Co-Evolution of Bargaining Strategies in a Decentralized Multi-Agent System

Torsten Eymann

Albert-Ludwigs-Universität Freiburg
Institute for Computer Science and Social Studies, Telematics Dept.
Friedrichstrasse 50, 79098 Freiburg im Breisgau, Germany
eymann@iig.uni-freiburg.de

Abstract

Software agents will personalize smart devices to act autonomously on behalf of their human owner. In dynamical electronic commerce environments, these agents could buy and sell goods and services from each other, without using a central market maker. From a system perspective, this creates a decentralized and continuously changing multi-agent system with the need for coordination of supply and demand. In this article we show how such a multi-agent system may be decentrally coordinated while software agents bargain with each other under the constraints of incomplete information, non-equilibrium and time pressure. The agents adapt to a changing environment with an evolutionary learning mechanism. It can be shown that the multi-agent system as a whole shows emergent coordination in absence of a centralized coordination institution.

Software Agents on Smart Devices

The discussion about using small, mobile devices in the context of *ubiquitous computing* (Weiser 1991) or *pervasive computing* seems to be largely hardware-centric. The software components are mostly considered to passively collect information from the environment or centralized web servers and store it in the device's *data shadow* somewhere in the Internet.

Small, passive devices with cumbersome human-computer interfaces and a constant need for interaction in ongoing transactions will soon create a demand for autonomously and proactive software components on these devices. Such software agents (Wooldridge 1999)can assist human buyers and sellers in digital business processes and environments, which are characterized by trading and money transactions, to save transaction costs. *Digital Business Agents* (DBAs) (Diebold Consulting (Ed.) 2001)will monitor other agents and the environment continuously, e.g. by making price comparisons between different suppliers in the on- and offline world (Youll et al. 2000), in order to fulfill their design goal if utility maximization for their human owner. They will be able to enter into negotiation with many potential trade partners at once, reaching an acceptable deal and setting up a contract in a matter of milliseconds (Preist 1998). In that case, the DBAs may execute a complete business transaction as *silent commerce* (Schenker 2000) in the background without the need for human intervention. Many, if not most of these transactions will occur locally and decentralized, directly between the devices.

A prerequisite is that the human principals of the agents are able to define economic goals, preferences and strategies in computer processable data structures (Kraus 1997). A goal is an abstract representation of a bidder's demand, e.g. a product identification with a price to match. Preferences can be classified as either hard constraints, which have to be matched unconditionally, or soft constraints, which can be ranked, e.g. a preference for certain colors (Guttman and Maes 1998). Strategies to reach the goals and to negotiate on maximal meeting of the preferences can either be rule-based/argumentative, game-theoretic or heuristic/adaptive (Kraus 1997; Lomuscio, Wooldridge, and Jennings 2000)

The use of DBAs is not confined to the consumers, however. Similar software can also be used by producers and retailers. As the consumers use comparison shoppers that allow automated price comparisons between different suppliers (Doorenbos, Etzioni, and Weld 1997; Youll et al. 2000), sellers are expected to begin to dynamically post prices to negotiate with individual buyers (Kephart, Hanson, and Greenwald 2000).

The research question pursued in this article is how different negotiation strategies of DBAs compete against each other in an unregulated environment, and the impact on the coordination of the environment as a whole. In the next section of chapter 2, we describe the technical requirements and the setup of a multi-agent system (MAS) that coordinates a model supply chain process using decentralized economic coordination. Some experiments with different bargaining strategies in the DBAs are described in chapter 3. The article concludes in chapter 4 with an outlook into possible application areas and the problems of linking economic concepts and technical realization of market coordination.

An Experimental Market Coordination Environment

In order to evaluate the future impact and functionality of agent-driven coordination issues, a prototypical MAS called AVALANCHE has been designed. Here, DBAs represent human users intending to buy, produce and sell different goods or services as "miniature automated businesses" (Kephart, Hanson, and Greenwald 2000). A simple and arbitrary supply chain is modeled in which three different types of DBAs produce a consumer good: "lumberjacks", that buy trees to produce and sell boards; "carpenters", that buy the boards and fix them together to sell as a panel; and "cabinetmakers", that buy the panels, build tables out of them and sell them. Two additional types, producers and consumers, which only define a seller or buyer strategy, respectively, provide the tight ends of the supply chain. This scenario is not considered to be a realistic model of a supply chain, but serves as a blueprint for a specific class of typical information system application environments.

The Technical Setup

AVALANCHE is realized in JAVA 1.3 and uses the ORB class library VOYAGER 3.0 (Objectspace Inc. 1999). All agents are independent JAVA threads. The system architecture consists of three basic classes, the marketplaces, trader agents, and one experiment control object. Both marketplaces and trader agents are initialized simultaneously in our experiments, but due to the openness of the architecture, trader agents could enter marketplaces at any point in time. Detailed information about AVALANCHE can be found in (Eymann 2000).

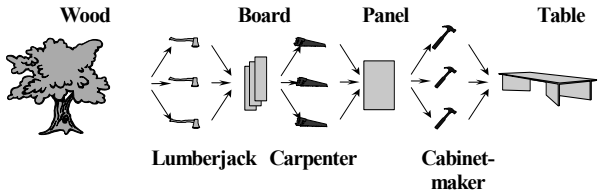

Figure 1: A simple supply chain scenario.

The agent class defines communication abilities and negotiation protocols (in the remainder of this paper, we use the terms "agent", "software agent" and "DBA" synonymously). Each agent communicates with every other object on the marketplace in direct, bilateral and unmediated fashion. The communication channel may be secured using an asymmetric public key infrastructure for encryption and digital signatures, which can be neglected for the purpose of this article. All agents are based on standardized concepts of JAVA, and distinguishable by a unique identity.

The electronic marketplaces are based on Java Virtual Machines (JVM), which run as a server process in a TCP/IP-based network and are identifiable by a unique IP port address. In order not to interfere with the negotiation process of the software agents, the functions and services of the object are kept to a minimum. The marketplace neither actively influences the software agents' strategies nor the communication – in particular, it does not explicitly synchronize or schedule the agents' activities. It only offers a single "white pages" directory service for any software agent to register as seller or buyer, and to inquire about other registered agents. Price bids are never publicly posted (as in catalogs), since the price information might partly reveal the agent's strategy to competitors and bidders, and prevents the negotiation of individual prices per bidder. The location may provide additional information about neighboring marketplaces for mobile agents, auction and notary functions or a reputation tracking mechanism as in (Padovan et al. 2001), but this is not realized in the version described here.

The experiment control object contains technical functions for logging the transaction data, which is not expected to have any effect on the market coordination.

The Negotiation Protocol

The key difference of AVALANCHE compared to many other agent-based marketplace projects is the absence of a central arbitrator or auctioneer. We are instead researching into self-organizing, emergent coordination, achieved through the implementation of a strictly decentralized automated negotiation (bargaining) protocol.

The goods that are traded are defined as commodities, so the only negotiation variable is the price. All agents in AVALANCHE follow the economic goal of profit maximization. They try to buy input goods for less and to sell output goods for more. Depending on the actual market situation, its equity, and stock, the agent decides autonomously which action to take next - whether to buy, sell, produce, move or self-terminate.

The agent lifecycle "follows the money": if the agent has finished goods in stock, it tries to sell. If the agent has no goods, but input factors in stock, it simulates the production of output goods (by waiting an appropriate length of time). If the agent has no input factors in stock, it tries to buy some. If the market situation is not satisfying at all, e.g. if there are no offers or demands within a certain time span, the agent tries to move to another marketplace. If the agent has spent its entire budget or all marketplaces are shut down, it has to terminate – every few milliseconds the agent has to pay utilization fees to the market anyway, so doing nothing is never a rewarding strategy.

In the case of buying or selling, the DBA goes through the three stages of a market transaction: information, agreement, and settlement (Lindemann and Schmid 1999). In the information phase of any transaction, a buyer or seller has to identify its potential trading partners. If a seller agent wants to sell its output, it currently advertises its offer in the directory, and waits until a prospective buyer agent demands a negotiation. Simultaneously, the

agent will actively search in the directory for potential transaction partners. Buyer agents apply a mirror procedure.

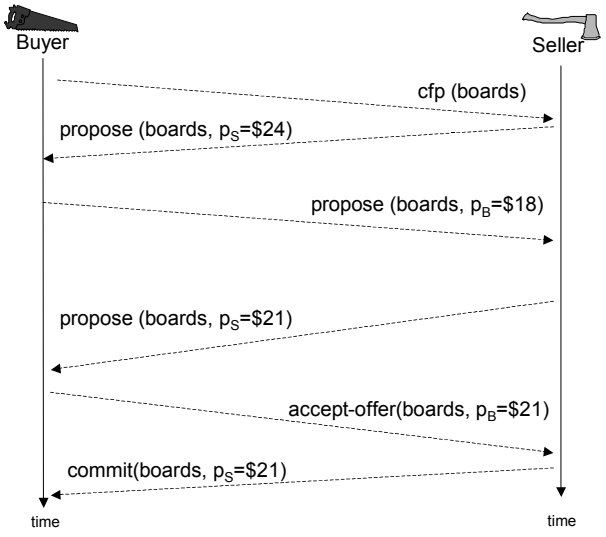

Figure 2: Monotonic Concession Protocol

The buyer agent initiates the agreement phase by communicating with any supplier from the list. Depending on the relation between offer price and internally stored valuation of the goods, the negotiation may immediately terminate. Otherwise, both software agents negotiate using a monotonic concession protocol (Rosenschein and Zlotkin 1994), where propose and counter-propose messages with subsequent price concessions are exchanged. If the negotiation process is successful, both agents will reach a compromise price agreement; otherwise, someone will sooner or later drop out of the negotiation. In this event, the agents will restart with other partners obtained from the directory.

In the final settlement phase, the transaction is carried out and monitored. Sellers and buyers exchange goods and money, respectively. If any party misbehaves in this phase, the resulting reputation information might be used to modify strategies and behavior to be used in future encounters with that agent as in (Padovan et al. 2001).

The Agents' Negotiation Strategy

The goal of the agents, and the reason to engage in negotiation at all, is to maximize the "equity" capital: the agents try to buy materials for a low price and sell their products at high price to other software agents which in turn use these as input goods. To maximize the spread and thus its utility, the agent follows a certain negotiation strategy. Comparable automated negotiation efforts in Multi-Agent Systems can be found in the research context of agent-mediated electronic commerce (Guttman and Maes 1998; Sierra 2000) and market-oriented programming (Wellman 1996). Human negotiation uses parameters such as demand level, concession, and concession rate:

> A bargainer's demand level can be thought of as the level of benefit to the self associated with the current offer or demand. A concession is a change of offer in the supposed direction of the other party's interests that reduces the level of benefit sought. Concession rate is the speed at which demand level declines over time. […] These definitions […are] unproblematic, when only one issue or underlying value is being considered, as in a simple wage or price negotiation (Pruitt 1981).

The human principal chooses the initial values of these parameters as part of the strategy definition during the initialization of the software agent. In the real world, the values are influenced by determinants such as expectations about the other's ultimate demand, position and image loss, limit and level of aspiration and time pressure (Pruitt 1981). Whatever strategy will be used, these parameters will have to be encoded in some way to make it useful.

The AVALANCHE agents implement a heuristic/adaptive strategy based on a stochastic automaton, in which action paths are taken depending on stochastic probes against certain internal parameters. As the strategy of DBAs will be defined by their human owners, the stochastic automaton approach attempts to capture the relevant parameters and sets them in relation to each other. Earlier work in agent-mediated electronic commerce used rule-based (Kreifelt and von Martial 1991) or price-curve models (Chavez and Maes 1996). From an economic aspect, probabilistic models have been experimentally found to outperform simple deterministic models (Erev and Roth 1998). From a technical aspect, the strategy calculation is fast, simple, and easy to define. In AVALANCHE, a combination of six distinct parameters, collectively called the *Genotype*, describes the strategy:

$$G = \begin{pmatrix} p\_acq \\ del\_change \\ del\_jump \\ p\_sat \\ w\_mem \\ p\_rep \end{pmatrix} \quad (1)$$

Acquisitiveness (*p_acq*) defines the probability of maintaining the agent's own last offer. The lower the acquisitiveness value, the higher the average concession rate. Whether an agent concedes in an actual situation is subject to a stochastic probe against this parameter. If RND is a random variable and p_s^t the new offer price of seller *S* at time *t*, the following computation is done:

$$p_s = \begin{Bmatrix} RND < p\_acq : p_s^t = p_s^{t-1} \\ RND \geq p\_acq : p_s^t = p_s^{t-1} - \Delta p \end{Bmatrix} \quad (2)$$

If the agent concedes, the *del_change* parameter calculates the amount of the price concession Δp between two negotiation steps. Seller *S* and buyer *B* calculate a percentage from the price difference of their original offers:

$$\Delta p = (p_S - p_B) \times del\_change \quad (3)$$

To reflect increased knowledge about the market price of one good and simultaneously to maximize income, the agents will set their initial demand level *P* for the start of the negotiation to the last agreement price $p_{B=S}$, modified by the relative value of *del_jump*:

$$P = p_{B=S} \times del\_jump \quad (4)$$

The Satisfaction parameter (*p_sat*) determines if an agent will drop out from an ongoing negotiation. The more steps the negotiation takes, or the more excessive the partner's offers are, the sooner the negotiation will be discontinued. Effectively, this parameter creates time pressure by continuously comparing the market price stored in the *memory* parameter with the actual offer (shown here for the buyer):

$$\begin{Bmatrix} negotiate : p_S > mem \wedge RND > p\_sat \\ reject : p_S > 2mem \\ accept : p_S < mem \end{Bmatrix} \quad (5)$$

All price information received from negotiations computes into a subjective market price for each agent, which modifies the parameter *memory* using a weighted exponential average with weight *w_mem*.

$$mem_t = p \times w_{mem} + mem_{t-1} \times (1 - w_{mem}) \quad (6)$$

Reputation finally *(p_rep*) affects the cooperative behavior of the software agents. In this article, all agents are assumed to cooperate; for other cases, see (Padovan et al. 2001).

The heterogeneity is achieved, if needed, by a uniform random distribution of the initial strategy parameter values, which effectively assigns a different strategy to each agent. The "fitness" denominator for the individual agent's strategy is obviously currency: an agent, which is faster and/or fitter in trading than another will have a relatively higher income. Apart from costs incurred for materials and

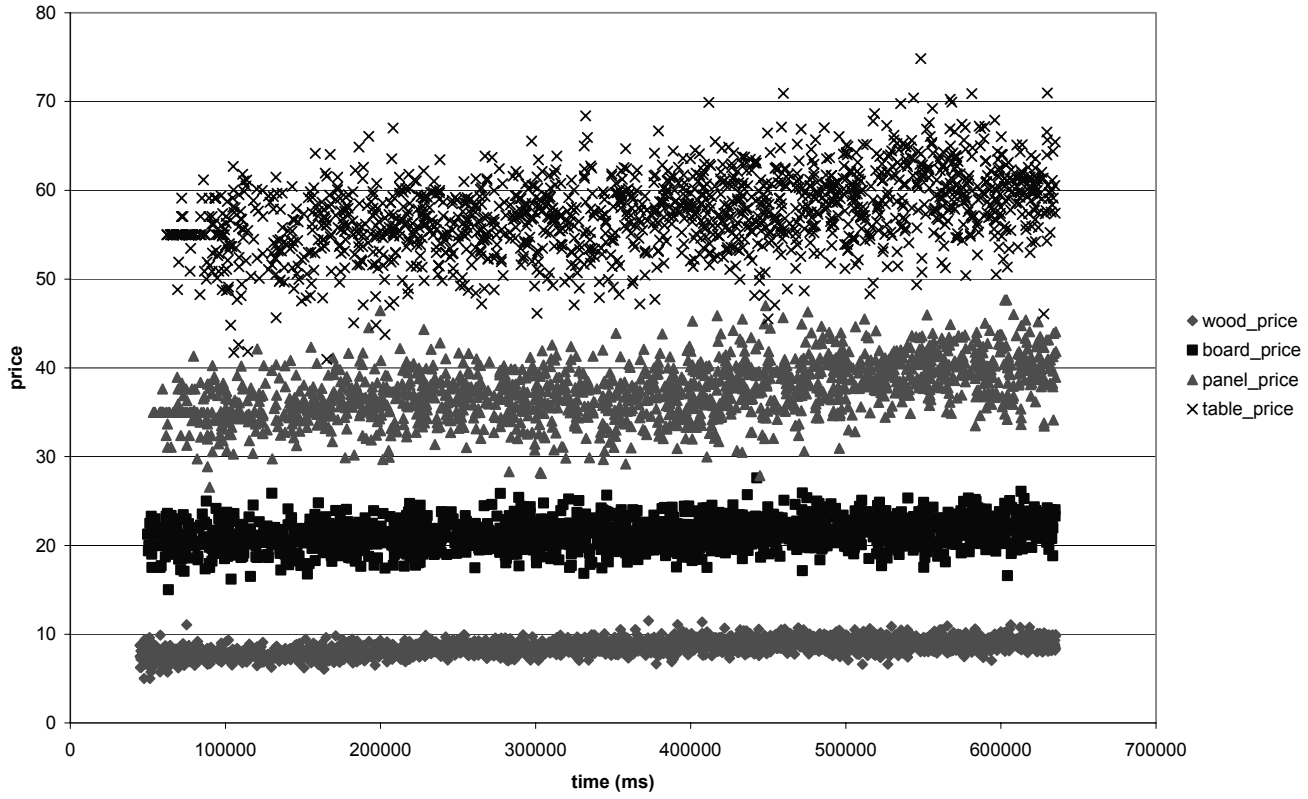

Figure 3: Price dynamics using homogeneous strategies

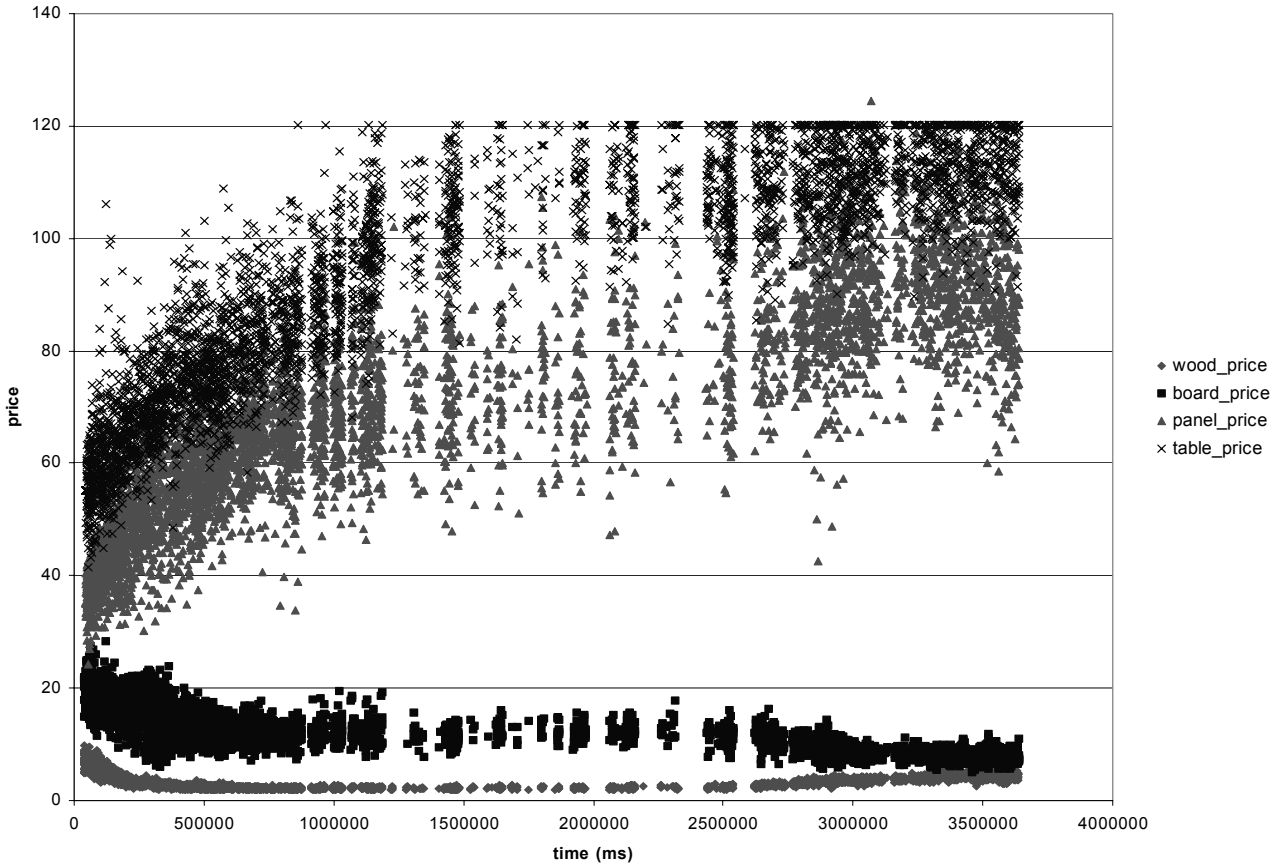

Figure 4: Different concession rates lead to different profits

production, the agents have to pay utilization fees e.g. for resources like computer memory, communication bandwidth and processor time. An agent who does nothing will run out of money after a short time and be discarded from the system.

Experimental Results

To test the behavior of the AVALANCHE system we conduct several test series. Every series consists of several experiment runs, which under similar conditions lead to comparable outcome patterns. In the current implementation we run a generator program, an experimental control object, a single marketplace and 50 trading agents of each type (for a total of 250 agents) in parallel on a Pentium PC under Windows NT for about 10 minutes. The following presentations all show similar patterns when repeatedly starting with the same initial values. However, absolute price levels or points in time have no meaning since they change in every run.

We start with some simple parameter combinations and subsequently move on to more complex experiments. The first experiment shown here has homogenous strategies, which means that all agents are equipped with an identical Genotype and there is no variation of parameter values in the population: p_acq = 0.50, del_change = 0.25, del_jump = 0.15, p_sat = 0.75, w_mem = 0.2, and p_rep = 1. The agents hold no stock, but some initial equity money. Every 10 milliseconds a new "wood" is produced by any producer agent. The consumer agents never run out of money. Initial goods valuations are preprogrammed equally for every agent type (e.g. 25 money units for boards).

Using the simple picture of Figure 3, which holds no great findings for the dynamics of the coordination mechanism, we explain the presentation charts. The horizontal axis measures the time in milliseconds, the vertical axis shows the price level. Every dot in the chart points out time and agreement price of a transaction, e.g. the x-value of a green triangle marks the time when some carpenter software agent has sold a table to a cabinetmaker software agent for y-value price. The type of goods is represented by different colors/gray scales and symbols as indicated by the legend, which shows the supply chain from top to bottom. In the chart, this succession has been turned upside down since woods are sold for the lowest price and tables for the highest. To make the graphic comparable there is a price ceiling for tables at 120 money

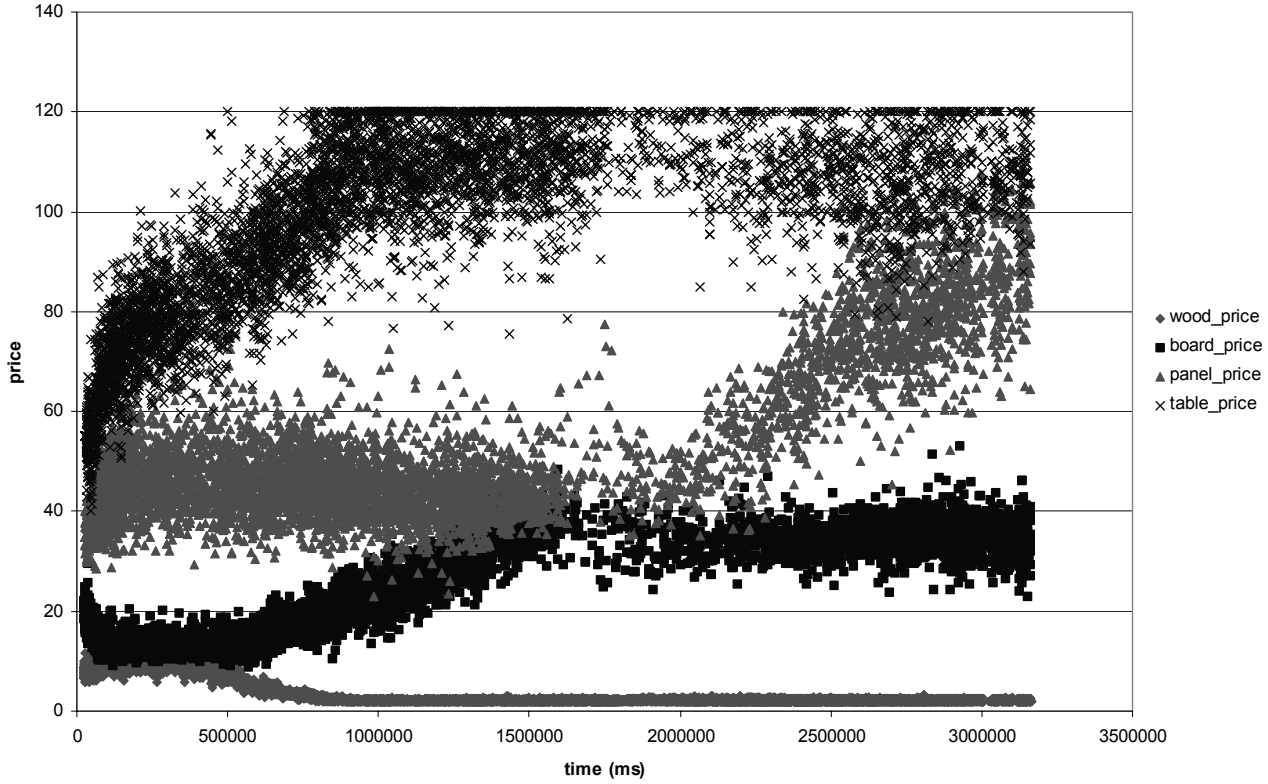

Figure 5: Price Level Development with Evolution

units, which can be seen in later experiments. The time lag at the beginning is caused by software initialization and initial negotiation until the first goods are moved up the supply chain.

In Figure 3 it is not possible for a single agent to gain an advantage, since all agents possess an identical static negotiation strategy. The price levels do not substantially change during the course of the experiment.

In the next experiment (Figure 4), the carpenter agents are initialized with a "greedier" negotiation strategy than all other types. The equipment of only one agent type (carpenters) with p_acq = 0.6, and all others with p_acq = 0.4, leads to a picture of winners and losers (see Figure 5, the vertical white stripes in the figure are experimental artifacts caused by the Windows NT environment and have no effect on the outcome). If the concession rate is lowered (by raising p_acq), this leads to a dramatic decrease of the number of transactions: if the probability of dropping out of a negotiation is greater than the probability of conceding, lesser compromises are reached during the same time span.

Since the other agent types concede faster than the carpenters, the latter are able to increase their income (the spread between board prices and panel prices) at the expense of both lumberjacks and cabinetmakers. It appears to be a dominating strategy to set the agent's acquisitiveness as high as possible, at least higher than any other agent it might trade with, and in fact this is where the gains of the carpenters come from. In an environment without evolutionary learning and open markets, the higher acquisitiveness of the carpenters is a dominating strategy. However, most of these experiments abort early, because the lumberjacks and cabinetmakers run out of money and are thus not able to continue the supply chain – in this case, the victory was short-lived.

Using Evolutionary Learning in the Market

By varying the setting of the Genotype parameters in subsequent experimental runs, we can show how dominant parameter combinations either succeed or are counterbalanced. The negotiation profit gained from a specific Genotype is correlated to the relative setting of other agent's strategies, not the absolute values. "Learning" the right values relative to the other agents is thus essential to gain a higher profit in the future. In Figure 4, the inability of the agents to change their strategy affects both winners and losers – the other agents should raise their own acquisitiveness setting to gain more profit, while the carpenters could lower their setting to conduct more transactions within the same time to raise turnover.

The design goal of AVALANCHE's learning algorithm was to avoid a centralized fitness evaluator with perfect knowledge about each agent's performance, in analogy to the decentralized coordination concept. The chosen

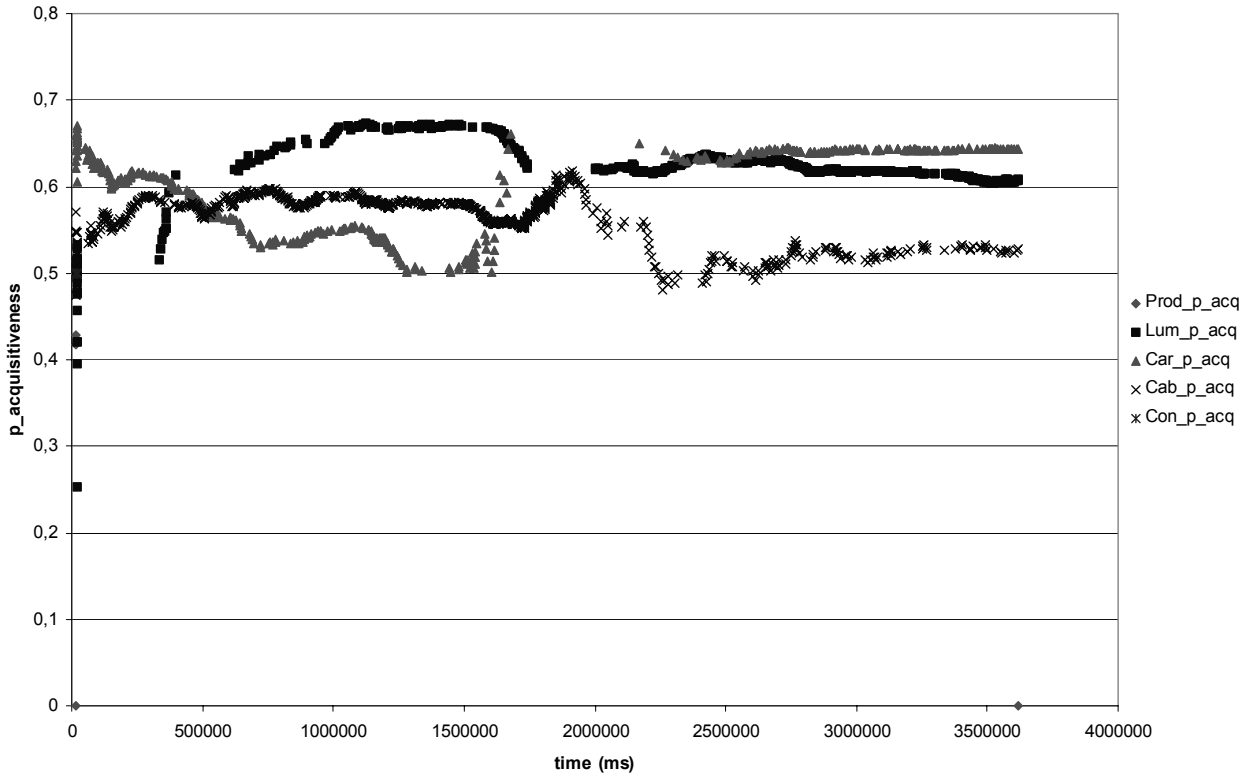

Figure 6: Evolutionary development of the acquisitiveness parameter

implementation uses a decentralized evolutionary algorithm taken from (Smith and Taylor 1998), which is based on posted information about the success of transactions. The *genotype* has been described above, the phenotype is the witnessed strategy and the fitness is the profit over time.

In AVALANCHE, every agent sends one *plumage* object after a successful transaction, advertising its average income and its genes to one random agent of the same type. This can be interpreted as someone receiving insider information about an otherwise private deal information of someone else. As the information is not publicly available, each agent will collect over time a different list of received *plumages*. Now begin the four phases common in evolutionary algorithms (Goldberg 1993). Upon having received a fixed number of 15 *plumages*, every agent ranks them and selects the *plumage* with the highest average income attached (evaluation and selection phase). The *genotype* of the sender is extracted from the *plumage* and compared with the agent's own *genotype* (reproduction phase). This *plumage* will then be crossed over with the agent's own strategy genes, slightly mutated, and "planted" into a new agent which then enters the marketplace (recombination and mutation phase).

In effect, the evolutionary algorithm relegates unsuccessful agents/strategies and promotes successful agents/strategies, changing the composition of the population over time. A variant of this mechanism (not shown here) could replace the *genotype* of the agent with a modified version, which creates a constant sized population of immortal, but learning agents.

Figure 4 has shown how heterogeneous strategies of the agents affected the overall behavior of the system, in case there is no evolution. In contrast, Figure 5 shows the results of an experiment where economic effects are combined with evolutionary learning. The carpenters were, as before, equipped with a lesser probability to make price concessions, and this is again evident in the first half of the experiment run: the spread between the second (boards) and third (panels) price curve increases. But this time, the lumberjack and cabinetmaker agents are (after some time) not only able to withstand the pressure, but to turn the tables and to dominate the carpenter agents.

This is graphically displayed in Figure 6, where every curve represents the average acquisitiveness values of a specific agent type over time. The highest (green) curve at the beginning represents the carpenters, and one can see that their initially high average of the acquisitiveness parameter soon declines too much.

After the first half of the experiment, the carpenters end up with the lowest setting of all three inner agent types. At the same time, the lumberjack and cabinetmaker agents are able to co-evolve and raise their acquisitiveness setting. At around 1500 seconds in the earlier Figure 6, the spread

between board and panel prices had finally vanished and with it the profits of the carpenter agents. In Figure 7 at that time, the average acquisitiveness setting of the carpenter agents reaches the lowest value compared to the other agent types.

This picture changes again in the second half of both charts, when the cabinetmaker agents learn and raise their acquisitiveness again. Similar to the beginning, now the cabinetmaker, and to a lesser extent, the lumberjack agents reduce their acquisitiveness values as shown in Figure 6. The result can be immediately seen in Figure 5: the carpenter agents gain ground, the lumberjack agents are able to withstand the pressure and keep their profit situation, and the cabinetmaker agents lose everything gained in the first half of the experiment run. In summary, a highly dynamic pattern of co-evolution arises, where learning successful parameter settings allows other participants to survive and prosper early while dominating strategies are counterbalanced. Because the size and strategies of the population constantly change, this result will probably never reach a static endpoint, like an evolutionary function optimizer does.

Conclusion: Market Coordination between Economic Concepts and Technical Realization

This article introduced a coordination mechanism for systems of autonomous decentralized devices, which is based on constant negotiation and price signaling. The mechanism is based on efforts from both agent technology and economics, notably agent-based computational economics (Tesfatsion 1997), to develop new technical possibilities of coordinating decentralized information systems consisting of autonomous software agents. It can be shown how this self-regulating market approach exhibits certain coordination patterns and leads to a co-evolution of strategies. If the agents are able to learn and revise their strategies as in human negotiation experiments (Pruitt 1981), a stabilization of price curves may be achievable.

The theoretical background of that coordination mechanism is rooted in the non-mainstream concept of neo-austrian economics (The Ludwig von Mises Institute (ed.) 2001). While the dominant paradigm, the neo-classical, centralized view on economic coordination, has already been successfully realized under the label of *market-oriented programming* (Wellman 1996; Ygge 1998), other fundamentally diverse concepts of how economic coordination works exist, c.f. (Anderson, Arrow, and Pines 1988; Witt 1993). AVALANCHE is regarded as an example of the neo-austrian concept of the *catallaxy* (Hayek et al. 1989), and as a prototype for a class of decentrally market-coordinated multi-agent systems called *catallactic information systems*.

A successful implementation of the distributed resource allocation mechanism introduced here has the advantage of a flexible structure and inherent parallel processing. From an application viewpoint, an appealing aspect lies in the inherently distributed structure, which mirrors the division of work in companies and the network of businesses and markets. By using the same coordination rules that govern the real world, decentralized market mechanisms may allow the creation of innovative fast, flexible, parallel working, and self-regulating IT systems.

Some problems still remain and show where future research work is needed. It is desired to design mechanisms for self-interested agents such that if agents maximize their local utility, the overall system behavior will at least be acceptable (Binmore 1992). Among the problems which such a complex system faces are the over-usage of a shared resource known as “tragedy of the commons” (Hardin 1968), chaotic behavior (Huberman and Hogg 1988; Thomas and Sycara 1998), and the appearance of malevolent agents, which may be countered by the embodiment of centralized and decentralized reputation mechanisms in the system (Padovan et al. 2001).

The findings of this article are only the beginning and clearly need further discussion, especially concerning the link between economic understanding and technological realization of market coordination.

References

Anderson, P. W.; Arrow, K. J.; and Pines, D. 1988. *The economy as an evolving complex system.* Redwood City, CA: Addison-Wesley.

Binmore, K. 1992. *Fun and games - a text on game theory.* Lexington: D.C. Heath.

Chavez, A.; Maes, P. 1996. Kasbah: An Agent Marketplace for Buying and Selling Goods. In *Proceedings of the First International Conference on Practical Applications of Agents and Multiagent Systems*, 75-90. London: The Practical Application Company.

Diebold Consulting (Ed.) 2001. *Digital Business Agents - Benefits and Potential of Multi-Agent Systems.* Eschborn: Diebold Deutschland GmbH. http://www.timelabs.de/

Doorenbos, R. B.; Etzioni, O.; and Weld, D. 1997. A Scalable Comparison-Shopping Agent for the World Wide Web. In *Proceedings of the 1st International Conference on Autonomous Agents (Agents'97)*, 39-48. Marina Del Rey, CA.

Erev, I.; Roth, A. E. 1998. Predicting How People Play Games - Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria. *American Economic Review* 88(4):848-881.

Eymann, T. 2000. *Avalanche - ein agentenbasierter dezentraler Koordinationsmechanismus für elektronische Märkte.* PhD Thesis, Albert-Ludwigs-Universität Freiburg. http:// www.freidok.uni-freiburg.de/ volltexte/ 147/.

Goldberg, D. 1993. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley.

Guttman, R. H.; Maes, P. 1998. *Agent-mediated Integrative Negotiation for Retail Electronic Commerce*. Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98), Minneapolis.

Hardin, G. 1968. The Tragedy of the Commons. *Science* 162:1243-1248.

Hayek, F. A.; Bartley, W. W.; Klein, P. G.; and Caldwell, B. 1989. *The collected works of F.A. Hayek*. Chicago: University of Chicago Press.

Huberman, B. A.; Hogg, T. 1988. The behavior of computational ecologies. In Huberman, Bernardo A. (ed.): *The Ecology of Computation,* 77-115. Amsterdam: Elsevier Science.

Kephart, J. O.; Hanson, J. E.; and Greenwald, A. R. 2000. Dynamic Pricing by Software Agents. *Computer Networks*, forthcoming. Hawthorne, NY: IBM Research. http://www.research.ibm.com/ infoecon/paps/ html/ rudin/ rudin.html, access date 20-3-2001.

Kraus, S. 1997. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence* 94:79-97. http:// www.cs.biu.ac.il:8080/~sarit/Articles/ct.ps, access date 20-3-2000.

Kreifelt, T.; von Martial, F. 1991. A Negotiation Framework for Autonomous Agents. In *Proceedings of Decentralized AI II*. Elsevier Science.

Lindemann, M. A.; Schmid, B. 1999. Framework for Specifying, Building, and Operating Electronic Markets. *International Journal of Electronic Commerce* 3(2):7-22.

Lomuscio, A. R.; Wooldridge, M. J.; and Jennings, N. R. 2000. A classification scheme for negotiation in electronic commerce. In Sierra, C.; Dignum, F. *A European Perspective on Agent-Mediated Electronic Commerce*, 19-33. Heidelberg: Springer.

Objectspace Inc. 1999. *Voyager Documentation*. Objectspace Website http://www.objectspace.com/ products/ voyager/, access date 1-3-2000.

Padovan, B.; Sackmann, S.; Eymann, T.; and Pippow, I. 2001. A Prototype for an Agent-based Secure Electronic Marketplace including Reputation Tracking Mechanisms. In *Proceedings of the 34th Hawaiian International Conference on Systems Sciences*. Outrigger Wailea Resort, Maui: IEEE Computer Society.

Preist, C. 1998. Economic Agents for Automated Trading. *HP Technical Reports HPL-98-77*. Bristol: Hewlett Packard Laboratories.

Pruitt, D. G. 1981. *Negotiation Behavior*. New York: Academic Press.

Rosenschein, J. S.; Zlotkin, G. 1994. *Rules of encounter - designing conventions for automated negotiation among computers*. Cambridge: MIT Press.

Schenker, J. L. 28-2-2000. The Trillion-Dollar Secret. TIME Magazine 155(8). http://www.time.com/time/ europe/ magazine/2000/228/trading.html, access date 13-3-2001.

Sierra, C. 2000. Agent-mediated Electronic Commerce: Scientific and Technological roadmap. In Sierra, C.; Dignum, F. *A European Perspective on Agent-Mediated Electronic Commerce*, 1-18. http://www.iiia.csic.es/ AMEC/ Roadmap.ps, access date 1-1-2001.

Smith, R. E.; Taylor, N. 1998. A Framework for Evolutionary Computation in Agent-Based Systems. In Looney, C.; Castaing, J. (eds.): *Proceedings of the 1998 International Conference on Intelligent Systems,* pp. 221-224. http://www.ics.uwe.ac.uk/~rsmith/fecabs.pdf, access date 1-1-2001.

Tesfatsion, L. 1997. How economists can get alife. In Arthur, W.B.; Durlauf, S.; and Lane D. (eds.), *The Economy as an Evolving Complex System, II*, 533-564. Redwood City, CA: Addison-Wesley.

The Ludwig von Mises Institute (ed.) 2001. The Ludwig von Mises Institute Website. http://www.mises.org/, access date 9-8-2001.

Thomas, J.; Sycara, K. 1998. Stability and heterogeneity in multi agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems*, 293-300. Paris, France.

Weiser, M. 1991. The Computer for the Twenty-First Century. *Scientific American* 265(3):94-104.

Wellman, M. P. 1996. *Market-Oriented Programming: Some Early Lessons*. Singapore: World Scientific.

Witt, U. 1993. *Evolutionary economics*. Aldershot, Hants, England: E. Elgar Pub.

Wooldridge, M. J. 1999. Intelligent Agents. In Weiss, G. *Multiagent Systems*, 27-78. Cambridge: MIT Press.

Ygge, F. 1998. *Market-Oriented Programming and its Application to Power Load Management.* PhD Thesis, Lund University, Sweden. http://www.enersearch.se/ knowledgebase/publications/conference-journals/ thesis/ thesis.html, access data 3-8-2001.

Youll, J.; Morris, J.; Krikorian, R.; and Maes, P. 2000. Impulse: Location-based Agent Assistance. In *Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000)*. Barcelona.