Agent Mediated Electronic Commerce Research at Hewlett Packard Labs, Bristol

Chris Preist
Hewlett Packard Labs
Bristol

Agent technology is a key enabling component in bringing about Hewlett-Packard e-services vision, and for that reason, HP Labs is making significant investments in research in this area. Work on agent technology at HP labs includes agent-mediated B2B e-commerce, distributed agent-based personalization, agent-based trading of Internet bandwidth, automated auction design, mobile agents, user profiling and FIPA standardization activities, among other things. In this article, we describe our work on Agent-mediated B2B e-commerce. We do that based on the lifecycle of a B2B interaction in which the contract plays a fundamental role. The lifecycle comprises the stages of matchmaking, negotiation, contract formation and contract fulfillment. We highlight our research in each of these areas and in service composition.

1. EVOLUTION OF ECOMMERCE

Electronic commerce is having a revolutionary effect on business. It is changing the way businesses interact with consumers, as well as the way they interact with each other. Electronic interactions are increasing the efficiency of purchasing, and are allowing increased reach across a global market.

E-commerce is not a static field, but is constantly evolving to discover new and more effective ways of supporting business. Initially, e-commerce involved the use of EDI and intranets to set up long-term relationships between suppliers and purchasers. This increased the efficiency and speed of purchasing, but resulted in lock-in in the relationship. Both suppliers and purchasers had to invest significantly up-front in the relationship, so were not easily able to move their business elsewhere. The technological relationship between the parties was a friction factor, preventing free competition in the longer term. Often, the relationship was

Address: Hewlett Packard Labs
Filton Rd
Stoke Gifford, Bristol BS32 8EH
UK
Chris_Preist@hp.com

(and still is) beneficial to both parties. However, the lock-in effect also meant that when the relationship became less beneficial to one party, they couldn't easily move elsewhere.

The second phase of e-commerce aimed to address this problem. With the increasing availability of the web, a more open e-commerce environment is developing, allowing businesses to trade more flexibly with each other. Some of this openness is achieved by competition between web portals, while some competition occurs within a single web portal, acting as a marketplace for buyers and sellers to meet. Some of the efficiencies of EDI can now be achieved in a more open environment, where relationships no longer need to be long-term.

However, there is a benefit of the EDI approach that is often lost in this new phase. Price negotiation was carried out in advance in the EDI world, so purchasing can be entirely automated. When a manufacturing planning and forecast system identifies the need for a purchase, it can initiate it automatically without any human involvement, increasing speed and efficiency. In phase two, each purchase may involve interaction with a new supplier, and so may involve new negotiation of terms. As a result of this, many of these purchases can't be made automatically, and instead require human interaction, mediated by the web. The third phase of electronic commerce is just beginning. It aims to address this issue, allowing automated business interactions to take place in a fluid environment. Technology will no longer be a friction factor to changing supplier or customer. Long-term relationships will still play an important role, but they will persist because of the choice of both parties rather than technological lock-in. The key building blocks of this new world, e-services, will be able to interact dynamically with each other to create short-term or long-term trusted trading relationships to satisfy the needs of different business partners. Hewlett Packard is engaged in research in many technologies to help bring this shift about, including distributed systems, encryption and PKI, XML and associated business ontologies, economic analysis and game theory. As automation and distribution are central to the vision, agent technology provides a fundamental role. In this article, we describe agent technology research in Business-to-business e-commerce that has been carried out by members of the E-Service Markets portfolio at HP Labs Bristol,

2. THE E-SERVICE MARKETS PORTFOLIO

The E-Service Markets portfolio is a team of 15 researchers (full-time staff and visiting academic placements) within the Solutions and Services Technologies Centre of HP Labs. Our charter is to generate innovation in the automation of B2B electronic commerce, and to support HP in exploiting these. We do this in several different ways;

- Consultancy work to solve a specific problem for a product or service division in HP.
- Prototype development of potential products, and collaborative refinement and productisation with a product division.
- Input into the strategy development process of different parts of the company.
- Joint work with HP Consultants and their clients to co-invent innovative solutions, and to support the development of these.

—Exploratory research, leading to publication and/or intellectual property.

This work results in us collaborating in different ways with many people both inside and outside HP. In particular, we work closely with other teams in HP labs engaged in related work. The Solutions and Services Technologies Centre is split between Bristol and Palo Alto. Within Bristol, we have carried out joint work with security experts on the application of security protocols to auctions and markets. We have carried out collaborative work with economists in Palo Alto on the dynamics of markets, and with distributed systems experts on protocols for electronic commerce.

In this paper we will focus on our more speculative research program, and some of the results so far.

3. THE E-COMMERCE LIFECYCLE

At HP, our portfolio of research activities is based around the lifecycle of a B2B interaction. In B2B, the contract plays a fundamental role. The lifecycle (based on that in [Jennings et al. 1996]) has the following stages;

(1) Matchmaking: A trader locates other traders that it could potentially do business with.

(2) Negotiation: The trader enters into negotiation with one or more of these potential business partners, to see if they can agree to mutually acceptable terms of business. These terms could include a definition of the good or service being traded, price, delivery date, etc.

(3) Contract Formation: These agreed terms are placed into a legally binding contract.

(4) Contract Fulfillment: The parties carry out the agreed transaction, within the parameters specified in the contract.

We now describe our research in each of these areas.

4. MATCHMAKING

During the matchmaking phase a trading agent locates potential trading partners. It does this by use of a description of the service (or good) it is interested in trading. It may advertise this description, attempt to search for relevant advertisements by others, or a combination of both. Adverts may be by service providers, or may be by service requestors. In the latter case, this is equivalent to a Request For Quote (RFQ) as used by companies when putting services out to tender.

We have defined a language for describing service descriptions used during a B2B interaction. During the matchmaking phase, these descriptions appear within the adverts and the queries over the adverts. However, the goal of our work is to design the language in such a way that it is sufficiently expressive to be used during negotiation and contract specification.

We performed a requirements analysis on such a language by developing a set of use cases for the matchmaking service. We determined the requirements to be:

(1) A high degree of flexibility and expressiveness, including capabilities for nested service specifications, and tree/graph structures.

(2) The ability to express semi-structured data. Businesses should be able to advertise under-specified services, without even knowing in advance the particular parameters of that service that a customer will require to make a decision. These additional parameters and values can be determined at a later stage of negotiation.

(3) Support for types and subsumption. It should be possible for a customer to make a generic request, such as 'travel', and locate services that are subtypes of this.

(4) Ability to express constraints. An advert or a service request is usually not for a single instance of a service, but rather a conceptual definition of acceptable instances. Constraints are a natural and compact way of doing this.

Existing frameworks for ECommerce do not meet these requirements. UDDI [Boubez and al. 2000] lacks a means of classifying the data structures, and searching is only done by equality matching on certain fields. As a result, it can only provide a first level filter, requiring direct communication with potential service providers to complete the matchmaking phase. ebXML does not support semi-structured data, inheritance and constraints. Other ecommerce frameworks such as RosettaNet provide only a basic ontology definition for the matchmaking stage.

Because of these shortfalls, we turned instead to RDF [Lassila and Swick 1999] and DAML [Hendler and McGuinness 2000], both of which offer a far richer language for expressing metadata. We describe an advertisement as an RDF graph or DAML description logic expression that defines a space of possible realisations of one or more services. We have defined a matching algorithm in both contexts. The algorithm allows advertisements of buyers and sellers to match not only if one is a special case of the other, but also if they are compatible with each other. For a given buyer query, it will efficiently identify all the compatible sellers, or vice-versa. Full details of this are given in [Trastour et al. 2001a].

The matchmaker system has been implemented on top of the FaCT Description Logic reasoner [Horrocks 1998]. These provided significant functionality. However, neither of them handled the required level of expressivity (SHOQ(D) logic). As a result of this, we had to transform the representation manually into a more counter-intuitive form to allow complete matching to occur [Trastour et al. 2001b].

Using this approach, we have successfully developed a system that allows rapid matching over large numbers of complex service descriptions to take place. The system is more expressive than existing industry standards such as UDDI, and academic prototypes such as DGQL [Reynolds 2001]. Currently, we are working to extend it to allow behavioural descriptions of complex services, and to integrate it with a persistent store.

5. NEGOTIATION

When a trader has located possible trading partners, it can enter into negotiation with them in an effort to reach an agreement with one or more of them. If negotiation is to take place automatically, traders must agree on a negotiation protocol, and each trader must have a private negotiation strategy. The role of the negotiation protocol is to coordinate the message flow between participants, and to impose the 'rules' of the negotiation 'game'. These rules may be chosen to ensure fairness, to encourage honesty, or to maximize the profits of one particular powerful trader. There are many possible protocols. For example, the English auction where many buyers place increasing bids to trade with one seller. The negotiation strategy used by a trader determines how it behaves within the rules provided by the protocol. For example, in an English auction, a buyer's strategy determines when and how much they bid.

Our research in the area of negotiation protocols has focused on the creation of a general framework. This framework can represent many different protocols, by parameterization with appropriate rules. It is able to handle 1 to 1, 1 to many and many to many negotiations. By using it, agents avoid the need to hold a large number of different protocols to cover different negotiation types, and are even able to participate (albeit in a flawed way) in negotiations where they do not know the protocol.

In our framework, negotiation consists of an exchange of proposals described using the service description language developed for matchmaking. The partial match tool described in the matchmaking section above is used to identify which proposals are closest to which, and what areas of disagreement remain. The agents can use this as input to their strategy.

In addition, we have carried out research in different issues related to negotiation strategy.

5.1 Negotiation in Multiple Markets

Through matchmaking, a trader will locate a potentially large number of possible trading partners. These partners may be spread across different markets. For example, some may be in an auction house, conducting trade via pre-arranged auctions. Others may be participating in a many-to-many trading exchange. Others may choose to do business directly, via 1-1 negotiation or simply offer a fixed price. One key issue facing the trader is how to simultaneously negotiate with all these potential traders, in different markets, to try to reach a good deal. Certain kinds of market, such as auctions, require a commitment to purchase if your bid is accepted. Hence, an agent trading simultaneously in several auctions must be careful not to accidentally buy too many goods. We are interested in developing algorithms that will allow negotiation simultaneously in many markets to get the best deal, but will avoid accidental purchase of too much.

Our work so far has focused on English auctions. We have developed an algorithm that will purchase an appropriate number of goods from a set of auctions, at a cheap price. The algorithm uses historical data together with utility analysis to determine the best set of bids to place at any given time [Preist et al. 2001]. TO determine the maximum bid it is prepared to place in the next auction(s) to close, it calculates the expected utility of withdrawing from that auction. It does this using the historical data of previous similar auctions, and the current bids in each auction, to determine the current expected distribution of closing prices in each auction that will remain. It uses this to find the remaining auction(s) with the highest utility. It then continues to bid in the auction until the actual utility of the bid it would place is less than the expected utility of withdrawing.

This work is the first published example of an agent that participates in multiple overlapping auctions to purchase similar goods. Subsequently, [Anthony et al. 2001] have proposed an alternative, heuristic, approach. They generalise beyond English auctions, but focus on the purchase of a single good rather than a number of similar goods.

We have carried out experiments using this algorithm, and have demonstrated that it performs better than the standard strategy. We have also shown that, if many buyers use the algorithm, it increases the allocative efficiency of the market. However, interestingly, in certain circumstances the benefit from this increased efficiency goes to the auctioneers rather than to the buyers [Preist et al. 2001].

We have also developed a more general mathematical specification for a class of algorithms to do this, where different tradeoffs between accuracy and efficiency can be chosen.[Byde 2001]

5.2 Negotiation for Service Composition

Negotiation in multiple markets becomes more complex when trying to buy a connected set of goods or services simultaneously. For example, when trying to send a crate from London to Prague, there are many different sets of flights that could be used (e.g. $LON\ !\ PRA$; $LON\ !\ AMS + AMS\ !\ PRA$; etc.). If you are negotiating simultaneously with different flight suppliers, you must be careful to buy exactly one set of flights, and not get stuck with partial (and therefore useless) sets. We have developed an algorithm that participates in multiple English auctions, and trades off the risk of bidding for the current best set with the potential risk of accidentally buying unwanted goods [Preist et al. 2001].

Our algorithm firstly calculates the expected utility of buying each potential set of goods from given auctions, assuming that we continue to bid in those auctions and do not switch. This is done in a similar way to the algorithm discussed in the previous section. It then bids in the auctions to buy the set with the best expected utility. However, as the auctions progress, it calculates the expected utility of switching to a different set. This is the expected utility of winning the alternative set, less the expected cost of accidentally winning one or more of the existing bids that have been placed for the original set. If the utility of switching is greater than the utility of remaining committed to the original set, then the algorithm takes the risk and places bids to buy the new set.

5.3 Negotiation over Multiple Parameters

Another important issue is the ability to negotiate over complex contracts with many different variables. For example, making a tradeoff between the price of a product, the quality and the speed of delivery. Previous work has been carried out on the development of algorithms for performing one-to-one negotiation of this kind. Faratin et.al. [Faratin et al. 1998] have developed a heuristic approach, and [Barbuceanu and Lo 2000] propose an approach based on multi-attribute utility theory. However, no prior work has considered many-to-many negotiations of this form.

To tackle this problem, we are building on previous work in HP Labs on lightweight trading algorithms in double auctions. [Cliff and Bruten 1998] developed a simple algorithm using adaptive heuristics to participate in a double auction trading a private-value commodity, and showed that it would result in the market stabilising to its expected equilibrium price. [Preist and van Tol 1998] adapted the algorithm to allow multiple parties to bid simultaneously, resulting in more rapid market stabilisation, and [Preist 1999] used it to determine an a-priori clearing price in a market where traders wish to all trade at the equilibrium price, but do not trust a central auctioneer.

These algorithms were not intended to be the 'best', but rather were intended to be the minimal reasoning ability to allow a market to stabilise. Indeed, subsequently, more sophisticated algorithms which outperform these, such as the fuzzy logic based approach of [He et al. 2001], have been developed. However, despite this motivation, recent experiments [Das et al. 2001] have shown that these algorithms are able to outperform humans in an experimental trading scenario.

We are generalising these algorithms to cover many-to-many negotiations where the good is not a commodity, but instead has several discrete parameters that may be varied. We have demonstrated that the generalised algorithm results in the market trading at the theoretical equilibrium. We plan to use these as a baseline for further research, to develop more sophisticated strategies in such a marketplace.

5.4 Preference Acquisition for Negotiation

The work we have presented above, like the vast majority of research in the field, is focused on the development of protocols and effective strategies for negotiation. While this is clearly important, it is only part of the picture. In most cases, agents negotiate on behalf of their owner (which might be an individual or an organisation). For this to be effective, agents must be able to adequately represent their owners' interests, preferences and prejudices in the given domain such that they can negotiate faithfully on their behalf. However, at this time, no thought has been given to the problems of specifying exactly what knowledge an owner needs to impart to their agent to achieve high fidelity negotiation behaviour, nor how such knowledge can be effectively acquired from the owner. These are serious shortcomings that need to be addressed if negotiating agents are to be widely used.

Against this background, we have started a joint project[1] with Nick Jennings and Nigel Shadbolt of the University of Southampton, UK, to develop tools and techniques to facilitate this process. The initial stage of the project is concerned with analysing the knowledge requirements of a range of negotiation models. The aim of this analysis is to identify the knowledge components that are common to all (or at least many) negotiation models and those that are specific to particular types of negotiation model. To capture these knowledge requirements the CommonKADS methodology is being exploited. This methodology has a long history of being exploited in knowledge intensive systems and a negotiating agent is certainly that! Having identified the knowledge requirements for negotiating agents, the next step is to identify the techniques that can be used to acquire this information from users. Here techniques from the knowledge acquisition community will be analysed to determine their appropriateness for this task.

6. CONTRACT FORMATION AND FULFILLMENT

When negotiation is over, the agreement that has been reached must be formally captured. We are working on a precise descriptive model of electronic contracts,

[1] ANNA: http://www.iam.ecs.soton.ac.uk/projects/anna/

which will be able to capture a wide variety of outcomes of the agreement formation process between negotiating parties. The resulting contract formally defines the relationship between the two parties. Each party must then behave within the agreed parameters, or alternatively transgress and face the appropriate sanctions. Finally, the two parties must coordinate their behaviour to ensure that both parties recognise that the agreement has been met or not. We have developed a prototype contract management system, and an associated contract fulfillment protocol, which can support an enterprise in the management of the post-contractual relationship.

6.1 The Contract Model

The contract is formed from normative logical statements stating that the parties have permissions, obligations and prohibitions to carry out certain acts (such as paying, delivering a good, etc.). Following [Dignum and Weigand 1995], our initial model uses Meyer's dynamic logic formalism of deontic logic [Meyer 1988] to represent these normative statements. Rather than treating an obligation as an absolute situation, we associate a sanction with it. If the obligation is violated, the obligee will suffer the sanction. Hence, our basic contractual term is of the form; ns:$\theta_{s,b}(\varphi \; ! \; \alpha/\pi)$. This can be read as: 'If φ holds, then s is obliged/permitted/prohibited by b to do α, or suffer the sanction π.'

The contract consists of a set of normative statements and relationships between them. The relationships state that one normative statement is conditional on the performance or the violation of another normative statement.

6.2 The Contract Manager

The contract manager can provide assistance or even complete automation of the execution of business processes to ensure that a contract is fulfilled. At any given time, the system identifies which obligations remain on the party, their relative priorities, and dependencies between them (possibly involving other parties). Based on this information, it schedules some normative statements to be fulfilled immediately, and others to be delayed. To fulfill the normative statements, it can use any business process execution system, such as a workflow engine. It maintains a database containing information on the relationship between abstract actions mentioned in the contract model, and possible concrete instantiations of such actions that are possible through the business process execution system. The contract manager for a given user communicates with other users, and their contract managers, using a contract fulfillment protocol. This is used to ensure that the users' different views of the process are reconciled.

7. SERVICE COMPOSITION

In section 5.2, we described our work on negotiation for service composition. We are also carrying out broader research on appropriate technologies to enable service composition to take place automatically. We have developed a system, named DySCo [Piccinelli and Mokrushin 2001], which allows services to specify models of how to interact with them in an integrated way - behavioural models, and specifications of the roles that potential interactors may take. It allows services to explicitly specify aspects of functionality they wish to outsource, and the roles the outsourcers would take.

When negotiating for provision of a service, two parties can use this representation to identify differences in the way they interoperate. They can use this information to avoid collaborating and search for more compatible partners. Alternatively, they can enter into negotiation, and attempt to negotiate changes in their behavioural models to allow interoperation to take place more smoothly.

The system also allows a single party to orchestrate several services, to ensure they collaborate effectively and smoothly. It allows a distributed workflow to be set up to represent the overall process and allows appropriate projection of this workflow for each role that a service provider performs. It also allows verification that each of the roles are behaviourally compatible, and identifies any incompatibilities so they may be resolved.

7.1 The Cooperation Model in DySCo

Key to the functionality of the DySCo system is the layered cooperation model, which allows a clean separation between the cooperative process and the internal processes used by individual members. The model has three levels of abstraction: cooperative step, cooperative process, and cooperative framework. A *cooperative step* is an atomic unit of activity inside a cooperative process. It can involve a single role, or it can require the coordination of two or more roles. The actions performed by the roles within the step are of three types. They can pass data between roles, they can share data with all roles involved in the step via a logical storage space, or they can execute specific service-oriented activities using role-specific internal processes. For example, a cooperative step may involve the transfer of a bill from a seller to a purchaser. A *cooperative process* is a process which orchestrates a specific aspect of cooperation between a number of roles. It consists of a process made up of cooperative steps, and modelled using the standards of the Workflow Management Coalition [Hollingsworth 1994]. An example process would be the payment of an invoice, which could include roles such as customer, seller, bank and archive provider. A *cooperative framework* consist of a set of cooperative processes, which interact via events. Example cooperative frameworks include customer interaction, order management, etc.

The *projection* of a cooperative framework onto a role or group of roles gives a view of that framework limited to interactions relevant to these roles. This allows an organisation to only concern itself with the aspects of a cooperative framework that are relevant to it, and ignore other aspects. This also allows a degree of privacy. Projections are used during the negotiation phase in order to express the level of commitment required by the company [Piccinelli et al. 2001]. DySCo is able to automatically generate projections from cooperative frameworks.

By using this layered approach, DySCo maintains a separation between internal and cooperative processes. A similar approach is taken in CrossFlow [Klingemann et al. 1999]. Other teams have adopted different approaches: Interworkflow [Hiramatsu et al. 1998] maps the cooperative process directly onto internal processes, while COSMOS [Griffel et al. 1998] uses a more centralised approach.

8. AGENTS AND HEWLETT-PACKARD

Agent technology is a key enabling component in bringing about Hewlett-Packard's e-services vision, and for that reason, HP Labs is making significant investments in research in this area. We have described our work on Agent-mediated B2B ecommerce above. There is much more agent-oriented work within Hewlett-Packard labs, which we can only touch on briefly. Further work in Bristol includes distributed agent-based personalization, and agent-based trading of Internet bandwidth. HP Labs Palo Alto has work on automated auction design, mobile agents, user profiling and FIPA standardization activities, among other things. For further information and access to our publications, see www.hpl.hp.com/agents.

9. ACNOWLEDGEMENTS

Thanks to the E-service markets team and Nick Jennings for help in compiling the information in this article. Thanks to Andrew Byde and Kayte Watts for help in preparing the manuscript.

REFERENCES

Anthony, P., Hall, W., Dang, V., and Jennings, N. 2001. Autonomous agents for participating in multiple on-line auctions. In *Proc. IJCAI Workshop on E-Business and the Intelligent Web* (2001).

Barbuceanu, M. and Lo, W. 2000. A multi-attribute utility theoretic negotiation architecture for electronic commerce. In *Proc. Fourth International Conference on Autonomous Agents* (2000).

Boubez, T. and al. 2000. UDDI data structure reference v1.0. In *UDDI Open Draft Specification* (2000).

Byde, A. 2001. A dynamic programming model for algorithm design in simultaneous auctions. In *Proc. WELCOM '01, Heidelberg, Germany*, LNCS (2001). Springer Verlag.

Cliff, D. and Bruten, J. 1998. Less than human: Simple adaptive trading agents for CDA markets. In *Proceedings of the 1998 Symposium on Computation in Economics, Finance, and Engineering: Economic Systems* (1998).

Das, R., Hanson, J., Kephart, J., and Tesauro, G. 2001. Agent-human interactions in the continuous double auction. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2001).

Dignum, F. and Weigand, H. 1995. Modelling communication between cooperative systems. In *Proceedings of the Conference on Advanced Information Systems Engineering (CAiSE'95)* (1995), pp. 140–153.

Faratin, P., Sierra, C., and Jennings, N. 1998. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems 3-4*, 24, 159–182.

Griffel, F., Boger, M., Weinreich, H., and Lamersdorf, W. 1998. Electronic contracting with cosmos how to establish, negotiate and execute contracts on the internet. In *Proc. 2nd International Enterprise Distributed Object Computing Workshop* (1998).

He, M., Leung, H., and Jennings, N. 2001. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. Submitted for publication.

Hendler, J. and McGuinness, D. 2000. The DARPA agent markup language. *IEEE Intelligent Systems 16*, 6, 67–73.

Hiramatsu, K., Okada, K., and Hayami, H. 1998. Interworkflow system: coordination of each workflow system among multiple organisations. In *Proc. 3rd International Conference on Cooperative Information Systems* (1998).

Hollingsworth, D. 1994. *The Workflow Reference Model.* The Workflow Management Coalition.

Horrocks, I. 1998. *FaCT Reference Manual Version 1.6.*

Jennings, N. R., Faratin, P., Johnson, M. J., O'Brien, P. O., and Wiegand, M. E. 1996. Using intelligent agents to manage business processes. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-96)* (April 1996), pp. 345–360.

KLINGEMANN, J., WASCH, J., AND ABERER, K. 1999. Adaptive outsourcing in cross-organisational workflows. In *Proceedings of the 11th International Conference on Advanced Information Systems Engineering* (1999).

LASSILA, O. AND SWICK, R. 1999. Resource description framework (RDF) model and syntax specification. *W3C Recommendations*.

MEYER, J. C. 1988. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic 29*, 1 (Winter), 109–136.

PICCINELLI, G. AND MOKRUSHIN, L. 2001. Dynamic e-service composition in DySCo. part of the 21st International Conference on Distributed Computing Systems (ICDCS-21).

PICCINELLI, G., PREIST, C., AND BARTOLINI, C. 2001. E-service composition: Supporting dynamic definition of process-oriented negotiation parameters. In *Proc. IEEE 2nd E-Negotiations Workshop* (2001). part of the 12th International Conference on Database and Expert Systems (DEXA).

PREIST, C. 1999. Commodity trading using an agent-based iterated double auction. In *Proc. Third International Conference on Autonomous Agents* (1999).

PREIST, C., BARTOLINI, C., AND PHILIPS, I. 2001. Algorithm design for agents which participate in multiple simultaneous auctions. In F. DIGNUM AND U. CORTES Eds., *Agent Mediated Electronic Commerce III*, Lecture Notes in AI. Springer Verlag.

PREIST, C., BYDE, A., AND BARTOLINI, C. 2001. Economic dynamics of agents in multiple auctions. To appear at Autonomous Agents 2001.

PREIST, C., BYDE, A., BARTOLINI, C., AND PICCINELLI, G. 2001. Towards agent-based service composition through negotiation in multiple auctions. In *ASIB'01 Symp. on Inf. Agents for Electronic Commerce* (2001).

PREIST, C. AND VAN TOL, M. 1998. Adaptive agents in a persistent shout double auction. In *Proceedings of the 1st International Conference on the Internet, Computing and Economics* (1998). ACM Press.

REYNOLDS, F. 2001. An RDF framework for resource discovery. In *Proceedings of the Second International Workshop on the Semantic Web* (2001).

TRASTOUR, D., BARTOLINI, C., AND GONZALEZ-CASTILLO, J. 2001b. Description logics for matchmaking of services. In *Proceedings of the KI2001 Workshop on Applications of Description Logic* (2001).

TRASTOUR, D., BARTOLINI, C., AND GONZALEZ-CASTILLO, J. 2001a. A semantic web approach to service description for matchmaking of services. In *Proceedings of the International Semantic Web Workshop on Infrastructure and Applications for the Semantic Web* (2001).