A Market Architecture for Multi-Agent Contracting

John Collins, Scott Jamison, Bamshad Mobasher, Maria Gini

Department of Computer Science
University of Minnesota, Minneapolis, MN 55455

{jamison, jcollins, mobasher, gini}@cs.umn.edu

Abstract. We present a generalized market architecture that provides support for a variety of types of transactions, from simple buying and selling of goods and services to complex multi-agent contract negotiations. This architecture is organized around three basic components: the *exchange*, the *market*, and the *session*. We also present a negotiation protocol for planning by contracting that takes advantage of the services of the market. We show how the existence of an appropriate market infrastructure can add value to a multi-agent contracting protocol by controlling fraud and discouraging counterspeculation.

1 Introduction

In recent years, many researchers and practitioners have focused on the design of market architectures for electronic commerce, and on protocols governing the interaction of self-interested agents engaged in such transactions. While providing support for direct agent negotiation, the existing architectures for multi-agent virtual markets usually lack explicit facilities and infrastructure for handling multiple and varied negotiation protocols. Since existing market architectures do not provide such protocols as an integrated part of the framework, they will have to be extended in order to provide such support.

Our goal in this research is to design a generalized multi-agent market architecture that can provide explicit and integrated support for complex agent interactions, such as in automated contracting, as well as other types of negotiation protocols, including sealed-bid auctions and open-bid or advertised-price buying and selling.

In addition, we show how the existence of an independent market infrastructure can add value and practicality to contracting protocols, such as providing protection against fraud and misrepresentation, and in curtailing unproductive value-based or time-based counterspeculation by participating agents. We also introduce a flexible contracting protocol which can take full advantage of the proposed market architecture to facilitate agent interactions.

Automated contracting protocols that have been developed recently generally assume direct agent-to-agent negotiation, and preclude the existence of an independent market infrastructure that can affect the timing and/or functionality of the protocol elements.

Smith [16] pioneered research in communication among distributed agents with the Contract Net protocol. The Contract Net, which was designed for cooperative agents, has been extended by Sandholm [13] to self-interested agents. Rosenschein and Zlotkin [12] analyze a variety of domains and propose a classification of problems into domains that are characterized by different types of negotiation among agents. They show that the behavior of multiple, interacting agents can be influenced by the set of rules (the protocol) that the system designers choose for the agents' environment. The purpose of these rules is to allow the agents to make constructive agreements. Their analysis assumes that the negotiating agents have similar capabilities. The protocol we present in this paper does not require that assumption.

Mechanisms to reduce counterspeculation, such as the Clarke tax mechanism [2] or the Vickrey auction [19] have been proposed for automated negotiation of self-interested agents. The architecture we present can support the Vickrey auction, and eliminates one of its limitations by providing a structure that can act as a trusted auctioneer [14].

A variety of architectures have been proposed for single and multiple agents in different domains (see, for instance, [7], [9], [11]). MAGMA [18], an open architecture for agents interested in buying and selling, supports both manual and automated negotiation with a limited form of Vickrey auction. Even though MAGMA already includes many of the features of the architecture we present here, MAGMA is intended for a more limited domain. Substantial work is underway in standardizing an open architecture for electronic commerce [10, 17]. Our architecture improves on these proposals by adding support for more complex negotiation protocols. Our architecture could be implemented by extending the framework presented in [17].

Existing architectures are generally designed for the kind of commercial activity that involves buying and selling of physical or electronic goods over a distributed electronic environment such as the Internet. They do not explicitly support more complex interactions such as those in a contracting domain where customer agents formulate plans and use the negotiation process to gain commitment from multiple supplier agents for the execution of these plans. For example, this type of interaction would be useful in multi-enterprise manufacturing.

This paper is organized as follows: In Section 2 we identify the requirements for a generalized multi-agent market architecture that can support complex agent interactions, and we present a novel architecture that satisfies these requirements. In section 3, we show how this architecture can be applied to a multi-agent contracting domain, and we present a flexible contracting protocol that can take full advantage of our proposed architecture. In section 4, we conclude and discuss further research opportunities.

2 A Generalized Multi-Agent Market Architecture

The architecture we propose is a distributed set of objects that can support electronic commerce in a variety of domains, from the simple buying and selling of goods to situations that require complex multi-agent negotiation and contracting. We begin by discussing a set of requirements that this architecture is intended to address, and then turn to a description of our proposal.

2.1 Requirements for a Market Architecture

The architecture should provide a framework for secure and reliable commerce among self-interested agents. Such a framework must satisfy a set of requirements including the following:

- Provide support for a variety of transaction types, including simple buying and selling, auctions, and complex multi-agent contract negotiation and execution.
- Effectively control fraud and misrepresentation.
- Discourage counterspeculation.
- Provide for secure and private credit and payment mechanisms.
- Provide a language in which the rich array of semantic content about commerce can be expressed. This is called the *profile service* in [17].
- Provide for robust exception handling.
- Scale smoothly from local to world-wide in scope.
- Be extensible, by third parties, in well-defined and interesting ways.
- Interoperate with other new and existing Electronic Commerce services.

We now focus on three of these general requirements that are particularly important to contracting domains, and that are not adequately covered in other market architecture proposals [10, 17].

Support for complex multi-agent negotiation

Protocols that require agents to do complex marginal cost computations [15, 16], or plan generation and composition [1, 12], may require extended periods of time to complete, during which a context must be maintained. The time during which the negotiated transaction extends can also span significant periods of time, at least in the range of weeks to months. In order to support this, the market infrastructure must maintain the state of each transaction over time. This is a prerequisite for many other functions we expect the market to perform, and it makes the system more robust in the face of hardware and communication failures. Most negotiation protocols involve time limits, such as a deadline for receipt of bids. All parties to a time-sensitive negotiation process must have a common time reference. We expect the market architecture to provide this. The architecture must also have the ability to validate non-performance and assess negotiated decommitment penalties.

Protection against fraud and misrepresentation

We must assume that participating agents will take advantage of any opportunities that exist in the design of the market to gain advantage. The structure of the market must recognize and protect against situations that allow agents to gain unfair advantage at the expense of other agents. Strategies that can result in this type of "unfair" gain, and related architectural requirements, include:

- Hiding one's identity or taking on the identity of another — the architecture must provide for secure identification of parties involved in transactions. This is part of the Certificate Services of [17].
- Dishonest auctioneer — In Vickrey-type auctions [20, 19], the motivation for truth-telling on the part of participants is predicated on their belief in the honesty of the auctioneer. The architecture should provide either a trusted auctioneer or an auditing service for such situations. This is an example of a negotiating situation that makes a market structure, or at least a trusted intermediary or third party, necessary.
- Miscommunication of the rules under which an auction is being conducted — By miscommunicating the timing parameters or price-setting rules under which an auction is being run, a customer can attempt to manipulate the behavior of suppliers. Such miscommunication should be prevented.
- Failure to follow through on commitments — the system must provide for credit checks, banking services, secure payment facilities, and audit trails. In addition, participants should be prevented from gaining advantage by simply turning their machines off when a contract becomes disadvantageous.

Discouragement of counterspeculation

Opportunities for counterspeculation arise when the rules of negotiation allow agents to gain advantage by making use of factors other than their own capabilities and valuations, such as their estimates of the capabilities and valuations of the customers or other suppliers [8]. We are concerned with two general types of counterspeculation. Value-based counterspeculation [12, 15, 20] occurs when agents use their own estimates of each other's valuations to set bid prices. In [1], we identified two classes of time-based counterspeculation opportunities in a contracting domain that can be controlled by the settings of certain timing parameters. One of these situations occurs when supplier agents are allowed to expire their bids before the customer's call-for-bids expires, and the other situation occurs when customers are perceived by suppliers to be considering bids and formulating plans before bidding is closed.

Mechanisms such as Vickrey auctions, and control of the timing of negotiation interactions, can minimize these opportunities and should be available in the market architecture. In particular, the market architecture must support robust auction protocols such as the Vickrey mechanism [20] that resist value-based counterspeculation. The system should also have the ability to control the timing of negotiation events, such as ensuring that customers do not see bids until after the bid deadline has passed, in order to discourage suppliers from speculating about the customer's processing capabilities [1].

We now turn to the architectural design of a market structure that will satisfy these requirements. The fundamental elements of this architecture are the *exchange*, the *market*, and the *market session*.

2.2 The Exchange

An *exchange* is a collection of domain-specific markets in which goods and services are traded, along with some generic services required by all markets, such as verifying identities of participants in a transaction, or a Better Business Bureau that can provide information about the reliability of other agents based on past performance. Architecturally, an exchange is a network-accessible resource that supports a set of markets and common services.

Specific functions provided to agents by the exchange through the API layer include:

- Register a new participant (customer or supplier) with the exchange registry.
- Register a new market with the exchange.
- Return a list of markets contained in the exchange. Markets are self-describing.
- Search the exchange for markets that handle specific commodities or services.
- Provide secure credit and payment facilities.
- Access and update transaction records in the Better Business Bureau.

The general structure of an exchange is depicted in Figure 1.

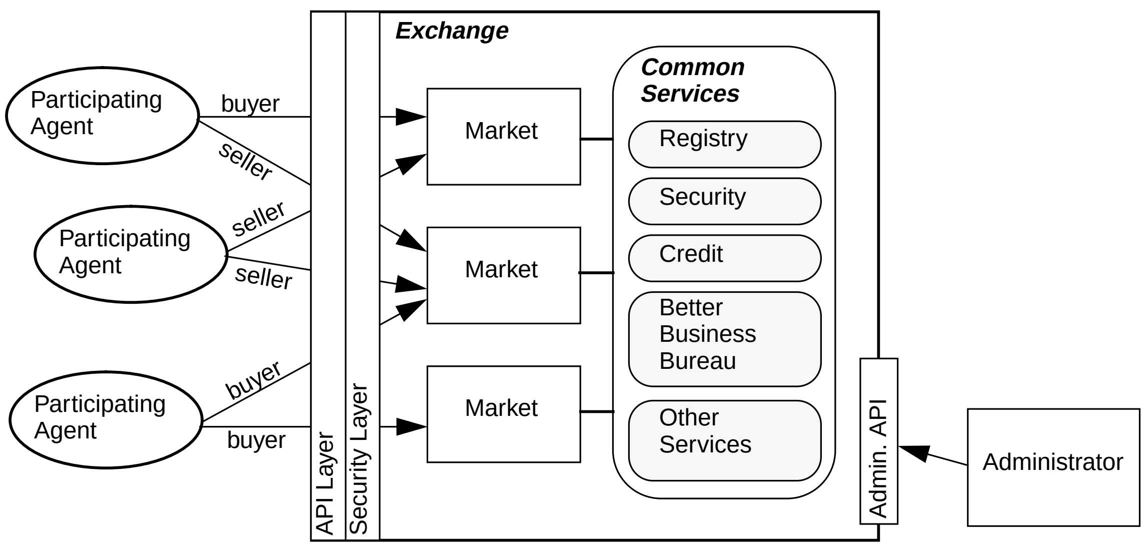

Figure 1. Exchange Structure

2.3 Markets

Each *market* within an exchange is a forum for commerce in a particular commodity or business area. There would be markets devoted to banking, publishing and printing, construction, transportation, industrial equipment, etc. Each market includes a set of domain-specific services and facilities, as shown in Figure 2, and each market draws upon the common services of the exchange.

An important component of each market is a set of current market sessions in which the actual agent interactions occur. Agents participating in a market may do so as either session initiators, or as clients, or both. As detailed in the next section, each session is initiated by a single agent for a particular purpose, and in general multiple agents may join an existing session as clients.

Important elements of the market include:

- An *Ontology* that is specific to the domain of the market, specifying the terms of discourse within that domain. In a commodity-oriented domain, it would include terms for the products or services within the domain, as well as terminology for quality, quantity, features, terms and conditions of business, etc. In a planning-oriented domain, specifications of services would be in a form that supports planning, such as Strips-type operators [3]. The underlying language structure could be KQML [4] or an equivalent knowledge-encoding language. Ontolingua [5] could be used to develop domain specific ontologies and to translate them into application oriented representation languages, such as CORBA Interface Definition Language.
- A *Protocol Specification* that formalizes the types of negotiation supported within the market. Within a planning-oriented market domain, these specifications would be limits on parameters of the negotiation protocol outlined in Section 3, such as the maximum decommitment protocol, whether bids can be awarded before the bid deadline, etc.
- A *Registry* of market clients who have expressed interest in doing business in the market. Entries in this registry would include the identity of a client, a catalog [6] (or a method for accessing a catalog) of that client's interests, products or capabilities, which can be used to locate clients to meet requests for new session participants, and a client agent that is empowered to negotiate contracts on behalf of the supplier. Client catalogs are required to express their interests and offerings in terms of the market's ontology.

The market also provides facilities that allow agents to register themselves as participants, request updates to the ontology, initiate new market sessions, retrieve a list of sessions initiated by that agent, retrieve a list of sessions of which the agent is a client, and retrieve a list of sessions that are open to new clients.

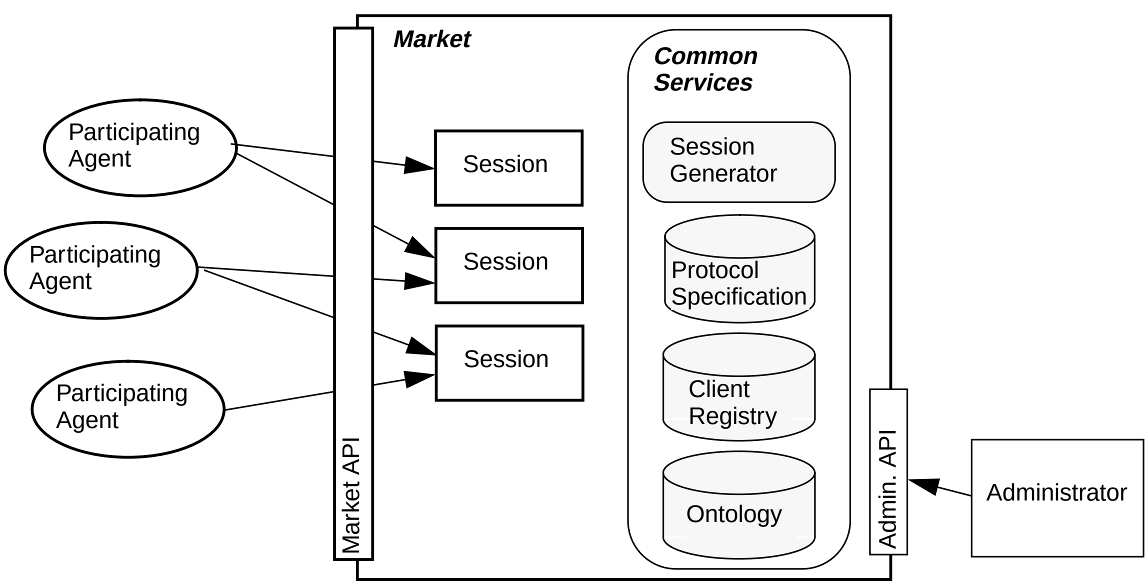

Figure 2. Market Structure

2.4 Market Sessions

A *market session* (or simply a *session*) is the vehicle through which market services are delivered dynamically to participating agents. It serves as an encapsulation for a transaction in the market, as well as a persistent repository for the current state of the transaction. We have chosen the term "session" to emphasize the temporally extended nature of many of these interactions. For example, in a contracting market, if an agent wishes to build a new house, it initiates a session and issues a call-for-bids. The session extends from the initial call-for-bids through the negotiation, awards, construction work, the paying of bills, and the final closing. In other words, the session encloses the full life of a contract or a set of related contracts. The session mechanism ensures continuity of partially-completed transactions, protects against fraud, limits counterspeculation, and relieves the participating agents from having to keep track of detailed negotiation status themselves.

Agents can play two different roles with respect to any given session. The agent who initiates a session is known as the *session initiator*, while other participating agents are known as *session clients*. A session can be initiated either for the purpose of buying or selling, depending on the type of market. In the above example of building a house, the initiating agent was the buyer or customer, and the other participants would be sellers or suppliers, whether they were supplying materials, labor, advice, credit, or other services. A session could also be initiated to sell items or services at auction.

At any given time, a session can be *open* to new participants, or *closed*. A public auction would typically be open to new participants, while the house-building session described above would be closed once the contracts were let. The market maintains a list of open sessions which may be accessed by participating agents.

Figure 3 shows the structure of a session. Two APIs are exposed, one for the session initiator and one for session clients. Each session contains an Initiator Proxy that implements the Initiator API and persistently stores the current state of the session from the standpoint of the initiator. A Client Proxy is provided for each client that similarly provides a Client API to the client agent, and persistently stores the current state of the session from the standpoint of the client. Proxies are market entities that act on behalf of the agents and enforce market rules. There are two reasons for the existence of the proxy components. The first is related to security: client proxy components cannot see the private data of the initiator or of other clients. The second is that in a distributed system environment, the processing and persistent data elements of the initiator and clients would presumably be at different locations in the network to maximize performance.

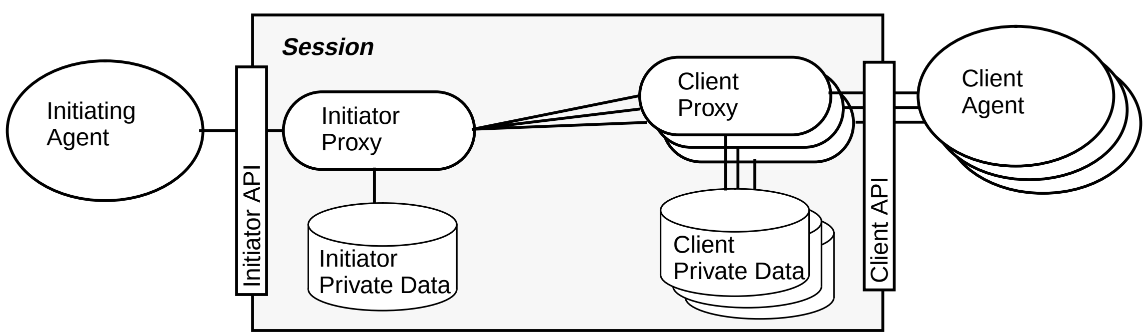

Figure 3. Session Structure.

Sessions can serve different purposes in different markets. For example, in a banking market, a session can represent a line of credit. That session, as long as it is active (meaning some amount of credit is still available), serves as a "credit token" in order to establish sessions in other markets. In general, session establishment proceeds as follows:

1. Initiator determines a need for a session, finds an appropriate market through the exchange.

2. Initiator confirms that needed services are available through the market by inspecting the Ontology.
3. Initiator requests a session from the market, providing identity, proof of credit (if required by the market), and negotiation protocol parameters. These parameters must be within the limits set by the market; for example, a market that allows only sealed-bid auctions would not permit a negotiation protocol that proposed consideration of bids before the bidding deadline.
4. Market checks the identity and credit of initiator.
5. Market grants session to initiator.
6. Clients join the session, and initiator and clients transact business through the session.
7. Once business is concluded, the initiator requests that the session be terminated.

Once the initiator has access to a session, it can conduct business in that market. The nature of the business and the range of semantics of a session are market-dependent. Because the session is contained within the market, and because it maintains, independently of the initiator and any clients, a persistent record of the activities encompassed by the session, it is able to perform many of the roles outlined above in Section 2.1. In particular, it can discourage value-based counterspeculation by acting as trusted auctioneer, it can guarantee that all clients are notified of the correct auction and negotiation protocol and protocol parameters, it can record commitments made by all parties and, within the limits of its enforcement powers, ensure performance against those commitments, and it can provide the necessary time reference and protocol timing control to discourage time-based counterspeculation.

3 A Contracting Market Structure

A primary motivation for the session construct is to support *Planning by Contracting* [1], an activity in which an agent, in order to formulate its plans and fulfill its goals, must contract with other self-interested agents for all or part of the necessary tasks.

3.1 Protocol for Planning by Contracting

The Planning by Contracting protocol is a three step process which begins after the session has been initiated by a customer agent: the customer issues a call-for-bids, suppliers reply with bids, and the customer accepts the bids it chooses with bid-accept messages. We have avoided the need for open-ended negotiation by means of bid break downs and a time-based decommitment penalty as described below.

Call for bids

Once the customer has developed a plan of subtasks chosen from the market’s ontology, it will send a call-for-bids message. The call-for-bids message will include, for each subtask listed, a time window during which the work must be done. The call-for-bids message will also include, among other information:

1. a bid deadline, or the time by which the suppliers must respond with bids,
2. the time at which the customer will begin considering the bids,
3. the earliest time at which bid acceptances will be sent, and
4. penalty functions for each subtask, which will be assessed against the supplier if the supplier commits to work, but fails (or decides not) to do it. These penalty functions are piecewise-linear functions of time that are intended to encourage suppliers to perform the work they commit to. If a supplier is unable to perform, the increasing value of the penalty function encourages it to explicitly decommit as early as possible.

This call-for-bids message, once created, is passed to the market session, which makes it available to all of the appropriate suppliers (those who are registered with the market, and are able to perform the necessary tasks.) In this sense, the call-for-bids message is public, while all of the remaining messages are private. Before forwarding it, the market session may check the message to make sure that it conforms to all market and exchange rules which may exist.

Bidding

Each supplier will inspect the call-for-bids, and will decide whether or not it should respond with a bid, according to its resources, time constraints, and knowledge of the work to be done, according to the catalog of services provided by the market agent. If it chooses to respond, it will send a bid message, which will be private (i.e. other suppliers will not see the contents of the bid). This bid message can include a combination of subtasks, which must be a subset of the subtasks listed in the call-for-bids. The content and number of bid messages will be monitored and may be recorded by the market session, before they are validated and forwarded to the customer.

In the bid, the supplier must indicate the cost (to the customer), the time window, and the estimated duration of the work for the whole subtask combination, and this same data for each of the separate subtasks (please see the explanation for this in the next section.) The bid-accept deadline must also be included, as well as a penalty function for each subtask which the customer will have to pay if it commits to giving this supplier the work but then decides to decommit. This penalty function will have the same structure as the supplier penalty function. Each supplier can send multiple bids for each call-for-bids, each including different costs and time windows, but each supplier will be awarded only one bid combination (or part of one). This is to enable the supplier to send many bids, but not over commit itself.

This bid is a commitment by the supplier to do work listed in the bid, should the customer accept it. If the supplier sends no bid message before the customer's bid deadline, the customer will assume that the supplier has decided not to send a bid for this particular call for bids. Thus, rejection is passive.

Bid acceptance

Having received the bids, the customer must decide which of the bids to accept, using knowledge about the bids, the task and subtask values, its own time constraints and the bidder (perhaps provided by the market agent). After completing this process, the customer must decide to do one of three things for each bid that it has received:

1. accept the whole bid,
2. accept a subset of the subtasks in the bid, or
3. reject the bid (again, this is done passively).

The motivation for these choices is to make open-ended negotiation unnecessary. If no acceptable set of bids together would cover every subtask to the satisfaction of the customer, then the customer can avoid negotiation because it knows how the supplier will break down the costs of the accepted subtasks, should it become necessary for the customer to accept a subset of the original bid combination. This scheme in conjunction with the time-based decommitment penalty functions make it possible to avoid open-ended negotiation without loss of generality.

The bid-accept message will be sent through the market session, which will verify, validate and time-stamp it before forwarding it to the customer. Note that either of the first two choices are commitments to give the supplier the work and at the point in time that this message is sent (according to the market session's time stamp), both the supplier and the customer penalty functions will be set into effect. A failure to send a bid-accept message means the customer is rejecting the supplier's bid.

Once commitments have been made, an agent may determine that it cannot do the tasks it has committed to, or that it would disadvantageous to do so. In these situations, the agent must send a decommitment message to the other agent, describing what parts of its commitment it will not be satisfying. Included in the decommitment messages will be an acknowledgment of the penalty that the agent will be paying as a result of the decommitment.

3.2 The Role of the Market in Planning by Contracting

We present a brief outline of the activities of customer and supplier agents engaged in planning by contracting:

1. Using the ontology provided by the market, a customer agent develops a partial plan, assigns a value to its goal, and estimates tentative values for plan components based on the goal value and the "criticality" of each component.
2. The customer initiates a session in the market.
3. The customer announces, through the session, tasks to be bid upon in a call-for-bids.
4. Suppliers join the session as clients, either because they have been notified by the market or because they have queried the market and found a session of interest.
5. Suppliers evaluate the call-for-bids, decide which subtasks to bid upon, and return bids through the session.
6. The customer receives bids from suppliers.
7. For each bid, the customer maps the bid to the plan and evaluates bid price vs. value. The value calculation could include time limits, decommitment penalty, task coverage, location of work, identity of bidder, etc.
8. The customer extends or modifies the plan, and may announce further bids.
9. The customer awards contracts to selected suppliers.
10. The customer monitors execution of the plan, making necessary adjustments as suppliers perform or fail to perform their contracted tasks.
11. Customer and suppliers finalize financial settlements and the session is terminated.

Within the market architecture outlined in this paper, these activities are encapsulated in a market session. Once a session is obtained, the customer is able to issue a call-for-bids and conduct other business. The interactions involved in the basic bidding cycle among the customer, supplier, and market session are illustrated in Figure 4. The customer's call-for-bids is passed to the session rather than directly to the supplier agents, and all interaction passes through the session. This allows the session to perform several important services:

- The customer need not search the market to find suppliers who might be interested in bidding. The session does this by searching the market's registry to find suppliers for which the intersection of services or goods requested in the call-for-bids and the advertised goods and services of the supplier is non-empty. The customer can override this with its own list, or it may retrieve the list from the market ahead of time and prune it. If the customer supplies the list, it has the option of specifying that the session be closed to other suppliers.
- The customer need not find a way to describe the services that it requires. It can use the catalog of services (and its syntax) which the market has listed in the ontology.
- The session time-stamps all interactions involved in the transaction, in order to avoid dispute among customers and suppliers over performance or non-performance with respect to deadlines.

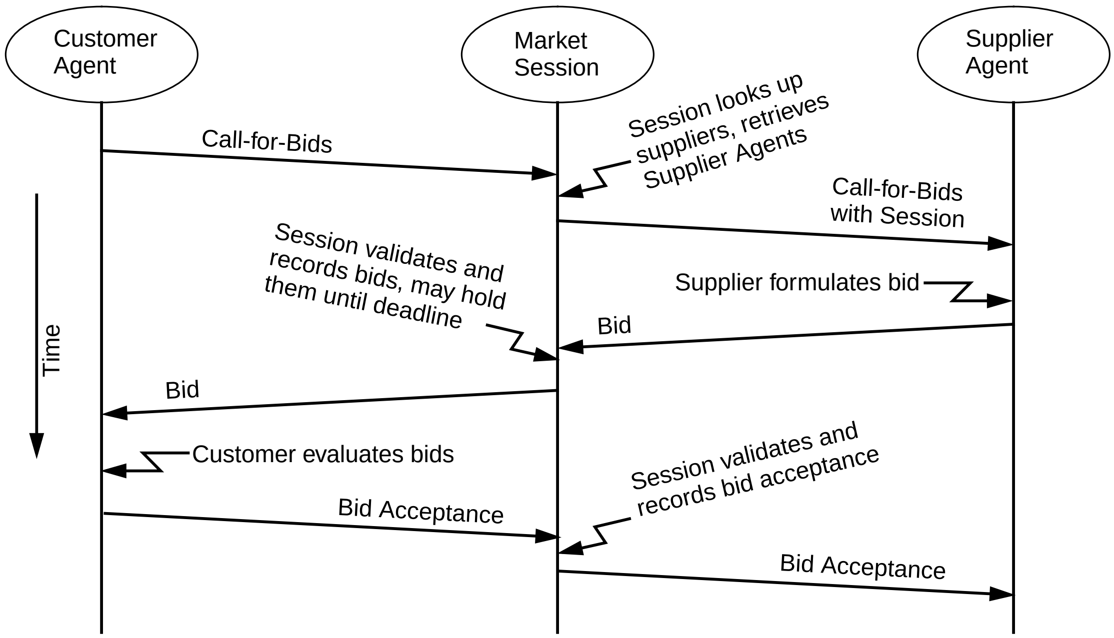

Figure 4. Session-Mediated Negotiation

- The session truthfully informs the suppliers of the conditions under which the bidding is being conducted, and then enforces those conditions. This eliminates the possibility of the customer misinforming the suppliers of bid conditions, and therefore unfairly affecting their pricing strategies.
- The session can limit the number of bids sent by each supplier to avoid overwhelming the customer with choices.
- If the customer desires or the market allows, the session may provide the customer with information about the suppliers for use in choosing which bids to accept.
- The session enforces the "rules of the market" with respect to deadlines, penalties, disclosure of identity, and auction rules. For example, if a sealed-bid auction is advertised, the session will hold bids back from the customer until the bid deadline has passed. This eliminates the possibility of temporal counterspeculation as identified in [1].
- If the customer desires or the market requires, the session can act as a "trusted auctioneer" in a Vickrey auction.

3.3 Example

We conclude this discussion with an example. A startup company, Acme Software and Screen Doors, Inc., is releasing version 1.0 of its new killer application. The product manager, who is also the president and chief programmer, has a goal: produce and ship 1000 shrink-wrap packages, including media, manuals, license forms, registration cards, and license key stickers. The old method would be to find a broker or publisher, and hand over the job.

Now assume we have an intelligent agent that can act as a broker in the Global Exchange. The manager gives the goal to her automated agent, which enters the exchange to assemble a set of contracts to satisfy the goals. We will assume the agent is already registered with the exchange. The agent searches the exchange and finds three markets that provide services to satisfy its goals. One deals with printing services, one with digital media, and one with packaging and shipping. Using operators obtained from the ontologies in the selected markets, the agent formulates a plan to satisfy its goals. Its plan looks like Figure 5.

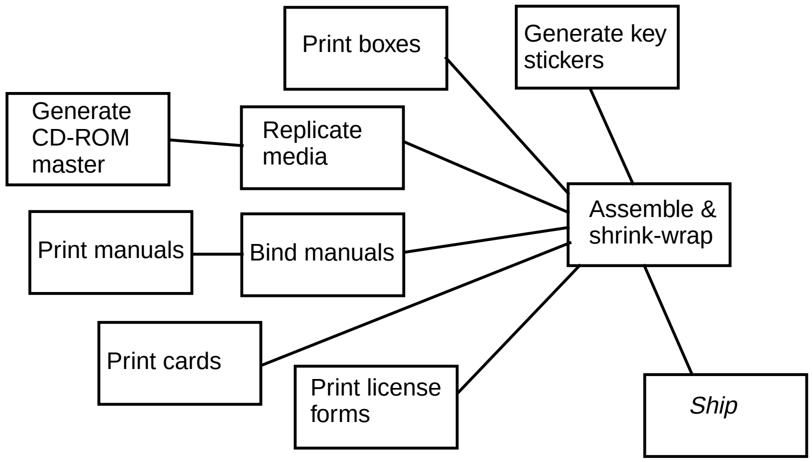

Figure 5. Plan for release 1.0 production

Next, the agent establishes the necessary credit with exchange, as specified by the chosen markets, and requests contract negotiation sessions in all 3 markets. The plan is now broken down by market, and the agent submits a call-for-bids to each session. It is acceptable if the submitted subplans overlap. This is because some potential contractors may give lower bids on combinations of multiple tasks than they would on single tasks.

Each session now invites potential contractors who have expressed interest, through the market registry, in the subject matter of the bids, to join the sessions as clients. Some of those contractors submit bids.

The customer agent receives the bids, and attempts to assemble them into an optimal feasible plan. Unfortunately, no supplier has bid on printing the license key stickers. At this point, the customer agent must decide whether its own resources can satisfy this need, whether the plan can succeed without the stickers, or whether to risk starting plan execution on the assumption that a supplier for the stickers can be found before plan execution reaches the point where they are needed. It decides (or is told by our manager) that the stickers can be done in-house.

The customer agent now awards the selected bids, and work commences. Part way through the process, the box printer backs out, and pays the required decommitment penalty. Our agent re-opens bidding for box printing, receives and awards a bid.

The customer agent monitors task progress, posting payments as agreed at task or subtask completion. Once all transactions are completed, the customer agent requests the respective markets to terminate the sessions.

4 Conclusions and Future Work

In this paper we have brought together ideas from recent work in market architectures for electronic commerce, and work in multi-agent contracting protocols. We have presented a generalized market architecture that provides support for a variety of transaction types, from simple buying and selling to complex multi-agent contract negotiations. We have also presented a protocol that takes advantage of the services of the market. Our market architecture is organized around three basic components: the *exchange*, the *market*, and the *session*. We have shown how the existence of an appropriate market infrastructure can add value to a multi-agent contracting protocol by controlling fraud and discouraging counterspeculation.

This work raises several interesting questions for future research. A game-theoretic analysis of the protocol could be done to determine optimal strategies for its use by agents. In particular, how should the decommitment penalties be used, and how should proposed decommitment functions be evaluated when computing marginal costs of plan alternatives. The methods for composition of possibly overlapping bids into feasible or even optimal plans must be worked out. Finally, the architecture we have presented should be implemented and tested in the real world.

References

[1] J. Collins, S. Jamison, M. Gini, and B. Mobasher, Temporal Strategies in a Multi-Agent Contracting Protocol, to appear in *Proceedings of the AAAI-97 Workshop on AI in Electronic Commerce*, Providence, RI, 1997.

[2] E. Ephrati and J. Rosenschein, Deriving consensus in multiagent systems, *Artificial Intelligence*, Vol 87, 1996, 21-74.

[3] R. Fikes and N. Nilsson, STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence*, Vol 2 (3), 1971, 189-208.

[4] T. Finin and R. Fritzson, KQML - A Language and Protocol for Knowledge and Information Exchange, University of Maryland, Baltimore, MD, 1994.

[5] T. R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition*, Vol 5 (2), 1993, 199-220.

[6] A. M. Keller, Smart Catalogs and Virtual Catalogs, in *Readings in Electronic Commerce*, ed. R. Kalakota and A. Whinston, Addison-Wesley, Reading, Mass., 1996.

[7] D. Kinny and M. Georgeff, Modeling and Design of Multi-Agent Systems, in: *Intelligent Agents III*, J.P. Müller, M.J. Wooldridge and N.R. Jennings, eds., Lecture Notes in Artificial Intelligence, Springer, Berlin, 1997.

[8] D. M. Kreps, A *Course in Microeconomic Theory*, Princeton University Press, 1990.

[9] M. Lejter and T. Dean, A Framework for the Development of Multi-Agent Architectures, in: *IEEE Expert*, 1996.

[10] S. McConnell, M. Merz, L. Maesano and M. Witthaut, An Open Architecture for Electronic Commerce, Object Management Group, Cambridge, Massachussetts, 1997.

[11] J. A. Rodriguez, F. Noriega, C. Sierra, and J. and Padget, FM96.5 - A Java-based Electronic Auction House, in: *Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology* (PAAM'97), London, April 1997.

[12] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter*, MIT Press, Cambridge, MA, 1994.

[13] T. W. Sandholm and V. R. Lesser, Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework, in: *Proceedings First International Conference on Multi-Agent Systems*, San Francisco, CA, (1995), 328-335.

[14] T. W. Sandholm, Limitations of the Vickrey Auction in Computational Multiagent Systems, in: *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, Keihanna Plaza, Kyoto, Japan, December 1996.

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

[16] R. G. Smith, The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver, in: IEEE Transactions on Computers C-29 (12), December 1980, 1104-1113.

[17] J. M. Tennenbaum, T. S. Chowdhry and K. Hughes, eCo System: CommerceNet's Architectural Framework for Internet Commerce, Object Management Group, Cambridge, MA, 1997.

[18] M. Tsvetovatyy, B. Mobasher, M. Gini, and Z. Wieckowski, MAGMA: An Agent-Based Virtual Market for Electronic Commerce, *Journal of Applied Artificial Intelligence*, (to appear).

[19] H. Varian, Economic Mechanism Design for Computerized Agents, in: *Proceedings of the Usenix Workshop on Electronic Commerce*, New York, NY, July 1995.

[20] W. Vickrey, Counterspeculation, Auctions, and Competitive Sealed Tenders, *Journal of Finance* 16, 1961, 8-37.