15

e-Negotiation Systems and Software Agents: Methods, Models, and Applications

Peter Braun[1], Jakub Brzostowski[1], Gregory Kersten[2], Jin Baek Kim[2], Ryszard Kowalczyk[1], Stefan Strecker[2], and Rustam Vahidov[2]

1 Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, Australia
Emails: {pbraun, jbrzostowski, rkowalczyk}@it.swin.edu.au
2 John Molson School of Business, Concordia University, Montreal, Canada
Emails: {kersten, jbkim, strecker, vahidov}@jmsb.concordia.ca

Negotiation is a decentralized decision-making process that seeks to find an agreement that will satisfy the requirements of two or more parties in the presence of limited common knowledge and conflicting preferences. Negotiation participants are agents who negotiate on their own behalf or represent the interests of their principals. When electronic negotiations enter the stage, these agents could be intelligent software entities that take part in the process of searching for an acceptable agreement. The degree of involvement of these "intelligent agents" in negotiations can range from supporting human negotiators (*e. g.* information search, offer evaluation) to fully automating the conduct of negotiations. Choosing the degree of involvement depends upon the characteristics of the problem in the negotiation. In this chapter, we review electronic negotiation systems and intelligent agents for negotiations. Different types of negotiation agents, their roles and requirements, and various methods for effective support or conduct of negotiations are discussed. Selected applications of intelligent negotiation agents are presented.

15.1 Introduction

Negotiation is a decentralized decision-making process used to search for and arrive at an agreement that satisfies the requirements of two or more parties in the presence of limited common knowledge and conflicting preferences. Negotiation processes appear in a multitude of forms. They occur in very different situations and are influenced by ethical, cultural and social circumstances. These processes and their participants have been a research topic of many disciplines including anthropology, psychology and sociology, political sciences (Ury 1993; Fisher *et al.* 1994), law (Wetlaufer 1996), economics (Young 1975, Roth 1995), applied mathematics (Harsanyi 1997), and computer science (Sycara 1997, Kraus 2001).

The use of software to support negotiation processes was put forward in the late 1970s. Empirical research on computer-mediated communication systems such as Hiltz and Turoff (1978) preceded research on systems supporting negotiations. Keen and Scott-Morton (1978), Sprague and Carlson (1982), and others proposed to extend decision support system (DSS) capabilities to aid the negotiators. This led, in the early eighties, to the design of negotiation support systems (NSSs) and group decision support systems (GSSs) (Korhonen *et al.* 1986, Jarke *et al.* 1987, Jelassi and Foroughi 1989). Negotiation support systems are designed to help and advise negotiators during the various phases of the negotiation process; they are used to structure and analyze the negotiation case, elicit preferences and use them to construct a utility function, determine feasible and efficient alternatives, set negotiation tactics, visualize different aspects of the problem and the process, and facilitate communication.

NSSs are based on the modeling approaches formulated in decision sciences, negotiation analysis, and game theory (Raiffa *et al.* 2003). The contribution of decision science to negotiation includes decision rules, decision trees, single- and multi-attribute utility theory, and statistical methods such as forecasting and regression analysis.

Negotiation analysis integrates decision analysis and game theory in order to provide methodological support to negotiation participants. Approaches based on negotiation analysis aim at bridging the gap between descriptive behavioral models and normative formal models of bargaining. These approaches have adopted a number of behavioral concepts, including reservation and aspiration levels, the best alternative to the negotiated agreement (BATNA), and integrative and distributive negotiations, and incorporated these concepts into quantitative models (Kersten 2001). This allowed advisors to conduct formal analysis of negotiations in order to support negotiators. Other approaches stemming from computer science, especially Artificial Intelligence, have also been used in the design of software that aids one or more negotiators (Matwin *et al.* 1987, Rangaswamy *et al.* 1989, Kersten 1993).

A good classification of NSSs and DSSs can be found in Starke and Rangaswamy (1999) who distinguish them by preparation and evaluation systems and process support systems, and in Kersten (2004) who further classifies them considering the phase of the negotiation process: (1) planning systems; (2) assessment systems; (3) intervention systems; and (4) process systems.

The Internet and new computing and communication technologies introduced new opportunities for the design and deployment of software capable of supporting negotiators, mediators and arbitrators. Negotiations conducted over the Web are commonly called e-negotiations and the systems used in e-negotiations are named e-negotiation systems (ENSs). ENSs are information systems that employ Internet technologies that are deployed on the Web. Defining ENSs as software deployed on the Web, capable of aiding one or more negotiators, mediators or facilitators allows us to include e-mail, chat and streaming video used in negotiations (Moore *et al.* 1999, Lempereur 2004), as well as software used for automated negotiations and auctions (Zlotkin 1996, Jennings *et al.* 2001).

e-Negotiation systems are unlike previous systems deployed on standalone computers or local- and even wide-area networks in terms of the implemented mechanisms and employed technologies. Specifically, the potential of intelligent software agents has been noted for their suitability in a distributed computing environment such as the Internet.

Software agents are programs that carry out certain operations on behalf of a user or another program with some degree of independence or autonomy and, in doing so, realize a set of goals or tasks for which they are designed (Jennings and Wooldridge 1998; Maes 1998; Jennings 2001). These programs differ from regular software because they are personalized, continuously running, and to a certain extent autonomous. The reasoning mechanisms of software agents can range from a set of simple “if-then” rules to sophisticated machine learning algorithms such as neural networks or Bayesian networks (Caglayan and Harrison 1997, Wooldridge 1999).

Software agents carrying out negotiation activities on behalf of users are known as negotiation software agents (NSAs). These agents have been developed to study the automation of different negotiation tasks that arise from buying and selling products over the Internet.

The purpose of this chapter is to investigate the full potential of NSAs and explain research issues. To do so, we first review e-negotiation systems and investigate models for positioning NSAs in ENSs. Then, we examine models and techniques for NSAs. Finally we present applications of NSAs, and conclude with an outlook to further development.

15.2 Foundations

The use of software in negotiations requires that a *process model* and a *protocol* is constructed (Kersten and Lo 2003, Kim and Segev 2003). The process model describes negotiation phases and assigns different activities to them. Its significance is in that it allows the negotiators to follow a methodologically sound approach (Lewicki *et al.* 1999). The protocol is a formal model, often represented by a set of rules, which governs software processing and communication tasks, and imposes restrictions on activities through the specification of permissible inputs (Jennings *et al.* 2001).

Behavioral research on negotiations has so far not included the processes in which support systems and software agents are involved as active participants and, therefore, no process models have been developed specific to e-negotiation.

Hence, we need to adapt a behavioral phase model to reflect the requirements imposed by an ENS. We have adapted a model proposed by Kersten (1997), which is based on Gulliver’s eight-phase model (1979). This model has been modified to allow for a wider range of negotiated decisions than the eight-phase model, including those which use ENSs. The model, presented in Figure 15.1, comprises the following five phases:

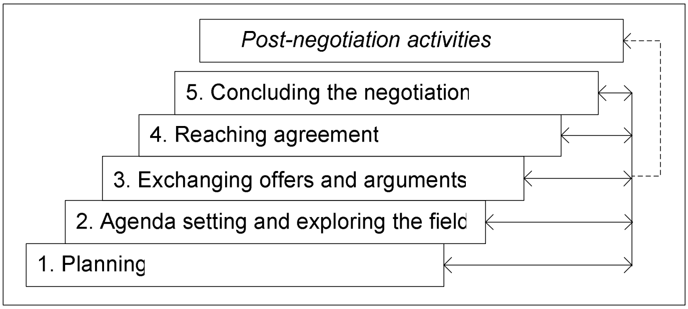

Figure 15.1. Negotiation process model

- The planning phase comprises activities that the negotiators undertake both individually and jointly. They formulate their representation of the negotiation problem including the specification of issues and options. In this phase the negotiators specify their objectives and preferences, and such negotiation-specific constructs as BATNA and reservation levels (Fisher *et al.* 1991). If the negotiators know or can learn about their opponents, they decide on strategies to be used. This phase's joint activity also includes the selection of the negotiation location and time, and the communication modes the negotiators will use.

- Agenda setting and exploring the field includes the negotiators' discussions about the negotiated issues and their meaning. The discussion's result may be that new issues and options are added or some may be deleted. The negotiators may also discuss the protocol they will follow, the timing of the exchanges, the deadline and—in some negotiations—their objectives, priorities and constraints. The result of these discussions is that the negotiators may have to revise the problem, objectives and preferences, and also their strategies and initial tactics.

- Exchanging offers and arguments allows the parties to learn about the others' limitations, and to identify the key issues and critical areas of disagreement. During this phase, the parties realize the potential of a compromise and can assess its main features. The analysis of a negotiation may focus on the modification of strategies, the determination of concessions and revision of aspiration levels, and on the restriction of efficient solutions to those that may be acceptable to the parties.

- Reaching an agreement means that the parties realize that the negotiation has been successful. Having identified the critical issues, they may develop joint proposals or soften their individual limitations. The parties may also identify a limited number of possible compromises.

- Concluding the negotiation takes place when the negotiators reach an agreement. They evaluate this compromise and consider its possible improvements. They also may discuss additional issues that, however, have no impact on the negotiations (*e. g.* the agreement implementation).

The negotiation phase model provides a structure to the negotiation process. This is not to say that the negotiator who conducts activities belonging to one phase cannot return to another. Negotiations rarely proceed in a linear fashion. While the negotiation methodology suggests that the parties should not bypass or enter a phase before completing the previous phase, the parties may in reality at some point during the course of the negotiation need to return to one of the previous phases. This situation could arise if information obtained during the process requires a revision of assumptions and/or specifications. For example, parties may suggest additional communication channels and propose new issues to be negotiated. Such changes may require the revision of reservation levels and of preferences. This possibility of revisiting previous phases and then returning to the current phase is indicated in Figure 15.1. The possibility of ignoring some phases is also shown in Figure 15.1. This practice, while it occurs in real-life negotiation, should not be allowed in ENS-supported negotiations. Instead, the process model and protocol underlying an ENS should be based on a sound methodology and carried out, among others, by the sequencing imposed by the phase model.

15.3 e-Negotiation Systems

The use of the early NSSs was limited due to: (1) limitations of information and communication technologies (ICTs); (2) limited computer literacy of managers, who, therefore, delegated the system usage to analysts; (3) complexity of the constructed models, often based on strong rationality principles that required a significant amount of users' input; and (4) insufficient consideration of psychological and sociological conditioning of negotiations.

New ICTs, including Internet, software architectures, and software development technologies made rapid development of systems for millions of users possible. They also exposed people to information systems and their practical use for shopping, communication, information retrieval and entertainment. These developments led to two new streams of theoretical and applied research:

- Behavioral research on the use of communication technologies (mainly email) in negotiations (Purdy and Neye 2000, Thompson and Nadler 2002); and

- Design and development of easy-to-use systems for negotiation support (Kersten 1999).

e-Negotiation systems (ENSs) have one or more of the following capabilities:

- To support decision and concession making;
- To suggest offers and agreements;

- To assess and criticize offers and counteroffers;
- To structure and organize the process;
- To provide information and expertise;
- To facilitate and organize communication;
- To aid agreement preparation; and
- To provide access to negotiation knowledge; experts, mediators or facilitators.

Early ENSs were designed by university researchers around a single model or a procedure. The Inspire system (Inspire 1996) is an implementation of a method for utility construction based on hybrid conjoint analysis (Kersten 1999). The Joint Gains system (JointGains 2000) implements the joint improvement directions method (Ehtamo *et al.* 2001). Since the mid-1990s these two systems have been used in teaching and research. The virtual property agency system (defunct) implemented linear weighted value function (Bui *et al.* 2001). Some of these systems do not support negotiations but facilitate them by providing an electronic bargaining table (Rangaswamy and Shell 1997). Other systems focus on the contracts' preparation and the content of documents (Schoop and Quix 2001).

Most of the early commercial ENSs were also single-purpose. CyberSettle (Cybersettle 2000) is an online system that supports its users in negotiating single-issue insurance claims. It has a simple conflict resolution mechanism based on expanding offers made by each party by 20%. The Electronic Courthouse (NovaForum 2000) provides alternative dispute resolution (ADR) services by linking claimants with a roster of lawyers and ADR professionals.

The common feature of these systems is their orientation to one specific type of negotiation interaction. They provide a service that is based on the assumptions underlying the theoretical model or—in the case of commercial ENS—the implemented business rules. These types of systems are *ENS applications* supporting one type of negotiation (Rebstock 2001, Neumann *et al.* 2003). Table 15.1 summarizes ENS functions and activities.

NSSs have very limited autonomy; their purpose is to provide support to one or more users to assess decision alternatives, select offers and evaluate counteroffers, and to communicate with their counterparts. In contrast, NSAs have significant autonomy in their decision-making and communication activities. The NSA acts for and on behalf of the principal, the agent actively helps the principal and seeks information, evaluates the principal and others decisions, and communicates with the counterpart. While both an ENS and NSA may try to help the negotiators understand the problem, express their preferences, represent the process and formulate the exchanges, an ENS is passive; it does not attempt to seek information from various sources, interfere in the process, propose and/or make offers, or assess and present arguments for offer acceptance or rejection.

Table 15.1. ENS functions and activities

Function	**Activities**
Communication, presentation and interaction	
Transport and storage	Transport of information among heterogeneous systems; storage in distributed systems; security.
Search and retrieval	Search of information; selection; comparison and aggregation of distributed information.
Formatting, presentation and interaction	Data formatting for other systems use; data visualization, alternative data presentation, user-system interaction.
Modeling and content formulation	
Decision problem formulation	Formulation and analysis of the decision problems; feasible alternatives; decision space, measurement.
Decision-maker specification	Specification of constructs describing decision makers; preferences; measures for alternative comparison; negotiators' models and styles.
Strategies and tactics	Evaluation and selection of the initial strategies and tactics.
Negotiation	
Offer and message construction and evaluation	Formulation of offers and concessions; analysis of messages and arguments; argumentation models.
Counterpart analysis	Construction and verification of models of negotiation counterparts; evaluation and prediction of their behavior
What-if, sensitivity and stability analyses	Analysis of offer and counteroffer implications; analysis of the implication of different offers on the counterparts' reactions; assessment of the potential compromise solutions.
Process, history and their analysis	Construction of the negotiation history; process analysis; progress/regress assessment; history-based predictions.
Knowledge seeking and use	Access and use of external information and knowledge about negotiation situations and issues arising during the process; comparative analysis.
Negotiation protocols	Specification of, and adherence to, the negotiation agenda and rules
Strategies and tactics	Assessment counterparts' of strategies and tactics; modification of strategies and tactics

The possible functions of NSAs depend—in addition to the available technologies and knowledge—on their required degree of the agent's autonomy, the type of negotiation, and the specificity of the principal's instructions. The functions depend also on the expected scope and form of the agent's interactions with other systems and agents. The agent may be highly specialized and may cooperate with other agents, interact directly with the principal, or it may communicate via a DSS or a NSS that supports the negotiators in the construction of problem representations and in their assessment and modification. The agent may follow the principal directives or it may suggest new issues/options and innovative (for the principal) approaches to cope with conflict, based on the information obtained from experts and extracted from other negotiation histories.

Negotiation software agents may take over well-defined and structured activities in a negotiation but it is not necessary for the agents to handle all of the tasks. For example, the agent may present offers, seek information about similar negotiation situations, collect information about the counter-parts, and alert the principal if pre-defined conditions are violated. The ill-defined and ambiguous issues, decisions regarding relationship between the parties, modification of the rules and parameters are better left to the principals.

Complex and rich processes comprise both routine and simple tasks as well as tasks that are original and require imagination. Business negotiations are often an example of processes requiring the use of both ENS and NSA technologies. There is a need to develop tools and infrastructure that can support and also independently conduct activities. In business-to-business negotiations flexible and extensible tools are needed to support both integrative and distributive activities. These tools have to be highly interactive and competent at managing the complexity of multilateral business-partner relationships, especially since each business negotiation tends to be unique, in small, but important, ways.

Among other things, a particular architecture depends on the complexity of interactions with the principal, level of support required, and the requirements for information processing by other systems (*e. g.*, financial, marketing and production). In the next section, we explain the types, functions, and architectures of NSAs for ENSs.

15.4 Negotiation Software Agents

This section aims at synthesizing the relationships between software agent capabilities and relevant tasks in different negotiation phases within a coherent framework. Despite the lack of a well-formulated and widely accepted definition of the concept of software agent (Wooldridge and Jennings 1995, Franklin and Graesser 1997), we adopt a natural metaphor view of an agent (Jennings and Wooldridge 1998). This allows us to use the notion of an agent as an abstraction tool for structuring the design of complex software systems (Jennings 2000, Jennings 2001, Luck *et al.* 2003).

The first important issue to be addressed is what types of agents will be useful in supporting negotiation tasks. Franklin and Graesser (1997) have proposed a classification scheme for agents based on the properties they possess. Nwana and Ndumu (1998) have identified autonomy, cooperation, and learning as subsets of dimensions for deriving classes of agents. In their schema, agents possessing cooperation and autonomy features would be referred to as “collaborative agents”, while those with learning and autonomy properties would be described as “interface agents”. Agents possessing all three features were identified as “smart” agents.

Kinny *et al.* have advocated the methodology for belief-desire-intention (BDI) agent-based systems that includes identifying the roles and duties of these agents as the initial step (Kinny *et al.* 1996). We will follow a similar approach here by first proposing the type of agents to be potentially employed in support of e-negotiations, and then detailing the responsibilities they will have in regard to different phases of the negotiation process. A similar approach was used in studies on the design of agent-based decision support systems. There, the authors identified groups of agents supporting the three phases of Simon's problem solving model (Simon 1977), including intelligence, design, and choice (Vahidov and Fazlollahi 2004).

In order to conceptualize the role of agents in e-negotiation support, it would be useful to think of the negotiation situations along two dimensions. One relates to the willingness of the negotiators to disclose their private preferences to a third party, which promises to make the negotiation process more efficient. The other one relates to the degree of certainty regarding negotiator preferences and strategies (*i. e.* the degree to which the negotiator's task can be regarded as being "structured"). The types of agents that specifically suit e-negotiation tasks are described below.

- *User profile agent*. The purpose of this type of agent is to elicit user preferences, and to assist the negotiator in deciding on objectives and strategies. Ideally agents of this type would be able to adapt to the changes in user behavior in the process of negotiations.

- *Information agent*. Agents of this type would engage in actively seeking, retrieving, filtering, and delivering information relevant to the issues on the table.

- *Opponent profiling agent*. The primary purpose of this agent type would be to identify the objectives, preferences and strategies of the opponent. Knowing the opponent better renders offer generation and evaluation a much better informed decision-making process. The information and opponent profiling agent could be regarded as "intelligence" agents.

- *Proposer agent*. The aim of this type of agent is to generate a set of promising offers to be considered for submission to the opponent. In negotiation problems that involve multiple issues, the generation of an offer may involve search in a very large space of possible offers.

- *Critic agent*. The purpose of the critic is to evaluate the offers received from and addressed to the opponent and provide "verbal" feedback on the drawbacks and, possibly benefits of these offers. The proposer and critic agents could be regarded as a type of "adviser" agents.

- *Negotiator agent*. This agent may be capable of conducting negotiations by itself in a semiautonomous or fully autonomous fashion. Applicability of full automation depends on the degree of certainty in objectives, preferences, and tactics of the negotiator (*i. e.* the level of structuredness of the negotiation task from the negotiator's perspective).
- *Mediator agent*. The main purpose of this agent is to coordinate the activities of the negotiating parties, and to attempt to generate mutually beneficial offers. The role of this agent increases when the parties are willing to provide their information to a third party agent.

In order to identify the tasks that could be potentially delegated to agents we have to revisit the phase model of negotiations and find out which activities are potentially amenable to automation. The major activities of the planning phase include formulation of negotiation problem, including the specification of issues, options, BATNA, reservation levels, and negotiation strategies. Any important knowledge about the opponent would help to better prepare oneself for negotiations. Agents can assist the negotiators by finding information about the markets related to negotiations (recent deals, prices, *etc.*), capturing user preferences (*e. g.* through use of such techniques as conjoint analysis), helping the negotiator define the adequate negotiation strategy, and profiling an opponent (inferences about opponent based on background information).

The second phase includes agenda setting, exploring the domain, and discussion of negotiated issues, negotiation protocol, timing of exchanges and the deadline. Agent contributions are limited in this phase as this involves mostly human-directed activities. One possible application is for the mediator agents to obtain negotiator preferences, try to match them and send messages to negotiators about the acceptable set of issues, time, deadline, *etc.*

Exchanging offers and arguments and, possibly reaching an agreement are the main "action" phases in negotiations. In these phases, information agents deliver up-to-date information to advise the negotiator about market prices, deals, *etc.* A critic agent can evaluate offers received from the component and provides its opinion to the negotiator on the acceptability of offers. It would also watch over the shoulder of the user when the user prepares an offer. This agent can interfere to criticize the offer in regard to its alignment with user's interests, strategy, and current market situation.

The proposer agent would help formulate offers by generating a set of different promising offers in accordance with current preferences and strategy selected by the user. A user-profile agent would adjust the user profile by watching the actual offers chosen or formulated by the user. Opponent profiling agents likewise would update the opponent profile by watching opponent's moves. The mediator may also watch the exchanges and learn about the parties' preferences. The mediator agent could get involved if parties agree to submit their preferences to the mediator that would do the matching and generate candidate agreement packages to the negotiators. Finally, if included, a negotiation agent could take over the negotiation process itself and interact with the opponent in an autonomous mode.

Table 15.2. Agent-supported tasks

Agent type	Negotiation phases			
	Planning	**Agenda setting**	**Conducting**	**Concluding**
User Profile	Eliciting user preferences, helping with the choice of a strategy		Tracking user behavior, maintaining user preferences	Updating and storing user preferences
Information	Delivering relevant information for planning		Delivering latest information relevant to an ongoing exchange	
Opponent profile	Deriving an initial profile of an opponent		Tracking and updating opponent profile	Updating and storing opponent profile
Proposer			Generating promising candidate offers	
Critic			Evaluating and critiquing offers and counteroffers	
Negotiator			Conducting negotiations (well-structured tasks)	
Mediator		Coordinating negotiation issues, protocols, settings	Generating set of potentially acceptable agreement alternatives (private information disclosed)	Offering possible improvements

In the concluding phase of the negotiations the mediator agent may analyze the estimated utility of an agreement and perhaps propose a few more alternatives to the negotiators if it finds potential room for improvement. Table 15.2 summarizes support provided by agents in different phases of negotiations.

Figure 15.2 shows the generic architecture for an agent-enhanced e-negotiation system. Analytical models and local data are included in the "toolbox" part of the system. These tools are used by the variety of agents in order to carry out their tasks. Information retrieved by the information agent can be used by different agents to assist in setting objectives, generating alternatives, and critiquing offers. Most of these activities could be also assisted by opponent profile information.

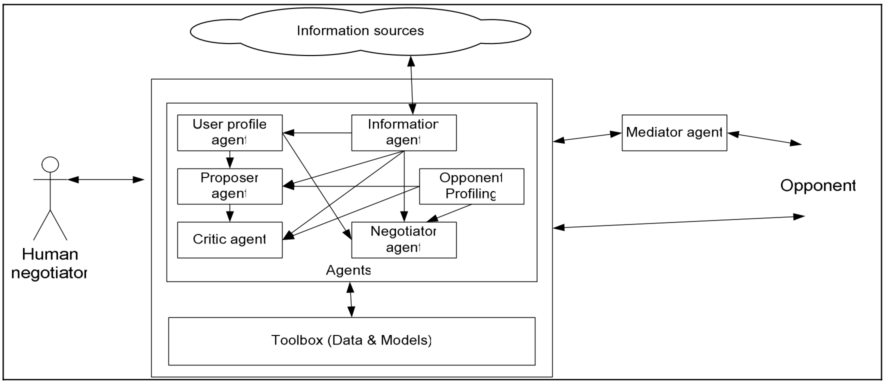

Figure 15.2. Generic architecture for agent-enhanced e-negotiation system

15.5 Models for Negotiation Software Agents

In this section, we describe several models proposed for negotiation software agents over recent years. They vary from decision-making models of negotiation to learning methods for supporting the negotiation, based on a variety of techniques including: probabilistic decision theory, possibilistic decision theory, Bayesian learning, possibilistic case-based reasoning, constraint-based reasoning, heuristic search, Q-learning and evolutionary computation.

Various techniques can be applied to automate the negotiation process and its selected activities. The first approaches were based on game theory. However, game theory makes a number of assumptions including knowledge of circumstances (Jennings 2001). This means that we should know rules of the encounter, specify our preferences, and know our partners' preferences or at least be able to formulate beliefs about their preferences. Another assumption of game theory is the full rationality of negotiators, which means that agents have sufficient reasoning and computational capacity to maximize their expected payoffs given their beliefs. Because of these drawbacks, we do not consider game-theoretic approaches in the remainder of this section.

In general, negotiation software agents require an appropriate protocol, specification of negotiation objects, and an apparatus for decision making. An interaction protocol (Binmore and Vulkan 1999) defines rules of negotiation, for example, who is allowed to say what and at which time. It defines roles and actions that negotiators can take at each moment of the negotiation process. Negotiation is usually a multistage process. Some protocols allow agents to submit proposals simultaneously, for example the monotonic concession protocol (Rosenschein and Zlotkin 1994). Other protocols allow for iterative exchange of proposals, for example the iterative negotiation protocol (Stahl 1972). In both types of protocols, an agreement is reached if one agent matches what the other one asked for, that is the agents agree on the same terms.

The object of negotiation may be a service or commodity and it is usually characterized by a number of attributes that are also called negotiation issues. The operators that can be performed on the attributes are defined on these objects, for example to change their values and add new issues to a negotiation.

The decision-making apparatus defines a model for decision making and strategies used by an agent during negotiation. Depending on what knowledge is available and the circumstances of the particular encounter, different apparatuses for decision making can be designed.

Real-life negotiation problems are typically ill-defined, information is not equally distributed among the participants, the participants have only partial knowledge about their counterparts and communication is often ambiguous or imprecise. Methods of artificial intelligence (AI) are particularly useful in problems, in which knowledge about partners' types and full rationality cannot be assumed. In such approaches negotiation agents can use AI-based decision-making mechanisms that satisfy the bounded information, bounded rationality, and bounded computational characteristics of agents (Binmore 1992, Rubinstein 1998). The lack of knowledge about partners' types can be compensated by the agents' ability to learn about and verify the acquired information. Agents need to be able to update their knowledge about their partners as well as the environment. This capability is the prerequisite for negotiating agents to be able to adapt their behavior to cope with changing partners and changing user preferences.

Given the negotiation problem two kinds of knowledge can be distinguished: knowledge of the opponent and knowledge of oneself. Considering approaches coming from game theory, knowledge of the problem can be reduced to the payoff tables. Knowledge about the opponents is required in order to be able to determine the moves (offers) they may accept and those they reject.

The acquisition of information such as counterpart's preferences, reservation price, or deadline, *etc.* allows us to increase knowledge about the partners. The second kind of knowledge is knowledge of oneself, which is used to select one strategy from the space of all possible strategies during the process of negotiation.

Strategy selection depends on the negotiator's objectives, preferences, and risk attitude. For a strategy to be effective it has to lead to a solution, which the negotiators' counterparts accept. This means that in constructing the set of possible strategies the counterparts' profiles have to be considered. The question that arises is: "How to learn about partners to devise adequate strategies?" Usually there are two kinds of information available, that is, the history of previous interactions and the behavior of an opponent during the current encounter. There is no universal method known for handling the available data to learn about partners' preferences in order to derive optimal moves during the negotiation process. Researchers apply and test various models of data acquisition and inference (Gerding *et al.* 2004). Below we present several models of decision-making and AI-based learning approaches for supporting negotiation. The models include probabilistic decision theory, possibilistic decision theory, constraint based reasoning, and heuristic search. The learning approaches for supporting the negotiation include: Bayesian learning, possibilistic case-based reasoning, Q-learning, and evolutionary computation. Table 15.3 summarizes the described approaches.

Table 15.3. Summary of negotiation approaches

Approach	**Characteristics**
Probabilistic DT	Selecting optimal decision
Possibilistic DT	Selecting optimal decision
Bayesian learning	Learning negotiation partner's type
Possibilistic CBR	Selecting most prospective negotiation partners
Constraint-based reasoning	Finding a solution that satisfies constraints of negotiating partners
Heuristic search	Determination of negotiation offer
Q-learning	Searching the set of potential strategies
Evolutionary computing	Searching the set of potential strategies

15.5.1 Probabilistic Decision Theory

In this model the uncertainty about the consequences of a decision $\boldsymbol{d}$ is modeled by a probability distribution $\boldsymbol{p_d} : \boldsymbol{S} \rightarrow [0,1]$ that assigns to each possible state a probability value. The decision makers preferences are encoded by the utility function $\boldsymbol{u} : \boldsymbol{S} \rightarrow [0,1]$ If the probability distribution is constructed for each of the possible decisions, then the expected utility can be calculated for each of these decisions (von Neumann and Morgenstern 1944):

$$EU(d) = \sum_{x \in S} p_d(x) u(x)$$

To maximize its outcome, an agent chooses the decision with the highest expected utility.

15.5.2 Possibilistic Decision Theory

In some situations the given information may not be sufficient to build a probability distribution. One of the alternative approaches is to employ possibility theory (Dubois and Prade, 1996). The basic notion of possibility theory is that of a possibility distribution, which is a counterpart of a probability distribution in the classical approach. The possibility distribution $\boldsymbol{\pi_d} : \boldsymbol{S} \rightarrow [0,1]$ assigns to each possible outcome a level of plausibility. The difference between probability and possibility is that the first notion is usually a measure of frequency of occurrence of an event whereas the second one is the measure of extent to which an event may occur. As in the previous approach the agent's utility function $\boldsymbol{u} : \boldsymbol{S} \rightarrow [0,1]$ also needs to be specified. The optimal decision can then be chosen according to the optimistic and pessimistic criteria

$$QU^{-}(\pi \mid u) = \min_{x \in S} \max(n(\pi_d(x)), u(x))$$
$$QU^{+}(\pi \mid u) = \max_{x \in S} \min(\pi_d(x), u(x))$$

15.5.3 Constraint-based Reasoning

Negotiation can be considered as a distributed constraint satisfaction problem (CSP) where the constraints of each party are partitioned between agents that exchange the coordination information in order to find a solution that satisfies all the constraints (Kowalczyk and Bui 2003, Sathi and Fox 1989, Yokoo 1998). The negotiation agents iteratively exchange their preferred solutions in the form of offers, and relax their preferences and constraints according to typically heuristic negotiation strategies, until all the constraints are satisfied and an agreement is reached.

The classical CSP considers constraints that can be precisely defined and fully satisfied (Yokoo *et al.* 1998), which may limit its applicability in many real-world negotiation problems, where preferences and constraints are imprecise and soft. These assumptions can be softened with a notion of fuzzy constraints that allow one to express the degrees to which the constraints are satisfied with different solutions and can be used to uniformly represent the constraints, preferences and objectives of negotiating parties (Kowalczyk 2000, Kowalczyk and Bui 2003, Kowalczyk 2002, Luo *et al.* 2003a, Luo *et al.* 2003b).

Fuzzy constraints are considered as fuzzy relations over and between the negotiation issues, and are represented by membership functions defining the degree of constraint satisfaction with the issues instantiations (possible agreements). An assignment satisfies a constraint fully if it is evaluated to 1 and violates a constraint when it is evaluated to 0. The intermediate values represent the degree of partial constraint satisfaction. For example, a fuzzy relation representing the constraints of an agent can be defined as follows:

$$C^j(x^j) = \bigwedge_{k=0,\dots,m_j} C_k^j(x^j)$$

Where $\boldsymbol{C^j(x^j)}$ is a fuzzy relation corresponding to the constraints $\mathbf{C^j} = \{ \mathbf{C^j_k} \}$, k = 1,.., $\boldsymbol{m_j}$ of the j^{th} agent, $\wedge$ is a conjunctive combination operator (*e. g.* a t-norm in the form of the min operator). The agents search for an agreement that satisfies the constraints of all the agents, that is:

$$C(x) = \bigwedge_j C^j(x),$$

through the exchange of their preferred solutions according to the level of constraint satisfaction. The search is typically guided by the negotiation strategies of each party, that is the rules for generation of offers (*e. g.* trade-off and/or concession) taking into account the information available to an agent including the individual preferences, constraints and objectives as well as the previous offers and counter-offers. The principles of fuzzy constraint-based reasoning can assist the search process by ordering and pruning the search spaces of the parties, and finding a solution that maximizes the agent's level of satisfaction subject to its acceptability by other agents (Kowalczyk 2002).

15.5.4 Heuristic Search

The strategic reasoning of a negotiating agent is usually computationally intractable. In such situations it can be supported in the search for the best strategy by some heuristic approaches.

Faratin *et al.* (1998) suggest approximating the rational choice of negotiation strategies with the use of decision functions. The idea is based on notions of heuristic strategies and tactics, which can be used by agents to calculate good proposals or counterproposals during the negotiation. The mechanism allows for the generation of offers and counteroffers in a responsive way, linearly combining simple functions that are called tactics. There are several factors substituting full rationality, which have to be taken into consideration during the negotiation. The agent has to consider deadlines and has to adapt to remaining time or any other resource constraints. The agent also has to be responsive, which means that it has to take opponent's behavior into account and this may be achieved by its imitation. Various levels of importance can be defined for different factors and these levels may also change during the negotiation because of agent's adaptation. Three types of negotiation tactics are distinguished: time-dependent tactics, resource-dependent tactics, and behavior-dependent tactics.

The time-dependent tactics are modeled by the negotiation deadline. The closer the deadline, the faster an agent concedes. A further distinction in this set of tactics can be achieved by varying function parameters. In the Boulware tactics model, the agent maintains the offered value most of the time and concedes up to the reservation value when approaching the deadline.[6] The conceder tactics are the opposite of the above tactics. Here, agents concede much in the beginning and quickly approach the reservation value.

The resource-dependent tactics are generated by the same family of functions as time-dependent tactics and constitute a broader set of tactics. Resource-dependent tactics can model any kind of resource, for example the number of negotiating agents.

Behavior-dependent tactics are the third group of tactics. These types of tactics are concerned with responsiveness to a partner's behavior during the negotiation and may be interpreted as an implementation of cooperation. They can imitate opponent's behavior in a variety of ways.

For each type of tactic there is a corresponding negotiation decision function that is used to calculate the value of the next concession. A decision function determines what should be the next offer taking into account a particular tactic. In a multi-issue scenario the decision functions can be considered separately for each issue. In some

---

6 This tactic has been introduced by Lemuel Boulware, Vice President of General Electric. After making an assessment of union strength in each organized facility, the company presented the same offer to all locals and resisted making any changes with the "take-it-or-leave" approach. Next, Boulware would approach the weakest local union and slightly improve the offer. After the weak union settled other unions were pressured to accept the same offer. They agreed on Boulware's offer because a single union could not organize a successful strike.

applications time can be the most important factor (Fatima *et al.* 2002). A negotiation decision function, $\boldsymbol{F^a}$ may be defined as follows:

$$F^a(t) = k^a + (1-k^a)(\frac{\min(t,T^a)}{T^a})^{\frac{1}{\varphi}}$$

where: $\boldsymbol{k^a}$ is a constant for agent $\boldsymbol{a}$ determining the value of attributes to be offered as the first proposal, $\boldsymbol{T^a}$ is the deadline of agent $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ determines the type of time-dependence (Boulware: $\boldsymbol{\varphi} < 1$, conceder: $\boldsymbol{\varphi} > 1$).

The offer proposed by an agent a to an agent $\hat{a}$ can be determined using the following formula:

$$p^t_{a\to\hat{a}} = \begin{cases} P^a_{\min} + F^a(t)(P^a_{\max} - P^a_{\min}) & for \quad a = b \\ P^a_{\min} + (1-F^a(t))(P^a_{\max} - P^a_{\min}) & for \quad a = s \end{cases}$$

where $[P^a_{\min}, P^a_{\max}]$ is the range of attribute, b denotes buyer and s denotes seller.

15.5.5 Bayesian Learning

The Bayesian learning model enables updating the knowledge or beliefs of one agent about other agents (Zeng and Sycara 1997, Zeng and Sycara 1998). Before negotiation starts an agent acquires knowledge. This knowledge consists of the information about the environment and information about other players, which can be gained from various sources, such as previous experience, second-hand knowledge, or rumours. In the Bayesian approach this knowledge is encoded in a form of subjective probability distributions. We can have beliefs about environment parameters such as product supply and demand, or interest rates. As far as other players are concerned we can have beliefs about their utility function, reservation prices, deadlines or even negotiation style.

During the negotiation encounter, our agent has to use this feedback by updating its subjective beliefs about the environment and other players after every move of other participants. This stage is performed using the Bayesian rules. First, prior knowledge about the probabilities of hypotheses $\boldsymbol{H_i}$ is given ($\boldsymbol{i=1,\dots,n}$). In other words, we have a prior probability distribution over the set of hypotheses. Then we need a new piece of evidence (new event denoted e) that can be derived from the action performed by other agents. Also, some conditional probability is needed stating how likely an event $\boldsymbol{e}$ is to occur given that the hypothesis is true. We update the prior distribution and obtain a posterior distribution according to the formula as follows:

$$P(H_i \mid e) = \frac{P(H_i)P(e \mid H_i)}{\sum_{k=1}^{n} P(e \mid H_k)P(H_k)}$$

where $\boldsymbol{P(H_i \mid e)}$ is the new probability of the hypothesis $\boldsymbol{H_i}$ assuming that a new event $\boldsymbol{e}$ occurred. Having the updated distribution we can perform the next stage which is the best action selection. The action may be an acceptance or an appropriate counter-proposal that maximizes the expected utility given the information available at this stage

The Bayesian framework enables the modification of a given subjective probability distribution during the negotiation but does not give an answer to the question of how to obtain the distribution. Classical statistical methods of constructing the distribution may be applied but it usually requires a large amount of data. Instead of the quantitative method, a qualitative paradigm may be employed to address this problem. An example of such a paradigm is the possibility-based case-based reasoning.

15.5.6 Possibilistic Case-based Reasoning

This method allows for obtaining the opponent's likelihood towards agreement in the form of possibility distribution based on past experience. The reasoning from previous cases may be performed through a possibilistic rule stating that: "The more similar the situations are the more possible are similar outcomes." (Dubois *et al.* 1998). This can be expressed by the following formula:

$$\mu(y) = Max_{(s^i, o^i) \in H} S(s^t, s^i) \otimes P(o^i, y)$$

where $\boldsymbol{S}$ and $\boldsymbol{P}$ are the similarity relations for situations and outcomes respectively, $\otimes$ is the $\boldsymbol{T}$-norm[7], $\boldsymbol{H}$ is the history of previous cases, $\boldsymbol{s^t}$ is the current situation, $\boldsymbol{i^t}$ is the situation $\boldsymbol{i}$ and $\boldsymbol{o^i}$ is the outcome of situation $\boldsymbol{i}$. The obtained function $\boldsymbol{\mu}$ is modified to a monotone function $\boldsymbol{\pi}$ corresponding to some decision. This function is aggregated with a utility function $\boldsymbol{u}$ in order to determine optimal decision in a similar way as described in the section about possibilistic decision theory.

Negotiation may be quite expensive and time consuming, especially in scenarios with a large number of agents. In such situations it is important to determine with whom there is a higher chance of successful negotiation and reaching better agreements. The possibility-based mechanism can predict the ordering of potential partners by placing the most prospective partners for negotiation at the top and the less prospective further in the ordering. This allows choosing from the whole set of all the agents a subset of the most prospective ones for negotiation.

Brzostowski and Kowalczyk (2005) presented a scenario in which reasoning is done by the main contractor who is offered services from a number of agents. The contractor may use them individually or aggregate them as a compound service. From the set of agents representing services we need to choose the subset of most prospective agents. In order to do this we model the system of all potential partners using the tools of possibility theory.

We noted previously that the obtained possibility distribution describes the likelihood of successful negotiation and is derived from the history of previous interactions. The distribution encodes the prediction of the main contractor about preferences of his negotiation partners. Based on this function and the utility of the main contractor the estimation of the outcome of current negotiation can be

[7] An example of T-norm that was used here is the min operator.

calculated. In order to do so the calculation of the qualitative expected utility is required; it is obtained by the aggregation of possibility distribution and the main contractor's utility function. The estimation of negotiation outcome allows us to rank the negotiation partners. The final ordering gives the information with whom to negotiate first and with whom to negotiate later.

The mechanisms described above treats the uncertainty about attributes or negotiation outcomes and are suitable in situations when prediction is required because the negotiation partner does not want to reveal his private information. However, information revelation occurs in some real world problems. Such problems may be: meeting scheduling, planning or resource allocation. In such scenarios the agent does not need to learn because the knowledge about partners' preferences is given, although it may sometimes contain some uncertainties. For solving such negotiation encounters the constraint based reasoning may be used.

15.5.7 Q-learning

The multiagent system SMACE (Oliveira and Rocha 2000) combines the idea of decision functions and reinforcement learning algorithms into a new approach called Q-learning. An agent that uses reinforcement learning takes actions in a dynamic environment and is rewarded or punished depending on the consequences of actions taken. Learning agents have to explore the environment by performing actions. An agent receives feedback from the reward function and based on this feedback, learns which actions should be carried out in which states. Q-learning is an example of reinforcement learning based on the update of Q values.

Faratin *et al.* (1998) defined the agent's current action (counterproposal made by agent a to agent b at time t) in p-issue and m-tactics negotiation as the matrix of weights:

$$\Gamma_{a \to b}^{t} = \begin{pmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2m} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \omega_{p1} & \omega_{p2} & \cdots & \omega_{pm} \end{pmatrix}$$

where ω_{ij} is a weight corresponding to a negotiation issue i and tactic j. The strategy in their approach is a function f mapping the action $\Gamma_{a \to b}^{t_n}$ in time t_n and the agent's mental state $MS_a^{t_n}$ in time t_n to the new action $\Gamma_{a \to b}^{t_{n+1}}$ for time t_{n+1}:

$$\Gamma_{a \to b}^{t_{n+1}} = f(\Gamma_{a \to b}^{t_n}, MS_a^{t_n})$$

But in such a model the question of specifying function f remains open. The Q-learning may be regarded as a complement to this model because it allows learning and updating of the so-called ***Q*** values. The ***Q*** values are some kind of rewards or utilities assigned to each pair of action and state ***Q***(***i***,***a***). At first, the optimal action has to be determined by using ***Q*** values acquired so far. The chosen action should maximize the expected utility. After determining an appropriate action the ***Q*** value can be updated. The whole process may be described by the following formula:

$$Q(i,a) = Q(i,a) + \alpha[r(i) + \gamma \max_{a'} Q(j,a') - Q(i,a)]$$

where ***α*** is a learning rate, ***r(i)*** is a reward gained by performing action ***a*** in state ***i***, γ is a discount parameter, ***j*** is the state attained. The reward may be positive or negative depending on whether the action gives good or bad results. The results are the achieved deals and their utilities.

The state in the negotiation scenario may be described by such factors as the number of negotiating agents and time left for negotiation. The action is encoded by the sequence of weights corresponding to the applied tactics.

An agent applying the mechanism described above is able to improve its performance by using experience to learn what tactics should be employed in what situations. However, the main disadvantage of this approach is that the knowledge acquisition process requires many trials. Q-learning also requires the determination of balance between trying new actions and applying the old ones that already proved to be good. For more details on Q-learning see Russel and Norvig (2003).

15.5.8 Evolutionary Computing

Another trial-and-error approach for learning good strategies is evolutionary computing. Evolutionary algorithms enable searching the space of potential solutions by applying the principle of natural law stating that fit parents would most likely produce fit children in the process of reproduction. The candidate solutions are called chromosomes. The search starts by creating a first random population of chromosomes chosen from the space of potential solutions. The next generation is created in two steps. During the first step which is the recombination, the chromosomes from the previous generation are paired two-by-two and "crossed over". The second step is mutation - the change of some part of the chromosome. This operation models errors occurring while copying genes from the previous generation.

The object to be encoded as chromosome is the agent's strategy. The first paper applying evolutionary computation for negotiation automation (Oliver 1997) had been published before the idea of decision-functions-based strategies was proposed. Therefore, the notion of strategy in this paper is defined in a much simpler way as a threshold decision rule. An agent applying this rule accepts an offer in the first step, which exceeds some threshold T_1. If the threshold is not exceeded, the agent makes a counterproposal. If the opponent does not agree on this proposal it makes a subsequent proposal. Our agent accepts this proposal if it exceeds the next threshold T_2. Again, if the opponent does not agree it makes a counterproposal and the process continues until an agreement is reached or one of the sides stops the negotiation. The strategy defined in this manner is encoded as a sequence of thresholds and counterproposals.

The learning process is done in the following way. For both negotiating agents (we consider bilateral negotiation here) the random population of candidate solutions is generated. Both sides select strategies from their populations which are then tested in the negotiation process. After negotiation, agents assign fitness to the tested strategies according to their performance during the encounter. The selection and test is carried out a number of times close to the number of strategies in the population, so that each strategy is chosen approximately once. Having the fitness assigned to population members the new population is created using a genetic algorithm and the process of testing is done again. The higher the fitness the higher the chance that a strategy will be chosen for reproduction. The whole process of population update and testing is repeated until an exit condition is satisfied.

The other papers dealing with the evolutionary computation approach for negotiation usually apply similar mechanisms of learning to that described above but the notion of strategy is more complex. Matos *et al.* (1998) encodes in the chromosome information like deadlines, domains of each attribute, monotonicity of each attribute, weights of all tactics and parameters specifying each tactic. Some reproduction mechanisms may be more sophisticated. Gerding *et al.* (2004) use the same notion of strategy as Olivier that is the sequence of thresholds and counter-proposals. The main difference is the application of mutation as a reproduction operator in this case. The recombination is not used because the authors claim that it does not have a large influence on evolving system.

15.6 Applications of Negotiation Software Agents

Agents can be applied to a variety of negotiation problems in e-commerce, planning, resource allocation, scheduling, and so on. Auctions, in addition to negotiations, have been recently widely used in resolving these these problems. In general, auctions are considered as “a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants” (McAfee and McMillan 1987). Although traditionally auctions have been considered distinct from negotiations, recent changes in auction mechanisms allowed their use for resolving more types and domains of problems (Ströbel and Weinhardt 2003).

One of the challenges in designing a negotiation software agent for one-sided auctions is the ability to join multiple auctions. By participating in many auctions the agent can purchase the required number of goods for the low price. Preist *et al.* (2000, 2001) describe agents that enable the identification of the most beneficial auctions (closing with low price) and the coordination of bidding in these auctions in order to win the lowest possible price. In this approach an appropriate coordination algorithm allowing the purchasing of the right number of goods is needed. The proposed learning mechanism allows for the construction of the belief function in which the probability that some number of participants value the good with valuation higher than some specific value is included. Based on this belief function an agent can decide whether to bid higher in the terminating auction or to place bids in the remaining auctions. This is done using a comparison of calculated expected utilities for these auctions.

Apart from monitoring auctions and selecting the ones to participate in, the decision making concerning how to bid remains a difficult problem. This problem is addressed by Anthony *et al.* (2001, 2003). The authors propose the design of bidding strategies based on decision functions (Faratin 2000). The marketplace is simulated with various types of one sided auctions. The current maximum bid is determined considering the bidding constraints such as: time left, remaining auctions left, the participant's desire to bargain and participant's level of desperateness. Based on the value of the current maximum bid potential, auctions are selected and the bid for each of these auctions is calculated. Then the auction and corresponding bid with the highest expected utility is selected. The authors also search the set of bidding strategies offline, using genetic algorithms in order to determine the best strategies.

Byde *et al.* (2002) developed a sophisticated decision theoretic framework that enables agents to bid rationally across multiple auctions. The framework is described for a few types of auctions. The rational agent will bid in an auction if the expected future utility of bidding exceeds the expected future utility of not bidding. To make this decision, agents need to estimate the future utility first. As a solution to this the authors propose two alternatives: backward induction or fixed auction strategies. Finally a heuristic algorithm allowing an agent to make appropriate decisions is described.

One interesting approach is described by Garcia *et al.* (1998) in which possibility-based decision theory is employed to calculate the best bid. The uncertainty about opponent's behavior is modeled by a possibility distribution that is obtained by case-based reasoning. This possibility distribution enables making decisions about the bidding strategy.

The traditional setting of auctions may be extended by introducing new attributes other than just the price of the goods under consideration. The multiattribute English auction is considered by Dawid *et al.* (2003). A scenario with one buyer and multiple sellers is presented. The situation here is more complex than in a single-attribute auction. Due to the multidimensionality, the utility function for each buyer and seller has to be specified. The seller specifies his requirements by announcing the scoring function, the minimum increment and the maximum number of rounds. Two types of auctions are described: sequential and simultaneous. The seller's bidding strategy is determined as an action maximizing the expected utility. In the continuous double auction both sellers and bidders submit their proposals and the process is stopped when the offer of one party meets the offer of another one. There are various types of strategies applied in this kind of auction. The continuous double auction is more efficient and flexible than the one-sided auctions. The mechanisms deciding what bid to make vary from very simple to very sophisticated.

One of the first approaches to bidding involved agents that used a "zero intelligence" strategy (Gode and Sunder 1993). This strategy generated a random bid within the allowed range. The "zero intelligence" strategy turned out to be quite efficient when compared with other, more intelligent strategies. Subsequently, other complex decision-making strategies have been proposed. Park *et al.* (1999) proposed an adaptive agent bidding strategy based on stochastic modeling. The authors claim that stochastic modeling is a good substitute for full rationality, but because of the computational costs and time consumption it should be decided in what situations to use it. Therefore the agent should be adaptive in the sense that it uses the appropriate mechanism when it is necessary.

Dynamic programming was also applied in auctions (Tesauro and Bredin 2002). They developed an algorithm for both sides: buyer and seller participating in CDA, based on fuzzy logic. They used heuristic fuzzy rules and fuzzy reasoning to calculate the optimal bid given the current state of the market. The agent based on this approach can also adapt its bidding behavior to changes occurring in the environment.

In the following section, we will give an overview of several applications that enable intelligent agents to negotiate and take part in electronic auctions. AuctionBot (Wurman *et al.* 1998) was a project at the University of Michigan to develop a flexible, scalable, and robust online auction server for many auction types. AuctionBot provides the service of hosting and processing an auction according to user's preferences via a Web interface (for human users) and an Application Programming Interface (for software agents). AuctionBot supports the widest possible range of different auction types by decomposing the auction design space to different parameters (Wurman *et al.* 2001), for example the number of buyers and sellers to participate in the auction, closing conditions, and the allocation policy. Users can create new auctions by specifying those parameters. Buyers and sellers can bid according to the auction rules using the Web interface or by allowing the software agents to bid on behalf of a user using the programming interface. AuctionBot was used to host the first two Trading Agent Competitions in 2000 and 2001.

Kasbah (Maes 1998) is an online virtual marketplace where users can create agents to buy and sell goods on their behalf. Kasbah is a multi-agent system, in which agents must act and communicate according to a specified protocol. When a user creates a selling agent, he gives a description of the item to sell. In addition, the user must specify parameters on a very high level of abstraction, such as the desired date to have the item sold, the desired price, and the lowest acceptable price. In addition, the user has control over the agent's negotiation strategy, that is, the user can specify the decay function (linear, quadratic, cubic) in order to lower the price and time. The agent can be specified to ask its owner before finalizing a deal. All these parameters can be changed by the user at any time after the agent has been created. The definition of a buyer agent works analogously.

The selling agent then proactively searches for other agents that are interested in purchasing this item and starts the negotiation process, which works straightforwardly. After a selling agent has found a buyer agent interested in the offered item, buyer agents are allowed to offer a bid to selling agents without any further restrictions regarding price, time, *etc.* Selling agents only reply with either "yes" or "no". Once a buying agent and a selling agent have reached an agreement on a specific price, both users are asked for their respective approval.

Given this protocol, Kasbah users can actually select from three different buying and selling strategies, respectively. If the user selects for example a linear increasing function for a buying agent, he or she follows an anxious negotiation strategy. This is due to the fact that the user must increase offers quickly in order to be able to win the negotiation. According to Maes (1998) the simplicity of this negotiation process is necessary for user understanding and user acceptance.

Whereas in Kasbah the only negotiable attribute is price, MIT's Tête-à-Tête project (Maes and Guttman 1999) provides buying and selling agents to cooperatively negotiate across multiple terms of a transaction, for example, warranty, delivery time, return policy, and other merchant value-added services. Based on bilateral argumentation, Tête-à-Tête's negotiation process works on XML documents that describe proposals, critiques, and counterproposals, thus making the negotiation process more complex than Kasbah's one. A buying agent receives proposals from multiple selling agents and evaluates them according to the user's multiattribute utility functions. If the agent is not satisfied with proposals, it can critique them using one or many attributes and broadcast this to the selling agents. After receiving critiques, selling agents can then use them to create better counterproposals. Using critiques, selling agents can place constraints on product features in order to influence the decision of whom to buy from and what to buy.

A slightly different negotiation protocol is implemented by Magnet (Collins *et al.* 1998) in which agents are used to negotiate contracts and later monitor their execution. First, a customer (buying agent) publishes a Request for Quotes. Then, suppliers (selling agents) respond by providing an offer detailing the price of the requested resource over a specified time period. Customers evaluate bids looking at price, risk, and time constraints and finally select the optimal set of bids that satisfy their goals. Suppliers are notified about the result. Second, an execution manager component is initiated to monitor the fulfillment of the contract and start re-negotiation process if necessary.

e-Negotiation agents (eNAs) and fuzzy e-negotiation agents (FeNAs) (Kowalczyk 2002) are prototypical intelligent trading agents that autonomously negotiate multiple terms of transactions in e-commerce trading. These agents engage in integrative negotiations in the presence of limited common knowledge about other agents' preferences, constraints and objections through an iterative exchange of multiattribute offers and counteroffers. Fuzzy eNAs allow the specification of fuzzy constraints and preferences. The FeNAs environment consists of many autonomous trading agents representing buyers and sellers that can engage in concurrent bilateral negotiations according to a number of user-selected negotiation strategies. The eNAs and FeNAs agents have been demonstrated with a number of test-beds of e-commerce trading (Kowalczyk and Bui 2003).

Kersten and Noronha (1999) proposed negotiation software agents that provide information and knowledge (*e. g.* statistics and inferences) about past negotiations, scan the negotiation transcripts and other process descriptions, and then provide a comparison of situations, interests and issues of past problems to the current problem. These agents may also receive knowledge from various sources, such as other agents, the environment, user input and databases, then interpret and understand that knowledge and intelligently use information to assist the negotiator throughout the negotiation processes (Torsun 1995).

Kersten and Lo (2003) developed Aspire, a Web-based system comprising software agents, and negotiation and decision support systems. Aspire's functionalities include supporting negotiators, providing context-dependent advice, and undertaking certain activities autonomously. A software agent monitors the process, facilitates the use of the Web-based negotiation support system, interprets the negotiators' activities and provides methodological advice. The architecture of the system separates user support functions from the autonomous software activities, separation of the support for individuals from facilitation and mediation, scalability and the ability to provide linkages with the existing software.

eAgora (Chen *et al.* 2004) is an e-marketplace that allows buyers and sellers to engage in multi-issue negotiations. Its services include a software agent that generates and critiques offers. A usability test for comparing negotiations with and without the agent, found the agent's services were useful and partial negotiation automation is desired.

For projects focusing on mobile networks, in which for example mobile agents are used as user representatives in online auctions, we refer to Agora (Fonseca *et al.* 2001), Impulse (Youll *et al.* 2000), MAgNET (Dasgupta 2002), and BiddingBot (Fukuta *et al.* 2001).

15.7 Conclusion

In this chapter, we discussed research on e-negotiation systems and presented theory and applications of software agents for electronic negotiations. Types of negotiation agents and their roles and requirements were discussed and various models for negotiation software agents were reviewed. We also presented applications of negotiation software agents.

We conclude this chapter by emphasizing that software agent technologies should be regarded as tools for effective support of negotiations. The research questions should focus on the problems of the user (or principal), not the capability of the agents and availability and feasibility of technologies. Negotiations are in many cases ill-structured problems that require human ability to reformulate the issues, redefine the negotiation process, understand participant's interests, and develop strategies and tactics. However, most research on the use of software agents in negotiations has focused on automation of the communication and decisionmakings in the negotiation process. This approach can only fit into negotiation processes involving well-structured problems where human learning and socialization attempts to build business or other relationships do not have significant effects.

DSSs and NSSs have focused on ill-structured negotiation problems that take shape and have issues clarified during the negotiation through human intervention as well as well-structured problems. The negotiation software agents have advantages in automating well-structured problems. From our point of view, negotiation software agents may take over well-defined and structured activities in a negotiation but it is not necessary for agents to handle all the tasks. For example, the agent may present offers, seek information about similar negotiation situations, collect information about the counterparts, and alert the principal if predefined conditions are violated. The ill-defined and ambiguous issues, decisions regarding the relationship between the parties, modification of the rules and parameters are better left to the principals. Therefore, we believe that it is important to first consider the effective mixture of both autonomous agents and DSS/NSS.

In this chapter, we approached software agents and e-negotiation systems from the perspective of the hybrid NSA/DSS/NSS architecture that allows for human-system-agent interactions. Such an integrated architecture allows utilizing the strengths and capabilities of the methods and models that are embedded in the support systems and software agents. It also allows us to better define the roles of the individual components, the collaboration patterns, and the scope and levels of the agents' autonomy and the systems support.

Acknowledgements

This work has been partially supported by the grants from the Department of Education, Science and Training of Australia (AU-DEST-CG060081), the Social Sciences and Humanities Research Council of Canada (SSHRC), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Anthony P, Hall W, *et al* (2001) Autonomous Agents for Participating in Multiple Online Auctions. Proceedings of the IJCAI Workshop on E-Business and Intelligent Web, ACM

Anthony P, Jennings NR (2003) Developing A Bidding Agent for Multiple Heterogeneous Auctions. ACM Transactions on Internet Technology 3(3): 185-217

Binmore K (1992) Fun and Games, a Text on Game Theory, D. C. Heath and Company.

Binmore K, Vulkan N (1999) Applying Game Theory to Automated Negotiation. Netnomics 1(1): 1-9

Brzostowski J, Kowalczyk R (2005) On Possibilistic Case-based Reasoning for Selecting Partners in Multi-Agent Negotiation. Proceedings of The 17th Joint Australian Conference on Artificial Intelligence, Cairns, Australia, Springer (in print).

Bui T, Yen J, et al (2001) A Multi-Attribute Negotiation Support System with Market Signaling for Electronic Markets. Group Decision and Negotiation 10(6): 515-537

Byde A, Preist C, *et al.* (2002) Decision Procedures for Multiple Auctions. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: Part 2, ACM, New York

Caglayan A, Harrison C (1997) Agent Sourcebook. New York, John Wiley & Sons, Inc.

Chen E, Kersten GE, et al (2004) Agent-supported Negotiations on E-marketplace. International Journal of Electronic Business: (forthcoming).

Collins J, Tsvetovat M, *et al.* (1998) MAGNET: A Multi-Agent Contracting System for Plan Execution. Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice, Albuquerque, USA, AAAI Press

Cybersettle (2000) White Plains, Cybersettle.com. 2004

Dasgupta P. (2002) Fault tolerance in MAgNET: A Mobile Agent E-commerce System. Proceedings of the Sixth International Conference on Internet Computing, Las Vegas, USA

David E, Azoulay-Schwartz R, *et al.* (2003) Bidders' Strategy for Multi-Attribute Sequential English Auction with a Deadline. Proceedings of the second international joint conference on Autonomous agents and multi-agent systems, Melbourne, Australia, ACM

Dubois D, Prade H (1996) Combining hypothetical reasoning and plausible inference in possibilistic logic. Multiple Valued Logic 1: 219-239

Dubois D, Esteva F, et al (1998) Fuzzy set modelling in case-based reasoning. International Journal of Intelligent Systems 13: 345-373

Ehtamo H, Kettunen E, et al (2001) Searching for Joint Gains in Multi-party Negotiations. European Journal of Operational Research 130: 54-69

Faratin P (2000) Automated Service Negotiation Between Autonomous Computational Agents. London, U.K., Department of Electronic Engineering, Queen Mary & Westfield College

Faratin P, Sierra C, *et al.* (1998) Negotiation Decision Functions for Autonomous Agents. Robotics and Autonomous Systems 24(3-4): 159-182.

Fatima S, Wooldridge MJ, et al (2002) The Influence of Information on Negotiation Equilibrium. Agent-Mediated Electronic Commerce IV, Designing mechanisms and systems, Springer. 2531: 180-193.

Fisher R, Kopelman E, et al (1994) Beyond Machiavelli. Tools for Coping with Conflict. Cambridge, MA, Harvard University Press.

Fisher R, Ury W, et al (1991) Getting to Yes: Negotiating Agreement Without Giving In. New York, Penguin Books.

Franklin S, Graesser A (1997) Is it an Agent, or Just a Program? A Taxonomy for Autonomous Agents. Intelligent Agents III: Agent Theories, Architectures, and Languages. J. P. Muller, M. J. Wooldridge and N. R. Jennings. Berlin, Springer Verlag: 21-36.

Fukuta N, Ito T, et al (2001) A Framework for Cooperative Mobile Agents and its Case Study on Bidding Bot. JSAI 2001 International Workshop on Agent Based Approaches in Economic and Social Complex Systems.

Garcia P, Gimenez E, et al (1998) Possibilistic-based Design of Bidding Strategies in Electronic Auctions. Proceedings of the 13th European Conference on Artificial Intelligence, Brighton, UK, Wiley

Gerding EH, Somefun DJA, *et al.* (2004) Multi-Attribute Bilateral Bargaining in One-to-Many Setting. Amsterdam, Centre for Mathematics and Computer Science (CWI)

Gode D, Sunder S (1993) Lower Bounds for Efficiency of Surplus Extraction in Double Auctions. The Double Auction Market: Institutions, Theories, and Evidence. D. Friedman and J. Rust. Reading, MA, Addison-Wesley: 199-219

Gulliver PH (1979) Disputes and Negotiations: A Cross-Cultural Perspective. Orlando, FL, Academic Press

Harsanyi JC (1997) Rational Behaviour and Bargaining Equilibrium in Games and Social Situations. Cambridge, Cambridge University Press

Hiltz SR, Turoff M (1978) The Network Nation: Human Communication via Computer. Reading, MA: Addison-Wesley.

Inspire (1996). Inspire ENS. Ottawa, G. Kersten & InterNeg. 2004.

Jarke M, Jelassi MT, et al (1987) MEDIATOR: Toward a Negotiation Support System. European Journal of Operational Research 31(3): 314-334.

Jelassi MT, Foroughi A (1989). Negotiation Support Systems: An Overview of Design Issues and Existing Software. Decision Support Systems 5: 167-181.

Jennings NR, Wooldridge NM (1998) Applications of Intelligent Agents. In: N. Jennings and M. Wooldridge (Eds) Agent Technology: Foundations, Applications and Markets, Springer-Verlag

Jennings NR (2000) On agent-based software engineering. Artificial Intelligence 117(2): 277-296

Jennings NR (2001). An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM 44(4): 35-41

Jennings NR, Faratin P, et al (2001). Automated Negotiations: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2): 199-215
JointGains (2000). Joint Gains Negotiation Support in the Internet. Helsinki, System Analysis Laboratory. 2004
Keen PGW, Scott-Morton MS (1978) Decision Support Systems: An Organizational Perspective. Reading, Addison-Wesley
Kersten GE (1993) Rule-based Modeling of Negotiation Processes. Theory and Decisions 34(2)
Kersten GE (1997) Support for group decisions and negotiations. In: J. Climaco (Ed.) An Overview, in Multiple Criteria Decision Making and Support, Heidelberg: Springer Verlag
Kersten GE (1999) Decision-making and Decision Support. Decision Support Systems for Sustainable Development in Developing Countries. G. E. Kersten, Z. Mikolajuk and A. Yeh. Boston, Kluwer: 29-52
Kersten GE (2001) Modeling Distributive and Integrative Negotiations. Review and Revised Characterization. Group Decision and Negotiation 10(6): 493-514
Kersten GE, Lo G (2003) Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation. International Journal of Internet and Enterprise Management 1(3): 293-315
Kersten GE, Noronha S (1999) WWW-based Negotiation Support: Design, Implementation, and Use. Decision Support Systems 25: 135-154
Kersten GE (2004) E-negotiation Systems: Interaction of People and Technologies to Resolve Conflicts, UNESCAP Third Annual Forum on Online Dispute Resolution, Melbourne, Australia
Kim J, Segev A (2003). A Framework for Dynamic eBusiness Negotiation Processes. IEEE Conference on E-Commerce, New Port Beach, CA
Kinny D, Georgeff M, et al (1996) A Methodology and Modelling Technique for Systems of BDI Agents, Australian Artificial Intelligence Institute.
Korhonen P, Moskowitz H, *et al.* (1986) An Interactive Approach to Multiple Criteria Optimization with Multiple Decision-Makers. Naval Research Logistics Quarterly 33: 589-602.
Kowalczyk R (2000) On negotiation as a distributed fuzzy constraint satisfaction problem. Proceedings of the Third International Symposium on Soft Computing for Industry of the World Automation Congress
Kowalczyk R (2002) Fuzzy eNegotiation Agents. Journal of Soft Computing, Special Issue on Fuzzy Logic and the Internet 6(5): 337-347
Kowalczyk R, Bui V (2003) On Constraint based Reasoning in eNegotiation Agents. In: F. Dignum and U. Cortes (Eds) Agent Mediated Electronic Commerce III. Springer Verlag, pp 31-46
Kraus S (2001) Strategic Negotiation in Multiagent Environments. Cambridge, MIT Press.
Lempereur A (2004) Innovation in Teaching Negotiation. Toward A Relevant Use Of Multimedia Tools. International Negotiations Journal 9(1): 141-160.
Lewicki RJ, Saunders DM, *et al.* (1999) Negotiation. Boston, MA, McGraw-Hill.
Luck M, McBurney P, *et al.* (2003) Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent-Based Computing. AgentLink II. http://www.agentlink.org/roadmap/distribution.html
Luo X, Jennings NR, et al (2003) A Fuzzy constraint based model for bilateral multi-issue negotiation in semi-competitive environments. Artificial Intelligence 148(1-2): 53-102.
Luo X, Lee JH, et al (2003) Prioritised Fuzzy Constraint Satisfaction Problems: Axioms, Instantiation and Validation. Fuzzy Sets and Systems 136(2): 155-188.
Maes P (1998) Artificial Life Meets Entertainment: Life-like Autonomous Agents. Communications of the ACM 38(11): 108-114

Maes P, Guttman RH (1999) Agents that can buy and sell. Communication of the ACM 42(3): 81-91
Matos N, Sierra C, et al (1998) Determining Successful Negotiation Strategies: An Evolutionary Approach. Proceedings of the 3rd International Conference on Multi-Agent Systems, Paris, France
Matwin S, Szpakowicz S, et al (1987) Logic-Based System for Negotiation Support. Proceedings of the 1987 Symposium on Logic Programming, San Francisco, IEEE Computer Society Press
McAfee RP, McMillan J (1987) Auction and bidding. Journal of Economic Literature 25: 699-738
Moore D, Kurtzberg T, et al (1999) Long and Short Routes to Success in Electronically Mediated Negotiations: Group Affiliations and Good Vibrations. Organizational Behavior and Human Decision Processes 77(1): 22-43
Neumann D, Benyoucef M, et al (2003) Applying the Montreal Taxonomy to State of the Art E-Negotiation Systems. Group Decision and Negotiation 12(4): 287-310.
NovaForum (2000). The Electronic Courthouse. Ottawa. 2004
Nwana HS, Ndumu DT (1998) A Brief Introduction to Software Agent Technology. In: N. Jennings and M. Wooldridge (Eds) Agent Technology: Foundations, Applications & Markets, Springer Verlag: 29-47
Oliveira E, Rocha AP (2000). Agents advances features for negotiation in electronic commerce and virtual organisations formation process. In: C. Sierra and F. Dignum (Eds) European Perspectives on Agent Mediated Electronic Commerce, Springer. 1991
Oliver JR (1997) A Machine-Learning Approach to Automated Negotiation and Prospects for Electronic Commerce. Journal of Management Information Systems 13(3): 83-112
Park S, Durfee H, et al (1999) An Adaptive Agent Bidding Strategy based on Stochastic Modelling. Proceedings of the third annual conference on Autonomous Agents, ACM, New York
Preist C (2000) Algorithm Design for Agents Which Participate in Multiple Simultaneous Auctions. Hewlett-Packard, Bristol
Preist C, Byde A, et al (2001) Economic Dynamics of Agents in Multiple Auctions. Proceedings of the fifth international conference on Autonomous agents, ACM, New York
Purdy JM, Neye P (2000) The Impact of Communication Media on Negotiation Outcomes. The International Journal of Conflict Management 11(2): 162-187
Raiffa H, Richardson J, et al (2003) Negotiation Analysis. The Science and Art of Collaborative Decision-making. Cambridge, Harvard University Press
Rangaswamy A, Eliasberg J, et al (1989) Developing Marketing Expert Systems: An Application to International Negotiations. Journal of Marketing 53: 24-39
Rangaswamy A, Shell GR (1997) Using Computers to Realize Joint Gains in Negotiations: Toward an Electronic Bargaining Table. Management Science 43(8): 1147-1163
Rebstock M (2001). An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations. Electronic Commerce and Web Technologies. K. Bauknecht, S. K. Madria and G. Pernul. Berlin, Springer: 196-205
Rosenschein, JS, Zlotkin G (1994) Rules of Encounter. Cambridge, The MIT Press
Roth AE (1995) Bargaining: Economic Theories of Bargaining. Social Science Encyclopedia. A. Kuper. London, Routledge.
Rubinstein, A (1998). Modelling Bounded Rationality. Cambridge, USA, The MIT Press.
Russell, S. and P. Norvig (2003). Artificial Intelligence: A Modern Approach, Prentice Hall.
Sathi A, Fox MS (1989) Constraint-directed negotiation of resource reallocation. In: L. Gasser and M. Huhns (Eds.), Distributed Artificial Intelligence Volume II, pp 163-195, San Mateo, California, 1989. Morgan Kaufmann.

Schoop M, Quix C (2001) DOC.COM: A Framework for Effective Negotiation Support in Electronic Marketplaces. Computer Networks 37(2): 153-170.

Simon HA (1977) The New Science of Management Decision. Englewood Cliffs, N.J., Prentice-Hall.

Sprague RM, Carlson ED (1982) Building Effective Decision Support Systems. Englewood Cliffs, Prentice-Hall

Stahl I. (1972) Bargaining Theory. Stockholm, Stockholm School of Economics.

Starke K, Rangswamy A (1999) Computer-Mediated Negotiations: Review and Research Opportunities. University Park, eBusiness Research Center: 37

Sycara, DZ (1997). How can an agent learn to negotiate. In: M. W. J. Mueller and N. Jennings (Eds), Intelligent Agents III. Agent Theories, Architectures, and Languages, Springer Verlag

Tesauro G, Bredin JL (2002) Strategic Sequential Bidding in Auctions using Dynamic Programming. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 2, ACM, New York

Thompson L, Nadler J (2002) Negotiating via Information Technology: Theory and Application. Journal of Social Studies 58(1): 109-124

Torsun IS (1995).Foundations of Intelligent Knowledge-Based Systems. Bradford, Academic Press Inc.

Ury W (1993) Getting Past No. Negotiating your Way from Confrontation to Cooperation. New York, Bantam Books

Vahidov R, Fazlollahi B (2004) Pluralistic Multi-Agent Decision Support System: a Framework and an Empirical Test. Information & Management 41(7): 883-898

Von Neumann J, Morgenstern O (1944) Theory of Games and Economic Behavior. Princeton: Princeton University Press

Ströbel M, Weinhardt C (2003) The Montreal Taxonomy for Electronic Negotiations. Group Decision and Negotiation, 12(2): pp. 143-164

Wetlaufer GB (1996) The Limits of Integrative Bargaining. Georgetown Law J. 85: 1-25

Wolfstetter E (1996) Auctions: An Introduction. Journal of Economic Surveys 10(4): 367-420

Wooldridge M (1999) Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. G. Weiss. Cambridge, The MIT Press: 27-78.

Wooldridge M, Jennings NR (1995) Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2): 115-152

Wurman PR, Wellman MP, et al (1998) The Michigan Internet AuctionBot: A configurable auction server for human and software agents. Second International Conference on Autonomous Agents, Minneapolis, MN, pp 301-308.

Wurman PR, Wellman MP, *et al.* (2001) A Parametrization of the Auction Design Space. Games and Economic Behavior 35: 304-338.

Yokoo M, Durfee EH et al (1998) The distributed constraint satisfaction problem: formalization and algorithms, IEEE Transactions on Knowledge and Data Engineering, 10(5):673-685, 1998

Youll J, Morris J, *et al.* (2000). Impulse: Location-based Agent Assistance. Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain.

Young OR (1975) Bargaining. Formal Theories of Negotiations. University of Illinois Press, Urbana, IL

Zeng D, Sycara K (1997) Benefits of Learning in Negotiation. Proceedings of AAAI-97

Zeng D, Sycara K (1998) Bayesian learning in negotiation. Int. J. Human-Computer Studies 48: 125-141

Zlotkin G (1996) Mechanisms for Automated Negotiation in State Oriented Domains Journal of Artificial Intelligence Research 5: 163 - 238