A Conceptual Framework for Modeling Automated Negotiations in Multiagent Systems

Mohammad Reza Ayatollahzadeh Shirazi and Ahmad Abdollahzadeh Barfouroush

In fully automated e-negotiation all involved parties are software agents, so negotiation takes place in a multiagent system between software agents that have been developed as a computer system for automating tasks in a specific application domain. A multiagent system is a group of agents that interact and cooperate with each other to fulfill their objectives or to improve their performance. How do these agents negotiate with each other to manage their task interdependencies? What negotiation mechanisms are needed? These are important questions.

In this article, we present a conceptual framework for modeling and developing automated negotiation systems. This framework represents and specifies all the necessary concepts and entities for developing a negotiation system as well as the relationships among these concepts. This framework can also be used to model human negotiations scenarios for analyzing these types of negotiations and simulating them with multiagent systems. The work reported in this article is the first unified framework that represents all the needed elements for modeling and developing automated negotiation systems and existing relationships between them.

Mohammad Reza Ayatollahzadeh Shirazi is a member of the academic board in the computer engineering and IT department at Amirkabir University of Technology in Tehran. His e-mail address is ashirazi@dpco.net.

Ahmad Abdollahzadeh Barfouroush is an associate professor in the computer engineering and IT department of Amirkabir University of Technology in Tehran. His e-mail address is ahmad@ce.aut.ac.ir.

10.1111/j.1571-9979.2007.00166.x

Key words: negotiation, automated negotiation, agent, multiagent system.

Introduction

Negotiation is a decentralized decision-making process in which two or more parties confer with each other with a view to reach an agreement that will satisfy the parties' requirements under conditions of limited knowledge and conflicting interests. In this process, negotiators exchange, evaluate, and use information as a basis for decision making.

In some domains, negotiation can be a complex, error prone, and ambiguous activity often performed under time pressure. Negotiations can occur frequently, almost continuously, in certain contexts, for example, in order to schedule network capacity.

Electronic negotiation (e-negotiation) can help to manage such complexity, facilitate negotiation processes, conduct checks, offer decision support, enhance efficiency, and reduce transaction costs. E-negotiation systems can also offer additional features not available through traditional methods of negotiation, such as decision analysis, graph support, and communication management.

We define a fully automated e-negotiation as one in which all parties involved are software agents (or programs), a semiautomated e-negotiation occurs when a human negotiates with a software agent, and manual e-negotiation takes place when all parties are human but their negotiation is supported electronically (Benyoucef and Rinderle 2005).

Fully automated e-negotiation typically takes place in a multiagent system that has been developed as a computer system for automating tasks in such application domains as information management, e-commerce, and manufacturing. In this case, the agent acts on behalf of other persons or organizations and interacts with other agents. In this setting, people interact with software agents to specify their needs, interests, policies, and priorities and to specify some of the negotiation parameters for the agents. Figure- One shows how interaction among people and agents takes place.

Multiagent systems can also be developed for simulation and testing purposes in which fully automated e-negotiation takes place among agents with the purpose of investigating the behavior of a real negotiation process among human negotiators. (Game-theoretic approaches to negotiation in multiagent systems, for example, can guide the design of the interaction mechanisms among agents and provide power insights into the problem of negotiation among participants.) In this case, people work with the system to observe the progress and behavior of the negotiation process and to modify the parameters in order to analyze the effect of the changes on the negotiation process and result. Figure Two shows how this interaction takes place

Figure One
A Typical Interaction between Human User and Software Agents that Act on Behalf of User

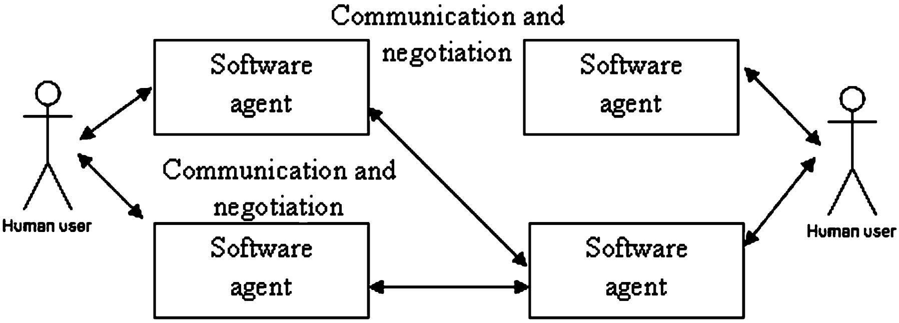
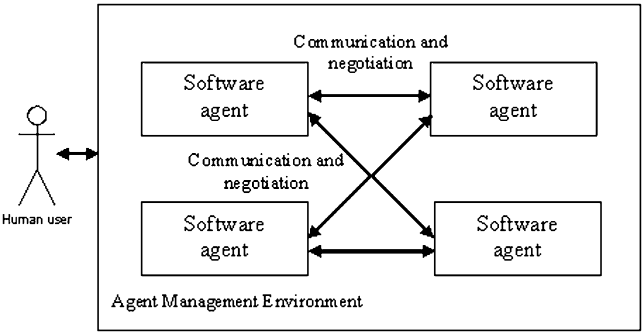


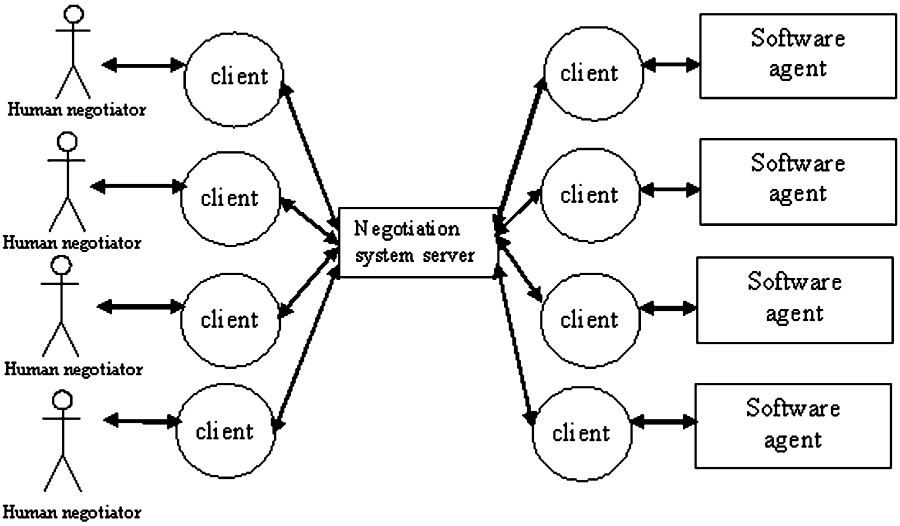
Figure Two
Interaction between Human User and the Environment that Manages Software Agents


Communication and negotiation
Software agent
Software agent
Human user
Communication and negotiation
Software agent
Software agent
Agent Management Environment

between the human user as a researcher or experimenter and the environment that manages the multiagent system for simulation and testing purposes.

In *semiautomated e-negotiation*, a human negotiates with one or more software agents according to a specified communication language, negotiation ontology, and negotiation protocol. (In information science, an ontology is defined as a data model that represents a set of concepts within a domain and the relationships between those concepts.) Each person has a client that permits its user to manage the negotiation process. The user can enter/exit the negotiation, send the offers, receive responses to the

Figure Three
Interaction between Human and Software Agent in a Semiautomated E-Negotiation System



offers, view the negotiation history and output, and see his/her assigned parameters. Figure Three shows how human negotiators and software agents interact in a semiautomated e-negotiation.

In all e-negotiation settings, the software agents perform the tasks necessary to achieve their design objectives. Typically, these agents are designed to perform some activity (e.g., a business activity such as trading) for their stakeholders (e.g., organizations, human negotiators). For example, in a multiagent system that has been designed to support a business activity such as an auction, the main role of agents can be ontology translation, matchmaking, network service provision, or anything else, and these agents will charge a fee for their goods or services and will negotiate both as buyers and as sellers with other agents or humans. These agents are economically intelligent and capable of making effective decisions about pricing, purchasing, or bidding.

We can say that e-negotiation systems have the following benefits and applications:

- supporting negotiation processes in real applications,
- improving quality of negotiation output (e.g., arriving at more satisfying agreements),

- enhancing the of study of the humanistic, social, and technical aspects of negotiations,
- providing needed infrastructure and platforms for teaching, training and self-learning in a negotiation domain.

Based on our survey of the literature, we define an "agent" as an autonomous system that interacts with its environment in order to satisfy its design agenda. A multiagent system is a group of agents or agent organizations that interact and cooperate with each other to fulfill their objectives or improve their performance.

Agent-based solutions are now used in industrial, commercial, medical, networking, and educational application domains. In most of these systems, agents interact and negotiate with each other. Also, agent-based solutions are used in modeling and simulating real human negotiation scenarios to analyze human negotiations processes and results and to investigate the behavior of human negotiators in complex negotiation scenarios. Furthermore, agent technology is a multidisciplinary field that interacts with other fields of science and technology, and the design of many elements of multiagent negotiation systems are based on techniques and methods developed in such disciplines as management, sociology, linguistics, and psychology.

In semiautomated negotiation, when negotiation takes place between humans and software agents, the software agent should be capable of interacting with the human negotiator. In automated e-negotiation systems, the agents are designed to negotiate with each other to reach agreements on scare resources, obtain needed resources for performing their tasks, and/or to manage task interdependencies and resolve possible conflicts (Abdollahzadeh Barfouroush, Masoumi, and Ayatollahzadeh Shirazi 2006). The agents use negotiation protocols and strategies to facilitate information exchange, coordination, cooperation, and negotiation. In many multiagent solutions to real-world problems, we should design and implement an automated negotiation system for the existing agents in a multiagent system that enables them to negotiate with each other.

In this article, we present a conceptual framework that is built from a set of main concepts, entities, and information that is linked to an existing system of relationships. This framework specifies a preferred approach for developers of automated negotiation systems. The framework defines main concepts that should be considered by developers during specification and implementation of automated negotiation systems. The main concepts in this framework are *negotiation application domain*, *negotiation scenario*, *negotiation system, negotiation approach*, and *negotiation evaluation criteria*. We use this framework as a template for capturing and representing the requirements of a practical negotiation system and for guiding the development process of a negotiation system. We have used this framework in our laboratory, in our research and development activities, and in the area of multiagent and automated negotiation systems. We believe such a framework can serve as a medium for communication between analysts and developer(s) of a computational negotiation system. It can also be used to represent human negotiations scenarios that we want to analyze and simulate with a multiagent system.

Researchers in the field of automated negotiation have proposed similar frameworks. For example, Claudio Bartoloni and his colleagues at Hewlett Packard Laboratories have created a general framework for automated negotiation dedicated to market mechanisms (Bartolini, Preist, and Jennings 2002). They present an abstract architecture for the negotiation framework in which they focus on just the shared protocol, not the negotiation strategy.

Carles Sierra and his colleagues have devised a general framework for negotiation through argumentation (Sierra et al. 1998). In this framework, they use dialogical frameworks to define shared ontology, social relations, communication, and protocol. Also, they define mechanisms for argument evaluation and interpretation. These related works are application domain and/or approach specific, and the components of these models are used to implement real-world applications. Jin Baek Kim and his colleagues have developed a more related framework for negotiation process in dynamic e-business environments (Kim and Segev 2003). Their framework includes five components for constructing dynamic negotiation processes: negotiation requirements, negotiation structure, negotiation process, negotiation protocol, and negotiation strategy.

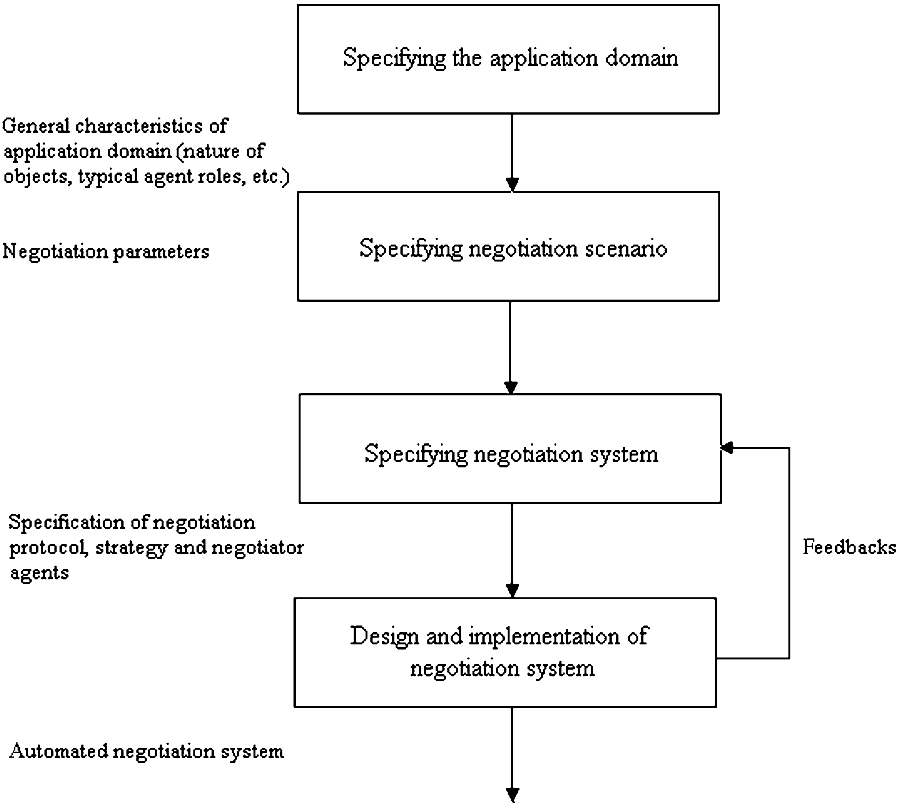
This article is the first work to analyze the relationships that exist between the general characteristics of application domain, negotiation scenario parameters, and elements of a negotiation system, and their relationships with each other in a single conceptual framework. We have used this framework as a reference model in our other research for defining various scenarios for evaluating a flexible negotiation system that we have designed in our laboratory (Ayatollahzadeh Shirazi and Abdollahzadeh Barfouroush 2006).

In this article, we first introduce a process for developing an automated negotiation system for multiagent systems, defining the main elements of this framework. We then describe how the concepts of the framework influence each other and show how the elements of a negotiation system are functions of the parameters of both the application domain and the specific negotiation scenario. Finally, we illustrate our framework using a distributed bargaining scenario between agents for a customer and a service provider in the domain of telecommunication service.

A Conceptual Framework for Automated Negotiation

As shown in Figure Four, the first step in developing a practical and computational negotiation system is to describe the application domain in

Figure Four
Main Steps for Developing a Negotiation System


Specifying the application domain
General characteristics of application domain (nature of objects, typical agent roles, etc.)
Negotiation parameters
Specifying negotiation scenario
Specifying negotiation system
Specification of negotiation protocol, strategy and negotiator agents
Feedbacks
Design and implementation of negotiation system
Automated negotiation system

which agents will negotiate with each other. Nowadays, multiagent systems are being used in an increasingly wide variety of industries and services, that is, for commercial, medical, networking, and educational applications. Each of these application domains has specific characteristics that will affect the design of an appropriate negotiation system for the multiagent system.

After defining the related characteristics that have implicit or explicit effects on the negotiation system, such as the negotiation objectives, a typical interaction scenario in the application domain, and the style of interaction, the developer should then define the specific parameters of a negotiation scenario. Next, the developer should specify the negotiation protocol and strategy as the basic elements of a negotiation system before its implementation.

Based on our literature reviews and experiments, we have captured the main concepts relevant to the developer of an automated negotiation system in a conceptual framework. The elements of this framework and the relationships among them are shown in Figure Five. As shown in this figure, the five main entities that we have identified are: negotiation application domain, negotiation scenario, negotiation system (which is composed of protocol, strategy, and agents), negotiation approach, and negotiation evaluation criteria. Each entity is defined by a set of parameters.

We have used this conceptual framework as a template for capturing and representing the requirements of a practical negotiation system in our development processes. It has served as a reference model and a medium for communication between analysts and developer(s) of an automated negotiation system. In the following sections, we describe each of the elements.

Negotiation Application Domain

The application domain in which agents will negotiate will determine the nature of many aspects of the negotiation, such as the negotiation objectives(s) and the rules that will govern that negotiation. We characterize the application domain according to the characteristics shown in Table One.

For example, consider a multiagent system in the domain of telecommunications service provisioning in which two agents will negotiate with each other over a digital subscriber line (DSL) service. The general characteristics of this application domain that are common to a wide range of negotiation scenarios within this domain are shown in Table Two.

After defining the general characteristics of the application domain in which agents negotiate with each other, we should define the specific negotiation scenario. After representing the scenario, we can define the appropriate negotiation system based on its elements.

Negotiation Scenario

As shown in Figure Four, after specifying the common characteristics of the application domain, we should specify and represent the parameters of the negotiation space in which agents currently interact with each other. According to Alessio Lomuscio and his colleagues (2003) and to Roy Lewicki and his colleagues (2000), every negotiation scenario can be defined based on a number of parameters.

A negotiation scenario is defined by a finite sequence (also known as a "tuple") of negotiation parameters that we call $\mathrm{NGSC} = \langle \mathit{Ocard}, \mathit{Icard}, \mathit{AChar}, \mathit{Env}, \mathit{Ochar}, E, I, A \rangle$, where:

- *Ocard* (object cardinality) defines the cardinalities of the set of issues being negotiated.
- *Icard* (interaction cardinality) defines the cardinalities of the interaction that takes place between agents.

Figure Five
A Model Representing the Conceptual Framework

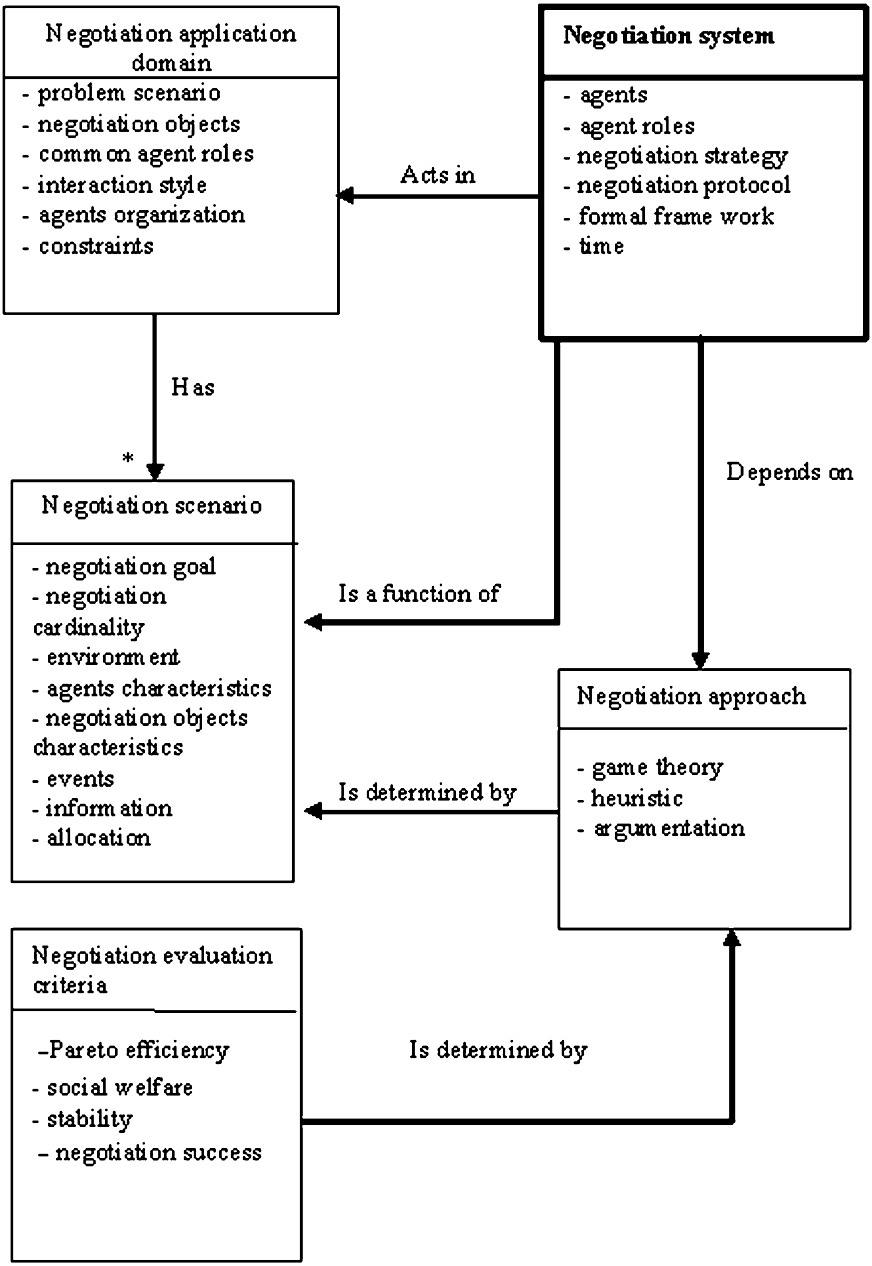


Table One
Common Characteristics of an Application Domain

Application Domain Attribute	Description
Application domain name	Common name of the application domain in which negotiation will take place, that is, telecommunication service provisioning or agricultural products market.
Problem scenario	A description of the scenario. This scenario specifies the typical interactions that take place in this domain when participants interact to achieve their design agenda.
Negotiation object(s) or issues(s)	List of issues to be negotiated.
Attributes of objectives	Negotiation involves one or more attributes of an object. In an automobile purchase negotiation, for example, such attributes could include price, delivery date, and warranties.
Possible roles of agents	Roles that agents will play.
Cardinality of negotiation objects	Number of issues/objectives at issue in the negotiation.
Style of interaction	The conflict orientation typically displayed by negotiators in this domain, that is, collaborative, competitive, accommodative, etc.
Agent organization	In this domain negotiation can be between individual agents or group of agents, that is, one-to-one, one-to-many, many-to-one, or many-to-many.
Constraints	What are the main constraints in this application domain? For example, negotiation time is a critical factor in some negotiations.

Table Two
Attributes of the Digital Subscriber Link (DSL) Service Provisioning Domain

Application Domain Attribute	Description
Application domain name	DSL Service Provisioning.
Problem scenario	The scenario begins when a customer with a specific need contacts the service provider to subscribe to DSL service. The scenario ends successfully when the customer and the service provider reach an agreement.
Negotiation object(s) or issues(s)	DSL service.
Attributes of object	Bandwidth, quality of service (QoS), delivery time, cost, duration, security, start time.
Possible roles of agents	DSL service provider, customer, broker, value-added service provider.
Cardinality of negotiation objects	Negotiations can range over a number of objects and multiple attributes of the objects, such as price, QoS, etc.
Style of interaction	Negotiations involve self-interested, utility-maximizing agents. Agents may share the same system goal but have different individual preferences.
Agent organization	Between individual agents or group of agents. Thus interaction cardinality can be one-to-one, one-to-many, or many-to-many.
Constraints	Time is a critical factor. Timing is important on two distinct levels: (1) the time it takes to reach an agreement must be reasonable, and (2) the time by which the negotiated service must be executed is important in most cases and crucial in others.

- *AChar* (agent characteristics) defines the characteristics of the agents. The characteristics of the agents depend on their attitudes, goals, motivations, roles, rationality, knowledge, commitment, social behaviors, trust and openness, predictability, aggressiveness, and decision-making strategies. (In agent technology, agents can have mental attitudes such as belief, desire, and intention, and we can represent these mental attitudes in the agents. Consider how we represent knowledge in the form of rules, frames, and other knowledge-representation techniques in expert systems. These mental attitudes may reflect the attitudes of parties that the agents represent.)
- *Env* (environment) is the type of environment in which the negotiation takes place. The environment can be static or dynamic.
- *Ochar* (object characteristics) is the set of characteristics of the issues under negotiation.
- *E* (event parameters) defines negotiation procedures and rules involving the validity and visibility of proposals, time-outs, schedules, etc.
- *I* (information parameters) defines information other than proposals that may pass between the negotiation participants.
- *A* (allocation parameters) is applied in many-to-one and many-to-many negotiations and governs the winner in a competitive negotiation system, that is, in which more than one party seeks to service a contract, for example.

After specifying the negotiation scenario based on these parameters, developer should specify the elements of the negotiation system.

Negotiation System

We use the term negotiation system to refer to the main elements of a computational system that should be implemented to achieve particular negotiation objectives in the specified application domain. The main components of this system are negotiation protocol, negotiation strategies, and the agents that participate in negotiation.

We represent a negotiation system with a finite sequence (or tuple) *NS*. This sequence is defined as $NS = \langle Agents, Roles, R, P, S, L, Time \rangle$, where:

- *Agents* is the set of negotiating agents.
- *Roles* is the set of roles that agents play in negotiation.
- $R: Agents*Agents \rightarrow Roles$ is a function that assigns a role to an agent.
- *P* is the negotiation protocol.
- *S* is the negotiation strategy.

- *L* is a formal language that defines the content of exchanges between agents. This language can be based on propositional logic or predicate logic. Propositional logic is the branch of logic that studies ways of joining and/or modifying entire propositions, statements, or sentences to form more complicated propositions, statements, or sentences, as well as the logical relationships and properties that are derived from these methods of combining or altering statements. Predicate logic is a system of deduction extending propositional logic by allowing quantification over individuals of a given domain of discourse.
- *Time* indicates a set of ordered discrete time intervals.

Negotiation Protocol

Negotiation protocol is a formal set of conventions governing the interaction among participants (Jennings et al. 2000; Rahwan et al. 2003); it specifies, at each stage of the negotiation process, who is allowed to say what.

Negotiation protocol is defined by a tuple $P = \langle A, \pi, I, S, O, R \rangle$, where:

- *A* is a set of valid actions that participants can perform in certain situations.
- π: $\mathrm{A} \rightarrow 2^{\mathrm{A}}$ is a protocol mapping function.
- *S* is a set of negotiation states, such as begun, offered, concluded, etc.
- *I:* A*S → S is a function that determines the next state.
- *O* is a set of negotiation objectives.
- *R* is a set of negotiation rules.

Negotiation Strategy

Negotiation strategy plans the action sequences of agents during negotiation. One simple and abstract way of defining negotiation would be to show it as a function *S*. This function maps the existing action set of the agent to a possible set of action sequences.

But, the previously mentioned definition is very simple and there are many parameters that are involved in determining what should be the agent's next action. According to R. W. Johnston (1985), the following characteristics of the negotiation and the parties combined to determine the overall negotiation strategy:

- Payoff structure: are the amount of resources to be divided among negotiators fixed or variable?.
- Goal pursuit: what is the relationship of each party's goals to the other party's goals?

- Relationships: do the parties expect to have a short-term or longer term relationship with each other?
- Primary motivation: does each party seek to maximize its own outcome, another party's outcome, or joint outcomes?
- Trust and openness: what is the level of trust and openness between the parties?
- Knowledge of needs/interests: do the parties know and understand their own interests? Do they seek to convey these to the other party or parties?
- Predictability: are the parties' actions predictable in the negotiation? Are they flexible?
- Aggressiveness: do the parties use threats and bluffs or share information honestly?
- Solution search behavior: do the parties stick inflexibly to their positions or do they try to find mutually satisfying solutions?
- Success measurement: how do the parties measure their own success, that is, by diminishing the other party or by minimizing conflict?
- Evidence of unhealthy extremes: do the parties display such unhealthy extremes as treating the negotiation as a zero-sum game or abdicating completely to the other side's position?
- Key attitude: what dominant attitudes do the parties display (hard bargaining, joint gains approach, etc.)?
- Remedy for breakdown: what are the possible solutions if there is an impasse (i.e., walking away from the negotiation, mediation, arbitration, etc.)

These characteristics can be used in choosing an appropriate negotiation strategy for the negotiator agents and also they can be used to represent a negotiation strategy. All these aspects should be considered when implementing a computational agent negotiation strategy.

Relationships among Framework Elements

In addition to specifying the entities necessary for the development of an automated negotiation system, the framework we propose for automated negotiation also specifies the main relationships among these elements. According to our experience in developing automated negotiation systems and our study of some of the developed negotiation systems, we have defined several relationships among the framework entities.

Application Domain and Negotiation Scenario

The application domain in which negotiation takes place (e-commerce, business process management, etc.) has a direct effect on negotiation scenario parameters, and we can consider the negotiation scenario that is specified as an instance of typical negotiations that take place in the specified application domain. For example, in the telecommunication application domain, a service provider can negotiate with other service provider(s) or connectivity provider(s) over price or other attributes of a service such as a service-level agreement (SLA) for providing a specific service to the customer. Or a customer can negotiate with a service provider over a specific telecommunication service. All these negotiation scenarios are affected by the general characteristics of the telecommunication domain (e.g., negotiators typically negotiate over telecommunication services, SLA, telecommunication equipment). The specific problem determines the negotiation scenario that will take place in that domain.

Negotiation Approach and Negotiation Environment

Iyad Rahwan and his colleagues (2003) have described three major classes of approaches to automated negotiations: *game-theoretic*, *heuristic*, and *argumentation-based* negotiation. Game theory analyzes the strategies each player uses to maximize the chance of winning and attempts to predict outcomes. We can analyze negotiation scenarios as a game between identical participants and determine each negotiator's optimal strategy given the game rules, assumed payoffs, and goals of each negotiator.

Heuristics are "rules of thumb" that produce "good enough" outcomes and are often produced in contexts with more relaxed assumptions about agents' rationality and resources. These rule of thumbs can be used in agents' strategies for decision making about incoming proposals and generating new proposals. Argumentation-based approaches to negotiation attempt to overcome the limitations of game-theoretic and heuristics-based approaches by allowing negotiator agents to exchange additional information in their locutions (Kraus, Sycara, and Evenchik 1998; Sierra et al. 1998). Agents can argue about their beliefs and other mental attitudes. (In this case, "argument" is a term meaning the exchange of information to persuade or justify.)

Such characteristics of the negotiation environment as uncertainty or lack of complete information can have a determining role in selecting the appropriate approach for negotiation (game theory, heuristic, or argumentation). For example, in cases in which there is incomplete or uncertain information a more appropriate approach for performing automated negotiation might be argumentation-based negotiation.

Negotiation Approach and Agents Design

Existing computational limitations in agents, their reasoning capabilities, and their mental attitudes (beliefs, intentions, and desires) are important in choosing an appropriate approach for negotiation. Also, choosing a specific approach for negotiation has direct effect on agent design. For example, if the agent will be using argumentation-based negotiation, then components for argument generation, argument interpretation, and argument selection must be considered in designing the particular agent.

Negotiation Approach and Negotiation Strategy and Protocol

The specific negotiation approach selected will determine the design of the agent's negotiation protocol and negotiation strategy. For example, if we choose an argumentation-based negotiation approach, then we would need a protocol that lets the agents exchange additional information in the form of description and elaboration, reasoning, and justification.

Negotiation Protocol and Negotiation Strategy

Negotiation protocol specifies the interaction rules, but the exact actions that an agent will undertake are the result of the chosen negotiation strategy. A number of negotiation strategies are usually compatible with a specific negotiation protocol, and each of them will produce different results. Thus, strategies that work with a specific protocol may not work with other negotiation protocols. Choosing an appropriate negotiation strategy is therefore a function of the parameters of both negotiation scenario and negotiation protocol.

Negotiation Approach and Evaluation Criteria of Negotiation System

As noted earlier, some scholars have proposed evaluation criteria for evaluating negotiation processes and results (see Lomuscio, Wooldridge, and Jennings 2003; McBurney, Parsons, and Wooldridge 2002). Some evaluation criteria are based on game-theoretic and heuristic approaches, while different criteria would be used to evaluate dialogue-based approaches. Although some of the proposed criteria overlap with each other, some of them would be specific to each approach. Thus, the chosen approach for negotiation should determine the evaluation criteria that are used. For example, for calculating the computational complexity of an argumentation-based negotiation system, three complexity categories are considered (Parsons, Wooldridge, and Amgoud 2003): first, the complexity of constructing an argument; second, the complexity of analyzing the minimality of the constructed argument (an argument's minimality is violated when a new fact allows for the construction of a new argument); and third, the complexity of determining the "undercut" of the statements made by the other party's agent. (In argumentation theory, one argument undercuts another argument when the conclusion of the first argument negates the premises of the second argument.)

Evaluation Criteria and Environment

In some cases, determining the necessary evaluation criteria will depend on the type of negotiation environment. For example, if social welfare is defined based on utility function, in an environment in which the number of agents changes, it is more appropriate to define the social welfare based on the average of utility functions (utility functions assign real numbers to members of a choice set) instead of the sum of the utility functions (Chevaleyre et al. 2005).

Negotiation Environment and Negotiation System

The characteristics of the negotiation environment guide decisions regarding the elements of a negotiation system. For example, utility functions should reflect the preferences of the parties via the agent. In a static environment, the agent will usually not "learn" during the negotiation, and its utility functions will, therefore, be designed as fixed functions. Obviously, however, fixed utility functions would produce less than optimum behaviors in more dynamic negotiation environments.

In a dynamic environment, if agent strategy changes or can be selected according to the environmental conditions, then we can say that the probability of reaching a satisfactory agreement — as well as the speed of reaching this agreement — will increase. Fixed strategies can be discovered or learned by the other party and other agents can use this knowledge in future negotiations. In a competitive negotiating situation, having a fixed strategy can put an agent at a disadvantage.

Parameters of Negotiation Scenario and Negotiation System

The parameters of the negotiation scenario will affect the design of the negotiation protocol, the negotiation strategy, and the negotiator agents. For example, the fact that negotiation cardinality is one-to-one (one party negotiating with just one other party) will determine the specific next steps that can be undertaken at a specific point in the negotiation: after each offer, just one proposal can be received, and after each proposal, just one counterproposal can be generated by the other side of negotiation. This rule would be different in many-to-one or other multiparty negotiations.

Likewise, the parameters of the negotiation scenario will determine the particular negotiation protocol. For example, in a negotiation that takes place in an e-commerce application domain in which seller agents negotiate with a small number of buyer agents, buyer agents would prefer a protocol in which they are removed from existing commitments after accepting a proposal, that is, their outstanding offers are withdrawn once they have accepted another offer. Of course, in this protocol we should assume that there is a limited time for accepting proposals and buyer agents cannot wait to receive all proposals from seller agents and then select the best one among the incoming proposals.

By analyzing such key parameters as cardinality, events, information, and allocation in detail, developers can define the negotiation system elements. For example, consider a classic buying–selling negotiation. This type of negotiation can take place among any number of agents.

Table Three
The Effect of Interaction Cardinality on Negotiation Protocol Design

Agent Role	Number	Negotiation Protocols	
Customer	One	Bargaining	Reverse auction, tendering, RFQ
	Many	One-to-one negotiation with each customer, auctions Auctions	Double auctions, market-based negotiations, stock exchanges
		One	Many
		Service provider	

Table Three shows possible negotiation protocols that are used in different negotiation scenarios, as well as the role of interaction cardinality on the chosen protocol for negotiation. The cardinalities of the negotiation objectives will determine the design of a utility function used to determine the relationship of the different objectives in the negotiation to each other.

Negotiation strategy effects negotiation cardinality. For example, one of an agent's actions for addressing stalemate in a negotiation could be to invite other agents to join the negotiation, changing the number of involving agents and negotiation cardinality. (In this way, negotiation agents can also be seen as engaging in "beyond-the-table" moves.) In a dynamic environment, agents should adopt their preferences as they progress in the negotiation process.

Event parameters form an important part of negotiation protocol specification for the design of negotiation agents. Event parameters will mainly determine the rules governing the validity and visibility of proposals, as well as the termination rules (Lomuscio, Wooldridge, and Jennings 2003).

If the agents must prenegotiate before starting the main negotiation, this information parameter will also affect the design of the negotiation protocol and negotiation strategy. Xiaoqin Zhang and his colleagues introduced a prenegotiation phase in semicooperative negotiation chains that allowed agents to transfer metalevel information (Zhang and Lesser 2007). Using this information, the agent should be able to build a more accurate model of the negotiation that better models the relationship between flexibility and the probability of reaching a successful agreement.

If such information as the description of the proposal, or the justification and reasoning for the offer will be exchanged in addition to straight offers, then a designer should consider protocols for dialogue-based negotiation (Amgoud, Parsons, and Maudet 2000) as well as techniques for interpretation and evaluation of arguments made as part of the negotiation strategy (Rahwan et al. 2003). In summary, we can say that the following two relationships exist among the negotiation parameters and elements of negotiation system:

1. (P) = $\mathcal{F}$(*Icard*, *Ochar*, *E*, *I*, *A*). (Negotiation protocol is a function of interaction cardinality, objective cardinality, event parameters, information parameters, and allocation parameters.)
2. S = $\mathcal{F}$(*Ocard, Icard*, *Achar*, *Env, I*). (Negotiation strategy is a function of objective cardinality, interaction cardinality, agent characteristics, environment, and information parameters.)

Representation technique also plays a role in the development of a negotiation system. We consider two levels of representation: a high level of representation of the main elements of the framework and the representation techniques that are used to specify the elements of the negotiation system. In this article, we have used a simple tuple-based representation for representation technique, just as we did to define negotiation scenario, negotiation system, and negotiation protocol.

For representing the elements of a negotiation system, existing representation techniques are used. For example, a variety of representation techniques have been used for representing negotiation protocols. The examples of such representation techniques are:

- *finite state machines* (FSM), a model of behavior composed of a finite number of states, transitions between those states, and actions (Parsons, Sierra, and Jennings 1998);
- *Petri Net*, one of several mathematical representations of discrete distributed systems, has place nodes, transition nodes, and directed arcs connecting places with transitions (Xu and Shatz 2001);
- *dialogue games*, a set of rules that determines for each participant what dialogue moves are allowed in a given dialogue context (Dastani, Hulstijin, and van der Torre 2000);
- *event calculus*, a logical language for representing and reasoning about actions and their effects (Yolum and Singh 2002);
- *logic programs*, the use of mathematical logic for computer programming (Sadri, Toni, and Torroni 2002); and
- *extended propositional logic* (Parsons, Sierra, and Jennings 1998).

Some of these representation techniques such as FSM or Petri Net are inflexible because they model protocols in terms of legal sequences of actions. In this way, agents do not compute transitions during the negotiation process and follow a preestablished formal technique for representing the protocol. Thus, this kind of formalism depends on the previous action and strictly enforces a certain behavior, and the flexibility of the agents in executing these protocols is limited and the protocols are overconstrained (Ayatollahzadeh Shirazi and Abdollahzadeh Barfouroush 2006).

Approaches such as dialogue games provide clear and precise declarative semantics of the interactions by stating the pre and postconditions of each locution as well as its effects on agents' commitments. In this way, agents that follow the protocol use the action semantics of protocol and reason about the next action.

In the next section, we describe how this framework can be used to develop a negotiation system for a multiagent system that works in a telecommunications service domain.

Representing a Distributive Bargaining Scenario

As an example, we consider an automated distributive bargaining scenario in the domain of DSL service provisioning. (The general characteristics of this application domain were mentioned in the first section of the article.) In this domain, the service provider creates software agent(s) that sell one or more DSL services to the customer's agent(s) directly or through broker agents. Resources are fixed and limited and each agent seeks to maximize its outcome. Each agent has a valuation function to compute the value of the received proposal. The agents consider three main issues in their negotiation with each other:

- The target point or final agreement point (x*) is the point at which the agent seeks to conclude the negotiation.
- The resistance point or reservation price is the minimum amount that the provider agent will accept and also the maximum value that the customer will agree to pay for the service (b).
- The asking value is the initial price set by the provider agent (s).

Typically, as in most distributive buyer–seller negotiations, the reservation price of each agent is unknown to the other agent. Other agents usually try to model their negotiation counterparts to estimate this value or they may try to persuade the other party to adjust their opinion of the product's value using argumentation. In this negotiation, the objective of both agents is to get their settlement point as close to the other agent's reservation price as possible. In distributive bargaining, if the reservation price (b) is lower than s, then there is no zone of possible agreement (ZOPA). If the reservation price is greater than **s**, then there is a ZOPA in which the parties may reach an agreement.

As shown in Figure One, after defining the main characteristics and assumptions of the application domain, the next step is to define the negotiation scenario based on its parameters. In this case, we consider a negotiation scenario in which some of the parameters change during the negotiation process because the nature of the environment changes (e.g., environment openness), and/or problems are resolved during the negotiation process (e.g., removing deadlocks), and/or the agent's negotiation strategy changes.

We define this scenario based on the following parameters:

- *Ocard* is a changing parameter in this scenario. At the beginning, only one issue is under negotiation, but during the negotiation, this parameter changes, and an additional issue is negotiated. For example, initially the agents might have negotiated only the price of the service, but then other terms of service, such as quality of service or delivery time, might be added to the negotiation object set.

- *Icard* is one-to-one; the negotiation takes place between one service provider and one customer.

- *AChar*, which includes:
  - *Achar role* (agent role, either service provider or customer).
  - *Achar rationality* (the agent's rationality, in this case it is bounded).
  - *Agent knowledge*, which is a changing parameter. At the beginning of the negotiation process, the agents know nothing about each other's reservation price and utility functions. During the negotiation process, however, one party gains an estimation of the reservation price of the other party. This means that this estimation is added to the knowledge base of each agent during negotiation.
  - *Commitment*, in which each agent agrees that once it has made an offer, it will not begin a different negotiation about the same object until it receives an acceptance or rejection of that first offer.

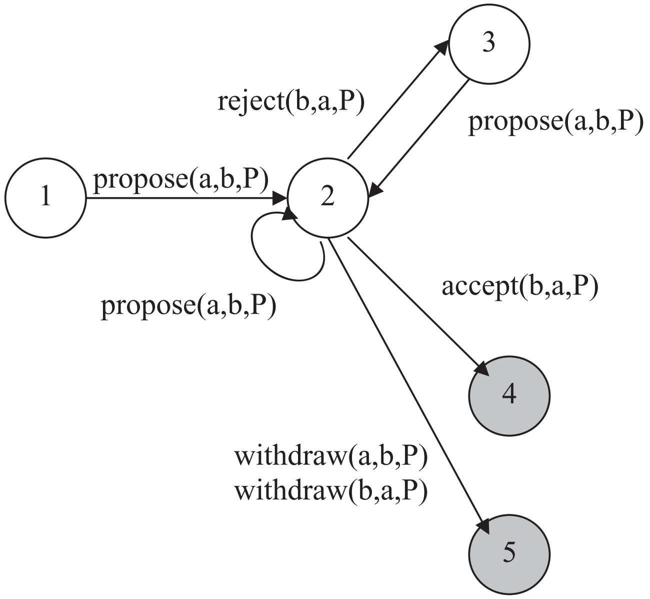
- *Env* is static.

- *Ochar* is for private use and discrete.

Now, based on this scenario, we must define the negotiation protocol and negotiation strategy for the negotiator agents. For negotiation protocol, we define the negotiation action set, negotiation object set, final state of negotiation, and behavior of the protocol based on a representation such as an FSM.

The negotiation strategy for a given scenario is defined for each negotiation party. We use a rule-based representation for defining the rules that an agent will use for making decisions about its next action. Strategy rules

Figure Six
Representation of Negotiation Protocol with a Finite State Mac



define the way an agent interprets an incoming message and generates an appropriate action.

The elements of the negotiation protocol for this scenario are defined as follows:

- Negotiation action set is defined as A = {propose, accept, reject, withdraw}.
- Negotiation objective is defined as O = {DSL service}.
- The final state of negotiation is defined as closed or withdrawn.
- Agent roles are defined as service provider or customer.
- Behavior and state mapping of the negotiation protocol is defined by using the state machine as shown in Figure Six.

Negotiation strategy plays a critical role in all negotiation scenarios as the decision-making mechanism that determines the next action the negotiator should take. In the defined scenario there are two parameters, negotiation objective and agent knowledge, that affect the design of the negotiation strategy.

In some situations, such as lack of progress in negotiation, agents may decide to remove an issue or objective or to introduce a new one. For example, in an automobile purchase negotiation, the dealer may offer a warranty to convince the buyer to make the deal. In some cases, the items that are added or removed are not already under direct negotiation and their role is to "sweeten the deal" and bring it to a conclusion.

During the negotiation, the knowledge of participant agents about each other and about the negotiation objectives and issues may change. For example, in negotiation for buying a product, the customer agent, the service provider agent, or both may learn information about each other's reservation price. This new knowledge will change the way that agents negotiate with each other. The strategy may change: one or both of the negotiator agents may terminate the negotiation, or one or both may change the quality or quantity of the issues being negotiated.

Each agent's negotiation strategy should be designed in such a way that the agent can handle these changes in negotiation parameters. We define a negotiation strategy for agents that can handle changes in negotiation objectives and issues and in the agent's knowledge during negotiation. In the scenario we will describe below, the service provider agent will be informed about the highest price that the customer agent is willing to pay for a service. To represent the agent's strategies, we have used predicates of predicate logic. We assume that the agents have a belief base in which they can store factual information about the world and we have included some fuzziness in the decision-making rules to better encode the reasoning rationale of human decision makers within the negotiator agents. Further in the text, we describe the strategy of the service provider and customer agents.

Strategy of the Service Provider

1. *If* the customer offers the service provider a price x for service m, and the price x is much lower than the reservation price of the service provider for service m, *then* the service provider rejects the customer proposal for service m.

2. *If* the customer offers the service provider a price x for service m, and the service provider knows the reservation price of the customer, *then* the service provider offers a proposal with a price that is near the reserved price of the customer.

3. *If* the customer offers the service provider a price x for service m, and the service provider knows the reservation price of the customer, and the customer's offer is not near this value, *then* the service provider proposes with a price that is closer to the customer's reservation price.

4. *If* the customer offers the service provider a price x for service m, and this price is near the gained benefit of the service provider for selling the service m at price x, *then* the service provider accepts the offer.
5. *If* the customer offers the same price for service m in two consequent rounds, and there is no progress in negotiation, *then* the service provider can add a new issue to the negotiation.
6. *If* the customer rejects the service provider's previous proposal, *then* the service provider decreases the asking price based on a defined function, but the new price will not be close to the reservation price.
7. *If* the negotiation has taken longer than was expected, then the service provider will withdraw the negotiation.

Strategy of the Customer Agent

1. *If* the service provider rejects the customer's previous proposal, *then* the customer increases the offered price based on a defined function so that the newly offered price is not close to the customer's reservation price.
2. *If* the service provider proposes a price x for service m that is much higher than the customer's reservation price, *then* the customer rejects the service provider's proposal.
3. *If* the customer's reservation price is much lower than the market valuation for the service, and customer needs the service, *then* the customer revises its reservation price based on information about the market valuation and utility of the service.
4. *If* the customer's reservation price is much lower than the market valuation for the service, but the customer cannot revise its reservation prices based on market valuation and utility, *then* the customer terminates the negotiation.
5. *If* the service provider proposes the price x for service m, and price x is higher than the market valuation of service m, *then* the customer rejects the offer.
6. *If* the service provider proposes price x for service m, and price x is close to the market valuation of service m and lower than the customer's reservation price, and the utility is higher than a specified threshold, *then* the customer accepts the proposal.
7. *If* the service provider proposes a price x for service m that is *lower* than the customer's reservation price, and the service provider has decreased the proposed price, *then* the customer proposes a price that is even lower, according to a certain function.

In the previously mentioned strategy, if the service provider agent receives the same price twice consecutively, it will try to introduce a new issue to advance the negotiation and avoid stalemate. Of course, this strategy is only one of several possible ways to break the deadlock.

Also, during the negotiation, the service provider agent may learn about the customer agent's reservation price. In this case, the service provider agent can change its strategy and offer prices closer to this reserved price while avoiding making offers substantially lower than this price. If the service provider's agent finds that the highest price that the customer will pay for the service is lower than the service provider's minimum price, then the service provider agent will terminate the negotiation.

Conclusion

Negotiator agents need to negotiate with each other to manage their task interdependencies, share resources, and obtain the necessary resources for performing their tasks or reaching their goals. Our experiences in developing multiagent systems show that negotiation exposes itself as a system. Negotiation protocol and strategy, and the architectural aspects of negotiator agents are the main elements of this system.

Many parameters of the application domain and the negotiation scenario will influence these elements. For example, such characteristics of the application domain as the style of the interactions and the nature of the negotiation issues and objectives affect the design of the negotiation protocol. Every negotiation scenario is generated by a number of negotiation parameters, which each influences the design of negotiation protocol and strategy.

It is critical to identify the relationships among the application domain and negotiation scenario parameters and the elements of the negotiation system. Thus, successful automated negotiation construction requires a *process* for constructing the negotiation system and a *framework* that represents the main concepts and entities in the development process and their relationships with each other.

In this article, we have described a process and conceptual framework for modeling and developing automated negotiation systems. This conceptual framework can be used as reference model for developing concrete automated negotiation systems. The framework has been used in our work for evaluating various negotiation scenarios for a flexible negotiation system as a template for capturing and representing the requirements of a practical negotiation system. It serves as a medium for communication between analysts and developer(s) of a computational negotiation system. It can also be used for representing human negotiations scenarios that we want to analyze and simulate with a multiagent system.

REFERENCES

Abdollahzadeh Barfouroush, A., B. Masoumi, and M. R. Ayatollahzadeh Shirazi. 2006. *An introduction to distributed artificial intelligence* [In Persian]. Tehran, Iran: Jelveh Publishing.

Amgoud, L., S. Parsons, and N. Maudet. 2000. Modelling dialogues using argumentation. *Proceedings of the 4th International Conference on Multi-Agent systems*. Boston.

Ayatollahzadeh Shirazi, M. R. and A. Abdollahzadeh Barfouroush. 2006. Flexibility in negotiation systems: An argumentation-based approach. *Proceedings of the 11th International Computer Society of Iran Computer Conference (CSICC2006)*. Tehran.

Bartolini, C., C. Preist, and N. R. Jennings. 2002. A generic software framework for automated negotiation. *Proceedings of AAMAS '02*. Bologna, Italy, July 15–19.

Benyoucef, M. and S. Rinderle. 2005. A model-driven approach for the rapid development of e-negotiation systems. Paper presented at Proceedings Enterprise Modeling and IS Architectures Workshop. Klagenfurt, Austria. October 2005.

Chevaleyre, Y., U. Endriss, S. Estivie, and N. Maudet. 2005. Welfare engineering in practice: On the variety of multi-agent resource allocation problems. *Engineering Societies in the Agents World V, Lecture Notes in Computer Science*, 3451: 335–347.

Dastani, M., J. Hulstijin, and L. van der Torre. 2000. Negotiation protocols and dialogue games. *Proceedings of the Belgium/Dutch AI Conference (BNAIC 2000)*. Kaatsheuvel, the Netherlands.

Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. 2000. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation* 10(2): 199–215.

Johnston, R. 1985. Negotiation strategies: Different strokes for different folks. In *Negotiation: Readings, exercises, and cases*, edited by R. Lewicki and J. Litterer. Homewood, IL: Irwin.

Kim, J. B. and A. Segev. 2003. A framework for dynamic e-business negotiation processes. *Proceedings of the IEEE International Conference on E-Commerce Technology, 84*. Newport Beach, CA.

Kraus, S., K. Sycara, and A. Evenchik. 1998. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence* 104(1/2): 1–69.

Lewicki, R., D. M. Saunders, and J. W. Minton. 2000. *Negotiation*, 3rd edn. Boston: McGraw-Hill.

Lomuscio, A. R., M. Wooldridge, and N. R. Jennings. 2003. A classification scheme for negotiation in electronic commerce. *International Journal of Group Decision and Negotiation* 12(1): 31–56.

McBurney, P., S. Parsons, and M. Wooldridge. 2002. M. Desiderata for agent argumentation protocols. *Proceedings of the international conference on autonomous agents and multi-agent systems*. Bologna.

Parsons, S., C. Sierra, and N. R. Jennings. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3): 261–292.

Parsons, S., M. Wooldridge, and L. Amgoud. 2003. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation* 13(3): 347–376.

Rahwan, I., S. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. 2003. Argumentation-based negotiation. *Knowledge Engineering Journal* 18(4): 343–375.

Sadri, F., F. Toni, and P. Torroni. 2002. Abductive logic programming architecture for negotiating agents. *Proceedings of the eighth European Conference on Logics in Artificial Intelligence (JELIA-2002), Lecture Notes in Computer Science 2424*. Berlin: Springer Verlag.

Sierra, C., N. R. Jennings, P. Noriega, and S. Parsons. 1998. A framework for argumentation-based negotiation. In *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997). Lecture Notes in Artificial Intelligence 1356*, edited by M. Singh, A. Rao and M. Wooldridge. Berlin: Springer Verlag.

Xu, H. and S. M. Shatz. 2001. An agent-based Petri Net model with application to seller/buyer design in electronic commerce. *Proceedings of fifth international symposium on autonomous decentralized systems*. Dallas, TX.

Yolum, P. and M. P. Singh. 2002. Flexible protocol specification and execution: Applying event calculus planning using commitments. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-agent Systems*. Bologna, Italy: ACM Press.

Zhang, X. and V. Lesser. 2007. Meta-level coordination for solving negotiation chains in semi-cooperative multi-agent systems. *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*. Honolulu, Hawaii, May 2007.