordination mechanisms, as opposed to a centralized command economy control. In addition to macroeconomic thoughts, Hayek’s work also provides concrete insight into the working mechanisms of economic coordination. 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 [18]. 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 enumerated below. Each property 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, as nobody has 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.

Hayek's Catallaxy concept is the result of descriptive, qualitative research about economic decision-making of human participants. In the following section, its results are taken literally to construct ALN markets with software participants, who reason about economic decisions using artificial intelligence.

3. Prototyping the Catallaxy

This section will pick up the requirements of the Catallaxy described above and will present fundamental components to satisfy these requests in ALNs. Starting with a decomposition of the application scenario into two distinctive markets, functionality and components needed are identified. Subsequently, a middleware architecture and a corresponding application scenario are presented.

3.1. A two layer ALN of services and resources

ALNs encompass heterogeneous resources, computational and data services in different administrative domains, which are logically coupled. ALNs will depend on basic services that are dynamically combined to form value-added complex services [8]. For their enactment, 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.

The environment is thus divided into two layers, the application/service layer and the resource layer. These two layers contain 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 also provide an interface to access computational resources for complex services. In both layers, the participants have varying objectives which change dynamically and unpredictably over time.

3.2. Market model

Current Grid Computing architectures exhibit a fairly static resource infrastructure, connected by physically stable links. The shift to a pervasive, ubiquitously accessible Grid demands for a more dynamic consideration of resources and connections. Figure 1 shows a perspective on a two-layered Grid Market, encompassing a distinct service and a resource market.

A complex service is represented by a proxy who needs (remote) basic service capabilities for execution – with support for a service selector instance. Complex services are therefore shielded from details of the resource layer implementation.

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). 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-tuple of resource types (e.g. CPU, storage, and bandwidth); co-allocation describes obtaining resources for one single service transaction from various local resource man-

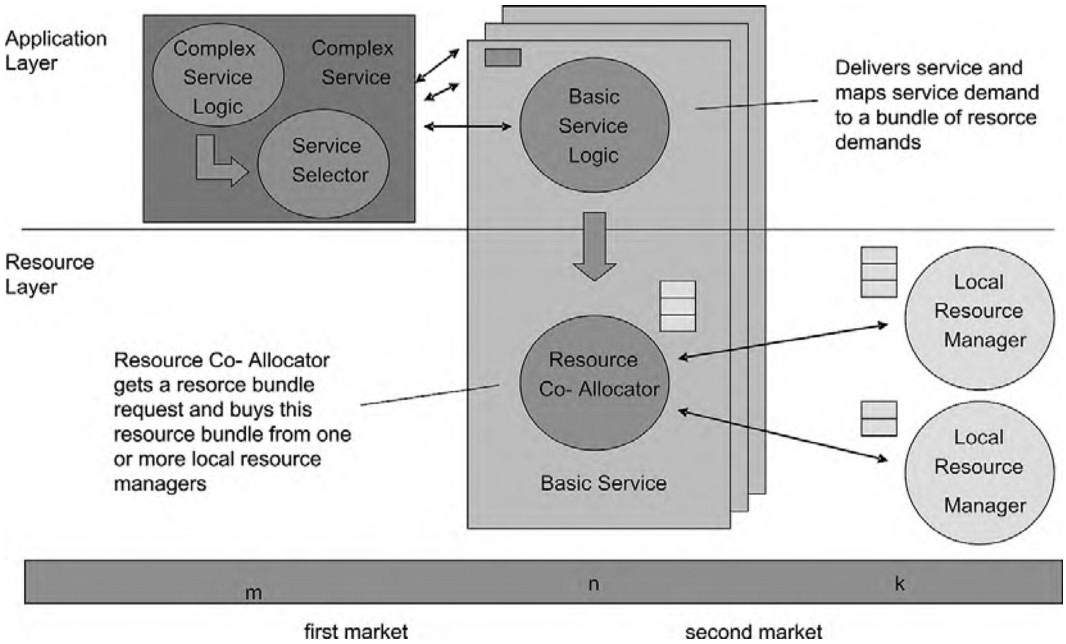

Fig. 1. Catallaxy-based Grid market model.

agers simultaneously. It is expected that a local resource manager hides all details of the local 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:k market: n basic service copies can bargain with k resource services. This takes dynamic resources into account. Resources, like basic services, can fail and are subject to maintenance and inspection procedures.

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 the resource layer and market). The following scenarios exist:

1. *Contracting resources in advance:* requires a forecast of future demand [2]. 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 failure.
2. *Contracting resources after closing the service contract:* this might lead to insufficient resource offers on the resource market and thus to non-accomplishable 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 thus forms the basis of the Catallaxy-based Grid market. For service execution the basic service logic requests a resource bundle. The further process of contracting/allocating the resource is done by the resource co-allocator. The selection of a resource bundle is analogous to the selection of a service, with the exception that a bundle is requested, whereas on the service market only one service can be negotiated per request.

The local resource managers offer resource bundles. The resource bundle could be a tuple consisting of resources 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. The negotiation is also initiated by the resource co-allocator.

4. Implementation in a MAS

In the Catallaxy approach, every player in the market is modelled as a software agent. The following sections present first the lifecycle of agents and their components, afterwards their integration into a middleware.

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 Fig. 2). The lifecycle starts with an initialisation phase – consisting of information and matching sub-phases (cf. [19]).

- During 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 pairs 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 about 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 multiattribute 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 the seller agent delivers, on demand.
- The buyer-seller relationship is analyzed in the evaluation phase. Information about evaluation of other participants can be exchanged. In a closed system or with all trustworthy participants, the evaluation phase can be omitted and the process begins again.

4.2. Components for realizing Catallaxy

For 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 adaptation mechanisms (machine learning), to react to changes in the environment and to implement a method that adapts to the behaviour of the surrounding agents.

4.2.1. 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.

The simplest decentralized search method is using an unstructured flooding mechanism like in Gnutella [13]. 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 search algorithms promise a guaranteed item discovery and a reduced message count. The usage of distributed hash tables (DHTs) in Chord [1] and Tapestry [21] offers a guaranteed search, and distributes the search process to the connected nodes in the network. The search does not rely on 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, the implementation of a simple flooding algorithm is regarded as a suitable approach.

4.2.2. 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 to 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 the one criteria "price", for example when delivery time affects the buyer's usage and therefore justifies a change of price.

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 [9,12]. In integrative negotiations, participants exchange information about objectives and priorities to seek a common solution. This concept is recommended 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

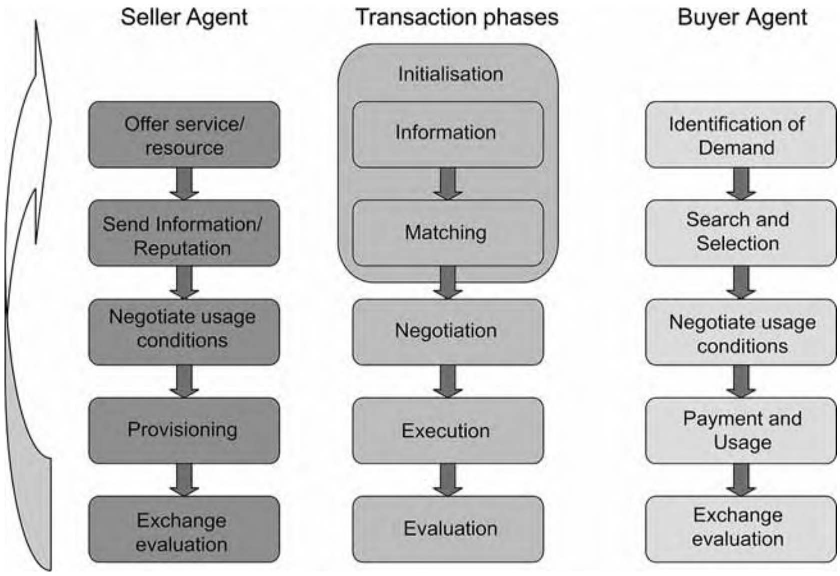

Fig. 2. Lifecycle of agents.

closer in their expectations in every negotiation round. Distributive negotiations are marked by existence of a common utility space [12] 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 system remains constant.

The goal is a system wide pareto-optimum that can be defined as an acceptable doctrine of general goodness [14]: 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. Sandholm [15] 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 a budget increase per transaction and per period, sales volume or other metrics taken from economics.

4.2.3. 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 [4].

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

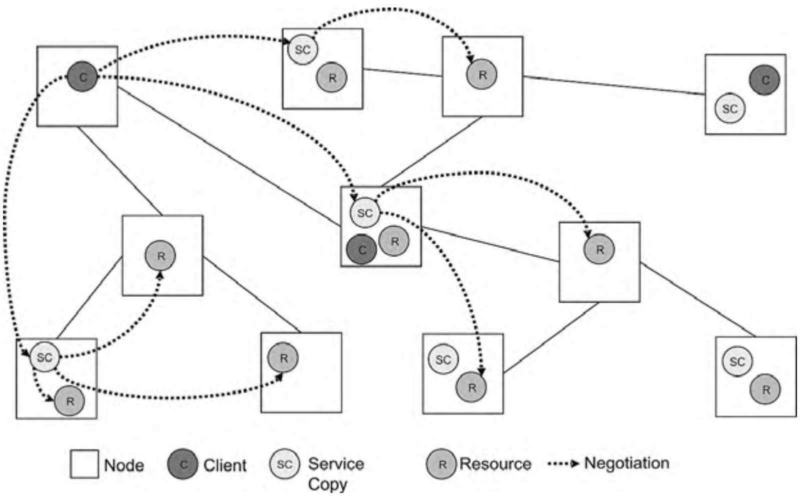

Fig. 3. Catallactic middleware as a network of agents.

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 Fig. 3.

In principle, this middleware would be applicable to all P2P network architectures. Instead of implementing Catallactic agents responsible for both the self-organization of the system and the management of the negotiation process, a layered architecture is proposed (see Fig. 4). In this architecture, economic agents are responsible for implementing high level behaviour (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 self-organization of the system, and the interaction with the base platform that ultimately manages the resources being traded.

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) to specific application domains.

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

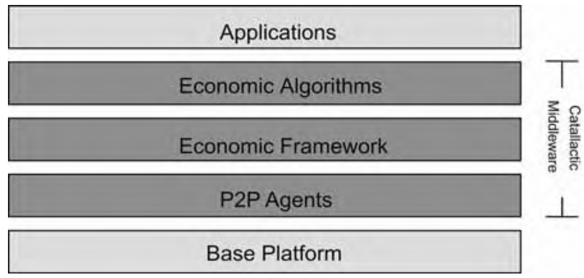

Fig. 4. Layered agent architecture.

– How to pass the description of the service requirements along with the desired conditions (preferences) 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 behaviour 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 suggest participating agents a certain level of confidence about the fulfillment of agreed conditions and service level agreements. Traditional approaches, using cen-

tralized policies require complete state information which often is not available in dynamic and complex networks [10].

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 overall architecture and offers a set of generic mechanisms, in addition to which specialized mechanisms can be dynamically plugged in 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 behaviour 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. Implementation of the application scenario

Figure 5 shows a scenario demonstrating the use of the Catallaxy approach. The figure demonstrates the use of various components identified in Section 3 for sharing database content, plus some application-specific modules which modify the division of work between objects, without altering the concept. This 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 are responsible for establishing and maintaining a P2P communications infrastructure. They are 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.

Site Monitors act as a rendezvous point in a P2P topology – essentially supporting the caching of messages that are propagated in the network. 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 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. Engineering the market scenario

The decentralized negotiation protocols following the Catallaxy paradigm will be compared with centralized auction protocols. Two extreme scenarios serve as benchmarks. One scenario is characterized by standardized commodities, whereas the other allows for highly heterogeneous goods.

For a reasonable benchmark in the centralized approach, one has to find adequate auction protocols for both scenarios. Unfortunately, the environment and the underlying auction protocol exert crucial effects on the outcome. For instance, in a sealed bid auction the bidders simultaneously submit bids to the auctioneer without knowledge of the 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 behaviour of the market participants and could therefore lead to different outcomes. As such, designing a market mechanism that achieves a desired outcome is extremely difficult, because it entails the anticipation of agent behaviour.

In order to approach this task, a systematic market engineering approach guides the design of tailored market mechanisms by providing a structured and theoretically profound procedure [20]. The approach provides a process model which is divided into four stages: In the

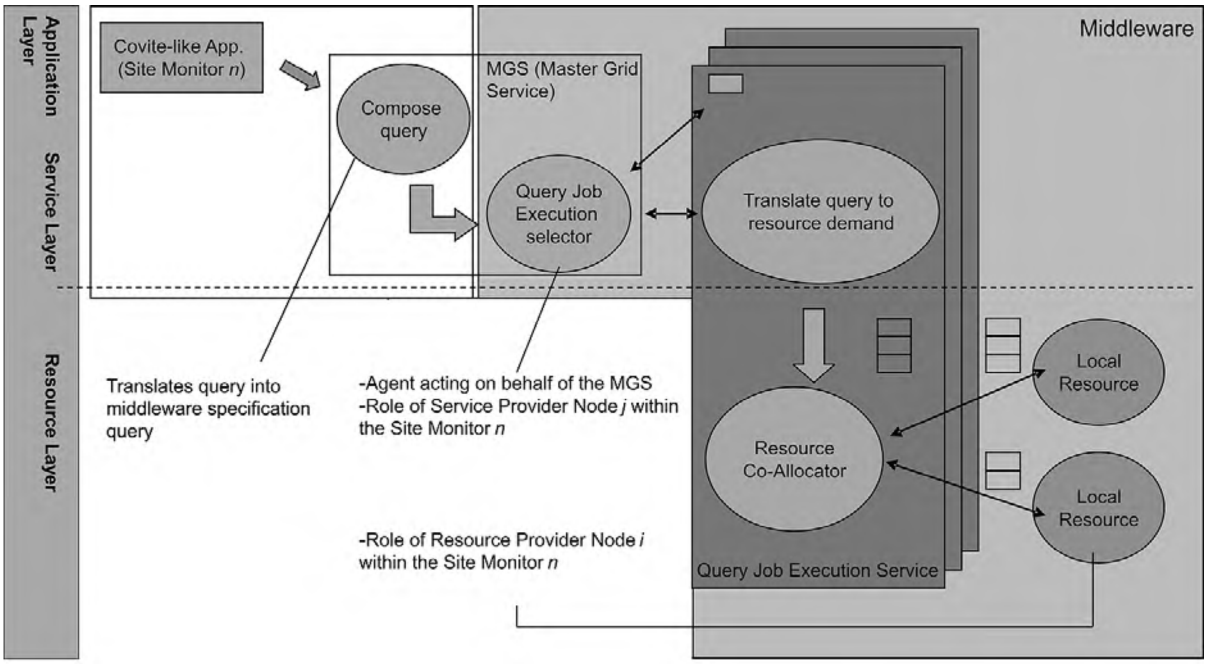

Fig. 5. The proposed multiagent system architecture for indexing and querying.

first stage – the environmental analysis – the requirements of the new market mechanism (i.e. which 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 deployed 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. While the market engineering approach has originally been invented for designing auction markets, many of its findings also apply for bargaining markets, especially the environmental analysis. The main difference lies in the second stage, the design of the allocation mechanism.

5.1. *A mechanism 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. Basically, buyers and sellers are services, which require other auxiliary services; 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 for the centralized market and serves as a comparable mechanism for the decentralized negotiation schema.

5.2. *A mechanism 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 30 GB or 200 GB of space. Requirements for the resource market are [16]:

– *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. This is based on the fact that computer resources (e.g. in the computational Grid) are complementarities. Complementarities are resources with superadditive valuations ($v(A)+v(B)\leqslant v(AB)$), as the sum of the valuations for the single resources is less than the valuation for the whole bundle. .
– *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 centralized resource market. Instead, there is a need for a multi-attribute combinatorial exchange that satisfies the described requirements. When comparing to the service market, the challenge for the bargaining mechanism are the high number of messages needed to establish a bundle trade – this is going to be part of the evaluation, whether the higher communication overhead will outperform the lacking scalability of the centralized mechanism.

6. Evaluation and conclusion

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

The comparison of both approaches is on one hand supported by applying a structured market engineering approach to both market designs. On the other hand, 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 accompanied by reference and application scenarios.

Future work 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 and the allocation efficiency under constraints of the number of participating entities. As an acceptable system-wide performance matrix is impossible to define, an economics-based paradigm for the management or resource allocation and orchestration will be used.

Acknowledgments

This work has partially been funded by the EU in the IST programme "Future and Emerging Technologies" under grant FP6-003769 "CATNETS".

References

[1] I. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris and I. Stoica, Looking up data in P2P systems, *Communications of the ACM* **46**(2) (Feb 2003), 43–48.

[2] R. Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing,* Ph.D. Thesis, Monash University, Melbourne, Australia 2002. http://www.buyya.com/thesis/thesis.pdf, last access 2005-10-01.

[3] T. Eymann, S. Sackmann, G. Müller and I. Pippow, *Hayek's Catallaxy: A Forward-looking Concept for Information Systems,* Proc. of American Conference on Information Systems (AMCIS'03), Tampa, FL, 2003. http://tserv.iig.uni-freiburg.de/~sackmann/publications/Hayeks%20Catallaxy.pdf, last access 2005-10-01.

[4] D.P. Friedman, *The Double Auction Market. Santa Fe Institute studies in sciences of complexity,* Addison-Wesley, Reading, MA, 1993.

[5] F.A.v. Hayek, The Use of Knowledge in Society, *American Economic Review* **XXXV**(4) (1945), 519–530.

[6] F.A.v. Hayek, W.W. Bartley, P.G. Klein and B. Caldwell, *The collected works of F.A. Hayek,* University of Chicago Press, Chicago, 1989.

[7] E. Hoppmann, Unwissenheit, Wirtschaftsordnung und Staatsgewalt, in: *Freiheit, Wettbewerb und Wirtschaftsordnung,* V. Vanberg, ed., Haufe Verlag, Freiburg, 1999, pp. 135–169.

[8] M.N. Huhns and M.P. Singh, Service-oriented computing: Key concepts and principles, *Ieee Internet Computing* **9**(1) (Jan–Feb 2005), 75–81.

[9] N.R. Jennings, P. Faratin, A.R. Lomuscio, C. Sierra and M.J. Wooldridge, Automated negotiation: prospects, methods and challenges, *International Journal of Group Decision and Negotiation* **10**(2) (2001), 199–215.

[10] K. Krauter, R. Buyya and M. Maheswaran, A taxonomy and survey of grid resource management systems for distributed computing, *Software-Practice & Experience* **32**(2) (Feb 2002), 135–164.

[11] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins and Z. Xu, Peer-to-Peer Computing; Hewlett Packard Labs, Palo Alto HPL-2002-57, 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html, last access 2005-10-01.

[12] D.G. Pruitt, *Negotiation behavior,* Academic Press, New York, 1981.

[13] M. Ripeanu, A. Iamnitchi and I. Foster, Mapping the Gnutella network, *Ieee Internet Computing* **6**(1) (Jan–Feb 2002), 50–57.

[14] J.S. Rosenschein and G. Zlotkin, *Rules of encounter – designing conventions for automated negotiation among computers,* MIT Press, Cambridge, 1994.

[15] T.W. Sandholm, *Negotiation Among Self-Interested Computationally Limited Agents,* Ph.D. Thesis, University of Massachusetts, Amherst, 1996.

[16] B. Schnizler, D. Neumann and C. Weinhardt, *Resource Allocation in Computational Grids – A Market Engineering Approach,* Proc. of The 3rd Workshop on e-Business (WeB), Washington D.C., US, 2004. http://www.iw.uni-karlsruhe.de/catnets/fileadmin/publications/SchnizlerNeumannWeinhardt_04_ResourceAllocation.pdf, last access.

[17] P. Sewell, J. Leifer, U. Nestmann, A. Serjantov and K. Wansbrough, *Required foundations for peer-to-peer systems,* PEPITO Project IST-2001-33234, Cambridge, UK D1.1, 01.06.2003 2003. http://www.sics.se/pepito/D1.1/d11.pdf, last access.

[18] H.A. Simon, *Models of Man – Social and Rational,* John Wiley & Sons, New York, 1957.

[19] D. Veit, *Matchmaking in electronic markets – An agent-based approach towards matchmaking in electronic negotiations,* vol. 2882, Springer, Berlin, 2003.

[20] C. Weinhardt, C. Holtmann and D. Neumann, Market Engineering, *Wirtschaftsinformatik* **45**(6) (2003), 635–640.

[21] B.Y. Zhao, J.D. Kubiatowicz and A.D. Joseph, Tapestry: A fault-tolerant wide-area application infrastructure, *Computer Communication Review* **32**(1) (Jan 2002), 81.

Authors' Bios

Torsten Eymann is a full professor of Information Systems at University of Bayreuth since 2004.

Oscar Ardaiz is full time adjunct lecturer at the University of Navarra.

Michele Catalano is a PhD student at the University Politecnica delle Marche since 2003.

Pablo Chacin is a PhD student at Technical University of Catalonia since 2004.

Isaac Chao is a PhD student at Technical University of Catalonia since 2003.

Felix Freitag is full time adjunct lecturer at the Computer Architecture Department of the Technical University of Catalonia.

Mauro Gallegati is a full professor at the University Politecnica delle Marche since 2001.

Gianfranco Giulioni is Lecturer at the University Politecnica delle Marche working on agent based models since 2000.

Liviu Joita is a Research Associate at Cardiff University, School of Computer Science, and Welsh e-Science Centre, as well as a PhD student at Cardiff University since 2002.

Leandro Navarro is associate professor at the Computer Architecture Department of the Technical University of Catalonia.

Dirk G. Neumann is an assistant professor at the University of Karlsruhe, Department of Economics and Business Engineering, Chair for Information Management and Systems, since 2004.

Omer F. Rana is a Senior Lecturer in the Department of Computer Science at Cardiff University, and the Deputy Director of the Welsh eScience Centre.

Michael Reinicke is a PhD student at University of Bayreuth since 2004.

Ruben Carvajal-Schiaffino was a researcher at ITC-irst between 2002 and 2005, currently he is an assistant professor at the Universidad de Santiago de Chile.

Björn Schnizler is a PhD student at University of Karlsruhe since 2004.

Werner Streitberger is a PhD student at University of Bayreuth since 2004.

Daniel Veit works since April 2003 as assistant professor at the University of Karlsruhe, Department of Economics and Business Engineering, Chair for Information Management and Systems.

Floriano Zini is a researcher at the Centre for Scientific and Technological Research (ITC-irst) in Trento, Italy, since 2000.