Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I
IMECS 2009, March 18 - 20, 2009, Hong Kong

Automated Services for Market-Based E-Commerce Transactions

Freimut Bodendorf and Florian Lang

Abstract: **Business transactions focus on the delivery of goods and services from one business partner to another. IT-based services support these transaction activities between suppliers and customers. Usually the transaction process consists of three phases - the information phase, the negotiation phase and the settlement or fulfillment phase. Within the negotiation phase one vital and challenging task is to come to a contract. This is especially true if you apply market mechanisms, i. e. addressing and negotiating with many potential partners. Managing flexible transactions with varying market partners, considering multidimensional terms and conditions, and acting within short periods of time are crucial challenges for automated negotiation services on electronic markets. After introducing a real-time market scenario the approach of intelligent software agents providing automated services is presented. Aspects on negotiation protocols and implementation are outlined. Insights into sample experiments and results are given.**

Index Terms: **E-Markets, E-Transactions, Software Agents**

I. REAL-TIME BUSINESS

Many businesses rely on products or services obtained from different suppliers. In order to keep a business system adaptive, e.g. to changing customer requirements, lean procurement procedures must be installed. Just-in-time procurement is a salient example. Switching suppliers includes analyzing the market, selecting a supplier based on a set of criteria, negotiating terms and conditions, signing a new contract, and fulfilling the transaction.

Real-time business applies techniques for automating the time and money consuming process of managing flexible transactions between suppliers and consumers. Many transactions are heavily time dependent, i.e. quantity and quality of transaction objects are changing within short periods of time. Examples are food or flowers, bandwiths of communication services, capacities of transport services or short term energy supply.

Manuscript received December 9, 2008.
Freimut Bodendorf is head of the Department of Information Systems II at the University of Erlangen-Nuremberg, Germany (e-mail: bodendorf@wiso.uni-erlangen.de)
Florian Lang is with the Department of Information Systems II at the University of Erlangen-Nuremberg, Germany (e-mail: florian.lang@wiso.uni-erlangen.de)

The market platform introduced in this paper provides automated transaction services to consumers and providers, thereby enabling them to trade in real-time.

II. MARKET SCENARIO

An electronic market supports the management of transactions between supply and demand by information and communication systems. Negotiating contracts is the central task of market transaction management. So, negotiating a contract is the focal transaction service among the transaction services a market platform offers to vendors and buyers. Often not only price but also other product or service parameters have to be agreed on. This is a big challenge for "intelligent" automated negotiation procedures.

Here, an agent-based approach to transaction automation in real-time scenarios is introduced. Fig. 1 shows an example of a real-time market scenario using knowledge based negotiating software agents. The real-time market platform comprises transaction services for different tasks. Matching supply and demand is done by the agents both on the supplier's and the consumer's side. Another transaction service covers for example the administration of service level agreements (SLA), i.e. the specification of negotiated services. Another service supports the "delivery logistics" while a monitoring service checks the delivery and payment process for compliance with the contract and a potential service level agreement.

III. AUTOMATED NEGOTIATION

The automated negotiation process follows a finite-horizon version of the alternating offer protocol. The negotiation protocol supports sequential alternating offer exchange among two agents that apply heuristic strategies to gain advantage over their competitors. Software agents need to have a specific transaction processing framework for their decision making and interaction. The transaction process is split up into an ordered sequence of basic actions triggering one another. Each basic action represents a singular decision problem of the negotiating agents. With the completion of such a basic activity, an agent may either begin the succeeding activity as given in the protocol or trigger another agent by deploying a standardized speech act via the message handling system provided by the negotiation platform.

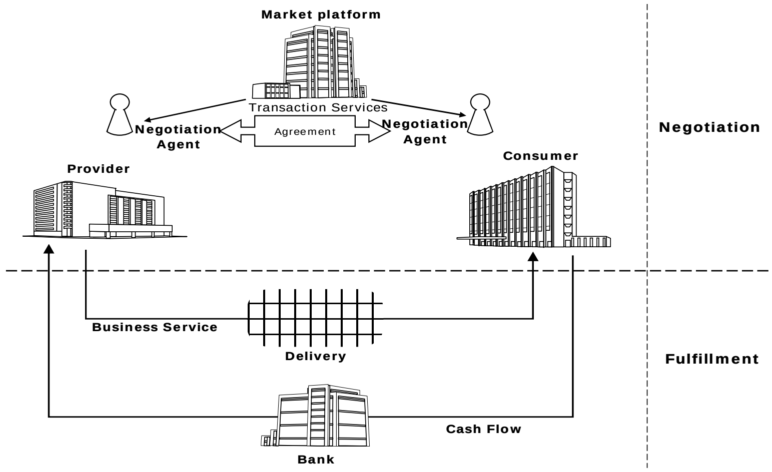


Fig. 1. Real-time market scenario

Fig. 2 shows the sequence of basic activities that form the negotiation process. The set of valid speech acts, the rules determining the usage of these speech acts and the sequence of basic activities form the negotiation protocol. The protocol allows the agents to repeatedly exchange and evaluate offers in terms of proposed joint decisions (contracts).

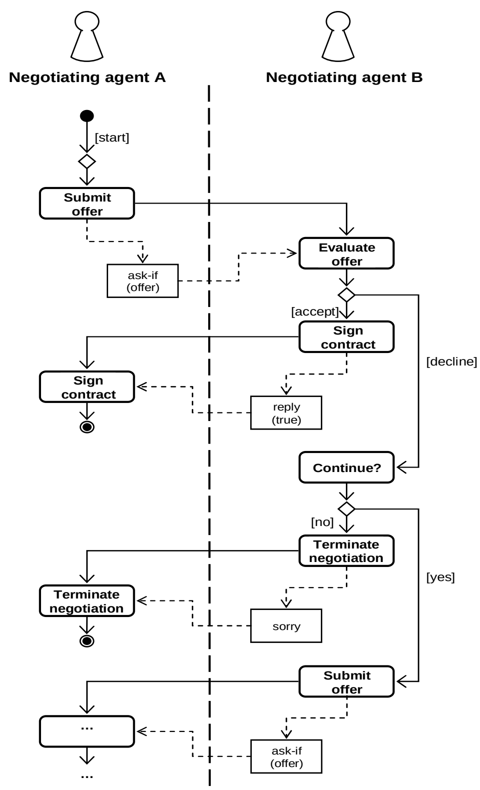


Fig. 2. Negotiation protocol

If a matching agent with compatible goals is found, an agent starts off negotiations by generating an initial offer containing a proposed value for each of the negotiable issues and sending it to the matching agent. The basic activities “submit offer”, “evaluate offer” and “continue?” are then iteratively performed by both agents until either an agreement is reached or the negotiation is terminated.

Since investigating favorable negotiation strategies is a major research goal, the negotiation protocol is designed to minimal restriction of the agents’ behavior. It only determines what decisions must be made and when, thus being a flexible container where any negotiation strategy can be plugged in. A negotiation strategy provides decision rules for each of the basic activities. A protocol with singular decision problems allows testing and modifying or replacing the strategy module of a single basic activity.

IV. PROTOTYPE IMPLEMENTATION

A browser-based implementation for the central market platform primarily serves as an administration back-end that allows management of tradable services, e.g., telecommunication capacities, energy supply, web services or grid services. It includes statistical analysis functions for administration purposes and control functions for the negotiation agents.

A negotiation client has been developed (see Fig. 3) which is the implementation of the knowledge based agent approach to automated negotiations. It communicates with other agents belonging to market partners to support fully automated matching and negotiation. A user can specify the degree of autonomy the client displays (that is why it is called a client, not an agent). Thereby, its usage ranges between merely providing a front-end that supports inter-personal negotiation and a fully-fledged automation suite that is just occasionally attended to by a user. It runs under Windows XP and Vista.

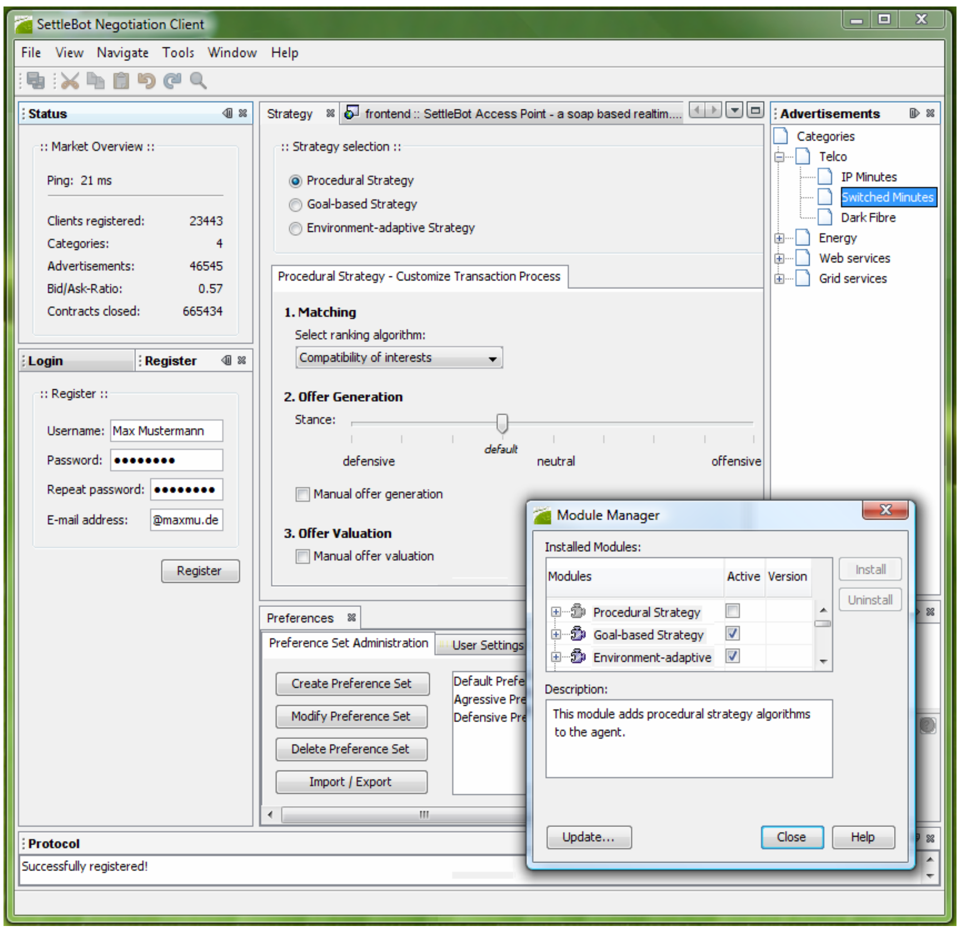


Fig. 3. Negotiation client

V. SAMPLE EXPERIMENT AND RESULTS

Fine-tuning the decision functions used by an automated agent means finding a good solution under very complex conditions. These conditions include a dynamic market environment, unforeseeable reactions of other agents, and improper knowledge about how to make decisions in a way that additional utility can be gained. For example, it appears rational for an agent to reduce its reservation utility as the deadline approaches that the user has set for a successful settlement. But will reducing the reservation utility u_{res} over time really increase an agent's success? If so, at what ratio should u_{res} be reduced? And does this strategy always work? What, if the agent is the lone seller for a good and can demand whatever utility it chooses to? Will reducing u_{res} over time still increase an agent's success if it is negotiating very aggressively or rather very defensively?

The point is, of course, there are many interdependencies to be considered that are beyond the reach of any theoretical analysis. Here, a simulation system that uses genetic engineering of negotiation strategies serves as a tool for handling this kind of complexity. In the simulation system, negotiating agents are evaluated by the results of a sequence of negotiations on a simulated market (see Fig. 4). Their genome that represents all variable aspects of their negotiation strategy is then crossbred with other agents and passed on to the next generation depending on the parents' success.

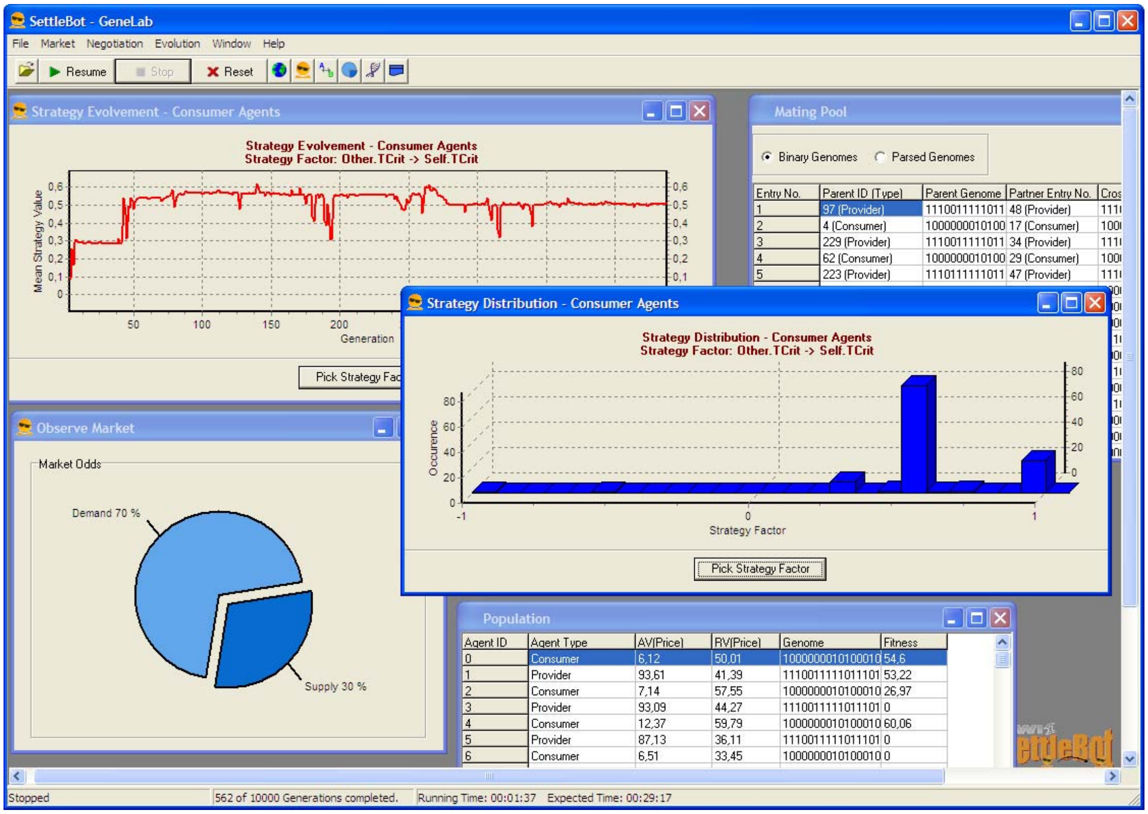
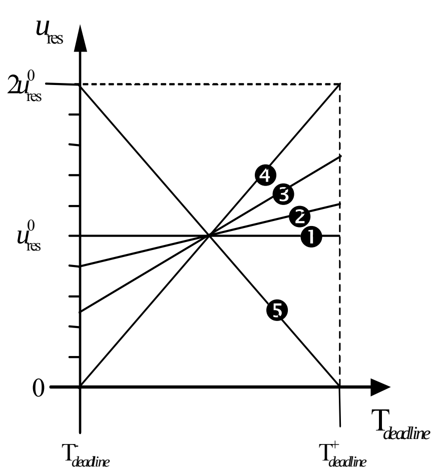


Fig. 4. Simulation and genetic engineering of negotiation strategies

After a number of generations, good solutions evolve that negotiate successfully under the given (dynamically changing) negotiation setting and preferences. Thereby, it is possible to quickly search a huge space of potential solutions. The parameter "ratio of diminishing u_{res} while deadline approaches" is just a tiny fraction of what a genetic algorithm can find favorable values for in parallel. Fig. 5 illustrates how this is done for a single parameter.



$$u_{res}(T_{deadline}) = u^0_{res}\left(1+\left(\frac{2\left(T_{deadline}-T^-_{deadline}\right)}{T^+_{deadline}-T^-_{deadline}}-1\right)\cdot\omega\right)$$

Fig. 5. Sample experiment

ISBN: 978-988-17012-2-0

u_{res}^{0} is the unmodified parameter before the strategy is applied. The straight line 1 has a slope of 0 and will not modify u_{res}^{0} in any way. It is likely that one of the lines 2 to 4 will increase an agent's success since they all reduce u_{res} as the deadline $\mathrm{T}_{deadline}$ gets shorter (it moves to the left on the x axis). Line 5 will rather diminish an agent's success. To find a suitable strategy for this decision function, negotiating agents use the expression given above as their strategy, with ω (omega) as a genetically encoded parameter. ω determines the slope of the line used to modify an agent's current u_{res}. To make this example complete: simulation shows that $\omega=0{,}3$ is favorable under most conditions. Further improvement can be achieved if ω is varied in the range $\omega \in [0;0{,}3]$ depending on changes in the negotiation setting.

VI. CONCLUSIONS

In their "Research Manifesto for Services Science", Chesbrough and Spohrer [1] suggest six aspects that form the common foundation of a services science field. The lessons learned through the work described in this paper relate to all of these aspects and support Chesbrough's and Spohrer's grasp of the field:

- Close interaction of supplier and customer:
  It is not the lack of semantically rich communication tools that limit the close interaction of suppliers and customers. It is limited by the effectiveness and even more by the efficiency of preference elicitation methods.
- Nature of knowledge created and exchanged:
  Superior agent-borne knowledge is all about adaptivity. Aware sensors and a flexible knowledge model are the key to this adaptivity.
- Simultaneity of production and consumption:
  Since there is no way to store a service away until it is needed, there is no doubt about this aspect being a requirement for any service-driven system. It is a restriction that gives any services science research work a design guideline.
- Combination of knowledge into useful systems:
  Designing an agent that prevails under limited information is achieved (however to a limited degree) by combining and integrating bits and pieces of knowledge won by experimenting and simulation.
- Exchange as processes and experience points:
  The real-time character of demand creation and fulfillment makes a transaction service for automated negotiation "business process attachable".
- Exploitation of IT and transparency:
  Leveraging the advances of IT for the profitable creation and beneficial consumption of valuable services is the prime challenge in this field of research. The knowledge-intensive character of a service and its uno-actu-principle (simultaneity of production and consumption) call for sophisticated knowledge management, real-time communication and real-time processing that can only be implemented by IT.

In a trading scenario like it is envisioned here, new business models will evolve especially trading business services as intangible goods. A service provider may be a mass producer or a re-seller for instantly deliverable services traded by the agents. Consumers may sell business services while their resources are idle and buy services back when they need them, dramatically reducing the need for their own service capacity. Here, the buzzword "prosumer" appears. Even more specific to this new form of e-commerce is the idea of trading valuable knowledge that the agents can use in their negotiations (e.g. market statistics) or the idea that promising negotiation strategies are traded. The prototype application presented in section 4 allows for plugging strategy modules into an agent on-the-fly. This flexible extensibility may ultimately lead to an agent that updates itself with new strategies depending on market requirements.

Despite promising advances, there are still many limitations to automated negotiations. The limitations lie in the intelligence of the negotiating agents. Only if an agent can be trusted to settle favorable contracts under all conditions an automated system will replace fixed contracts or inter-personal negotiation. Given the technical possibilities to develop sophisticated negotiation strategies with the help of simulation and genetic engineering, a perpetually evolving e-business structure as presented in sections 1 and 2 is possible by the current state of the art.

REFERENCES

[1]. H. Chesbrough, J. Spohrer, "A research manifesto for services science", *Communications of the ACM*, 2006, pp. 35-40.