Chapter 10
A Multi-Agent Environment for Negotiation

Koen V. Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov

Abstract In this chapter we introduce the System for Analysis of Multi-Issue Negotiation (SAMIN). SAMIN offers a negotiation environment that supports and facilitates the setup of various negotiation setups. The environment has been designed to analyse negotiation processes between human negotiators, between human and software agents, and between software agents. It offers a range of different agents, different domains, and other options useful to define a negotiation setup. The environment has been used to test and evaluate a range of negotiation strategies in various domains playing against other negotiating agents as well as humans. We discuss some of the results obtained by means of these experiments.

10.1 Introduction

Research on negotiation is done in various research disciplines; business management, economics, psychology, and artificial intelligence. The foundations of negotiation theory are decision analysis, behavioral decision making, game theory, and negotiation analysis. The boost of literature on negotiating agents and strategies of recent years is in line with the continuous advance of ecommerce applications, such as eBay, and Marketplace in which negotiations play a role. In essence it focuses on the development of ever more clever negotiation agents, that are typically tested in one domain, against one or two other negotiation agents, almost never against humans. In our opinion, in order to become acceptable as negotiators on behalf of human stakeholders, negotiation agents will have to prove their worth in various domains, against various negotiation strategies and against human negotiators. In order to gain a better understanding of the negotiation dynamics and the factors that influence the negotiation process it is crucial to not

Koen V. Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov
Man-Machine Interaction group, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands, e-mail: {K.V.Hindriks,C.M.Jonker,D.Tykhonov}@tudelft.nl

R.H. Bordini et al. (eds.), *Multi-Agent Programming*,
DOI 10.1007/978-0-387-89299-3_10,

only mathematically evaluate the efficiency of negotiation outcomes but also to look at the pattern of offer exchanges, what Raiffa [30] calls the negotiation dance. In the remainder we present architecture of a formal toolbox to simulate negotiations and analyze patterns in offer exchanges and present some initial findings in the literature. The System for Analysis of Multi-Issue Negotiation[1] (SAMIN) is developed as a research tool, to improve the quality of negotiating agents, and as a training environment to develop negotiation skills of human negotiators. To that purpose SAMIN offers a range of analytical tools, a tournament tool, a preference elicitation tool, and a number of negotiation domains, negotiation agents, and user interfaces for human negotiators.

10.2 Application Domain

Negotiation is an interpersonal decision-making process necessary whenever we cannot achieve our objectives single-handedly [32]. Pruitt [28] emphasizes the process of negotiation and the fact that the outcome should be a joint decision by the parties involved. Typically each party starts a negotiation by offering the most preferred solution from the individual area of interest. If an offer is not acceptable by the other parties they make counter-offers in order to move each other closer to an agreement. The field of negotiation can be split into different types, e.g. along the following lines: (a) one-to-one versus more than two parties; (b) single-versus multi-issues; (c) closed versus open (d) mediator-based versus mediator-free. The research reported in this chapter concerns one-to-one, multi-issue, closed, mediator-free negotiation. A special case of one-to-many negotiation is considered. In this case, an auction mechanism [10] is approximated by a negotiation setup [16]. For more information on negotiations between more than two parties (e.g., in auctions), the reader is referred to, e.g., [31]. In single-issue negotiation, the negotiation focuses on one aspect only (typically price) of the object under negotiation. Multi-issue negotiation (also called multi-attribute negotiation) is often seen as a more cooperative form of negotiation, since often an outcome exists that brings joint gains for both parties, see [30]. Closed negotiation means that no information regarding preferences is exchanged between the negotiators. The only information exchanged is formed by the bids. More information about (partially) open negotiations can be found, e.g., in [20] and [30]. However, the trust necessary for (partially) open negotiations is not always available. The use of mediators is a well-recognised tool to help the involved parties in their negotiations, see e.g., [19, 30]. The mediator tries to find a deal that is fair to all parties. Reasons for negotiating without a mediator can be the lack of a trusted mediator, the costs of a mediator, and the hope of doing better. The SAMIN system is developed to support research into the analysis of negotiation strategies. The analysis of negoti-

[1] This negotiation environment, user manuals, and a number of implemented negotiation agents can be downloaded from http://mmi.tudelft.nl/negotiation.

ation strategies provides new insights into the development of better negotiation strategies.

Negotiation parties need each other to obtain an outcome which is beneficial to both and is an improvement over the current state of affairs for either party. Both parties need to believe this is the case before they will engage in a negotiation. Although by engaging in a negotiation one party signals to the other party that there is potential for such gain on its side, it may still leave the other party with little more knowledge than that this is so. Research shows that the more one knows about the other party the more effective the exchange of information and offers [30]. Furthermore, humans usually do have some understanding of the domain of negotiation to guide their actions, and, as has been argued, a machine provided with domain knowledge may also benefit from such domain knowledge [6]. It is well-known that many factors influence the performance and outcome of humans in a negotiation, ranging from the general mindset towards negotiation to particular emotions and perception of fairness. As emphasized in socio-psychological and business management literature on negotiation, viewing negotiation as a joint problem-solving task is a more productive mindset than viewing negotiation as a competition in which one party wins and the other looses [7, 30, 32]. Whereas the latter mindset typically induces hard-bargaining tactics and rules out disclosure of relevant information to an opponent, the former leads to joint exploration of possible agreements and induces both parties to team up and search for trade-offs to find a win-win outcome. Different mindsets lead to different negotiation strategies. A similar distinction between hard- and soft-bargaining tactics has also been discussed in the automated negotiation system literature where the distinction has been referred to as either a boulware or a conceder tactics [5]. Emotions and perception of fairness may also determine the outcome of a negotiation. People may have strong feelings about the "rightness" of a proposed agreement. Such feelings may not always be productive to reach a jointly beneficial and efficient agreement. It has been suggested in the literature to take such emotions into account but at the same time to try to control them during negotiation and rationally assess the benefits of any proposals on the table [7, 32]. Apart from the factors mentioned above that influence the dynamics of negotiation, many other psychological biases have been identified in the literature that influence the outcome of a negotiation, including among others partisan perceptions, overconfidence, endowment effects, reactive devaluation [25, 32].

10.2.1 The Added Value of the MAS Paradigm

Negotiation involves conflicting interests, hidden goals, and making educated guesses about the preferences and goals of the other parties involved. A system that supports closed negotiation needs to protect the integrity of the parties or stakeholders that participate in a negotiation and it is natural to provide every stakeholder with an agent of their own. It thus is natural to use the MAS paradigm

to model the interaction between negotiating parties. Parties in a negotiation are autonomous and need to decide on the moves to make during a negotiation. This decision problem is particularly complex in a closed negotiation where negotiating parties do not reveal their preferences to each other. Moreover, other factors such as the complexity of the domain of negotiation may pose additional problems that need to be solved by a negotiating agent.
SAMIN contributes to the MAS paradigm as a research tool that facilitates research into the design of efficient negotiation strategies. The tool more specifically facilitates the evaluation of the performance of a negotiation strategy by means of simulating multiple negotiation sessions and feeding the results of the simulation to the analytical toolbox of SAMIN. We have found that the results of a well-defined negotiation setup may help analysing the strengths and weaknesses of a strategy and may be used to improve a negotiation strategy significantly. It has also been shown that strategies may perform quite differently on different domains. A variety of negotiation domains and agents is available in SAMIN to evaluate a negotiation strategy in different negotiation setups. The open architecture of SAMIN, moreover, facilitates the integration of new negotiation domains and agents.

10.2.2 Design Methods Used

An earlier version of SAMIN, see [2, 17], was designed using the DESIRE method [3]. Redesign was necessary to open the system for agents designed and implemented by others and to ease the definition of new negotiation domains. The redesigned version is implemented in the Java programming language that is supported my the majority of computer platforms.
The current version of SAMIN implements the architecture proposed in [13]. Figure 10.1 illustrates this architecture. The architecture is based on an analysis of the tasks that need to be supported by a generic negotiation environment that is capable of integrating a variety of negotiating agents and is able to simulate negotiations between such agents. The architecture provides a minimal but sufficient framework including all features necessary to simulate a wide range of negotiation scenarios and to enable integration of negotiation agents. The architecture consists of four main layers, a human bidding interface, and a negotiating agent architecture. The four layers include an *interaction layer*, an *ontology layer*, a *graphical user interface layer*, and an *analytical toolbox*.
The *interaction layer* provides functionality to define negotiation protocols and enables communication between agents (see Section 10.4.2 for details). The *ontology layer* provides functionality to define, specify and store a negotiation domain, and the preferences of the negotiating agents (see Section 10.4.3 for details). The architecture can also be used for education purposes and for the training of humans in negotiation. For that purpose, a *graphical user interface layer* is available that facilitates human user(s) to participate in a negotiation setup (see Section

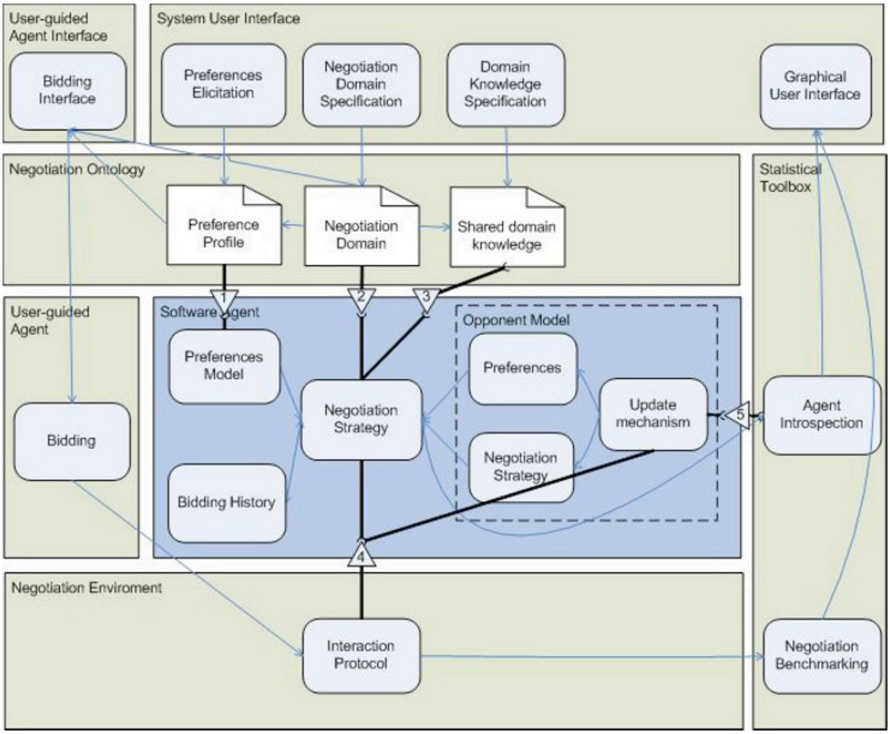

Fig. 10.1 The Open Negotiation System Architecture

10.2.3 for details). The *analytical toolbox* provides functionality to organize tournaments between agents, and to review the performance and benchmark results of agents that conducted a negotiation. It provides a variety of tools to analyze the performance of agents and may also be used to compute quality measures related to e.g. the quality of an opponent model [15].

The architecture that is introduced here identifies the main integration points where adapters are needed to connect a negotiating agent to this architecture. The agent architecture itself defines the common components of a negotiating agent. This architecture may be instantiated with various software agents, as illustrated below.

The integration points or interfaces to connect software agents to the negotiation environment which allows them to interact with other agents available in the environment are numbered 1 through 5 in Figure 10.1. To integrate heterogeneous negotiation agents, such agents have to be aligned with these integration points. Alignment by complete redesign of the agent typically requires significant programming efforts and may also cause backward compatibility problems. To minimize the programming efforts, a better approach is to use a set of adapters or wrappers which are used to wrap the agent code. We have used the adapter design pattern [22] for this purpose. The minimal set of adapters that has to be implemented includes a negotiation domain adapter, a preference profile adapter and an interaction protocol adapter, which each correspond to an essential element of a negotiation. The shared domain knowledge adapter and the agent introspection adapter are optional. The shared domain knowledge adapter provides additional knowledge about the domain to all agents, making this knowledge shared and publicly available. The agent introspection adapter facilitates the introspection of internal components of an agent, such as an opponent model. The latter adapter is mainly available for analysis purposes and research. For more details about the adapters the reader is referred to [13].

10.2.3 User Interaction

The user interaction in SAMIN takes place in the graphical user interface layer and can be divided in two categories of user: researchers and human subjects in experiments. We implemented a graphical user interface that enables a user to define the *negotiation game*. the parameters of the negotiation, the subject or domain of negotiation, and preferences of the agents (which also means that the preferences of a human subject can be predefined).

10.2.3.1 Negotiation Domain and Preference Profile Editor

The *Negotiation Domain and Preference Profile Editor* of SAMIN (see Figure 10.2) is used to create and modify negotiation domains and preference profiles. A *negotiation domain* is a specification of the objectives and issues to be resolved by means of negotiation. An objective may branch into sub-objectives and issues providing a tree-like structure to the domain. The leafs of such a tree representing the domain of negotiation must be the issues that need to be agreed upon in a negotiation. Various types of issues are allowed, including discrete enumerated value sets, integer-valued sets, real-valued sets, as well as a special type of issue used to represent a price associated with the negotiation object. For every issue the user can associate a range of values with a short description and a cost.
A *preference profile* specifies personal preferences regarding possible outcomes of a negotiation. The profile is used to convert any offer in that domain to a value indicating how the user would rate that offer, the so called utility value. The current version of SAMIN supports linear additive utility functions [30]. The profile is also called a *utility space*.
A *weight* that is assigned to every issue indicates the importance of that issue. A human user (see Figure 10.2) can move sliders to change the weights or enter their values by hand, which are automatically normalized by the editor. In the issue editor the user can assign an evaluation to every value of the issue. The

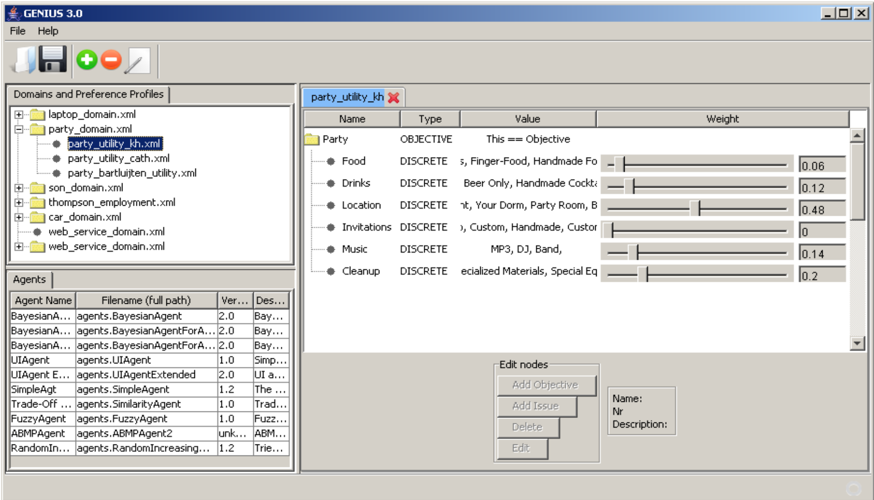

Fig. 10.2 A graphical user interface for preferences elicitation.

evaluation values are positive integers starting with 1. The evaluation values are automatically normalized for each issue to ensure they are in the range [0; 1].

10.2.3.2 Human Negotiator User Interface

A human subject playing in a negotiation game, is provided with a graphical interface for the bidding phase of the game. The bidding interface is implemented with a dummy agent that exchanges the messages between the graphical user interface (GUI) and the environment. Therefore, the GUI for the human negotiator is not hard coded in SAMIN. The GUI can be easily extended without modifications of the SAMIN code. Furthermore, the dummy agent can be replaced with an algorithm that would provide negotiation a support to the human negotiator. It provides, for example, an analysis of the opponent's behaviour or even advise the human negotiator upon the next offer to propose and an action to be taken. Figure 10.3 presents human player GUI that is currently available in SAMIN. This GUI has three main components: a bidding history table (top), a utility history plot (bottom left), and a bidding interface (bottom right). The bidding history shows all bids exchanged between the negotiating parties in a single session. The bids are represented by the values assigned to every issue in the negotiation domain. In addition, the utilities of the bids according to the human player's preference profiles are shown in the table. Note that in a closed negotiation the negotiating parties have no access to the preference profiles of each other and, therefore, utilities can be calculated only on the basis of own preference profile.

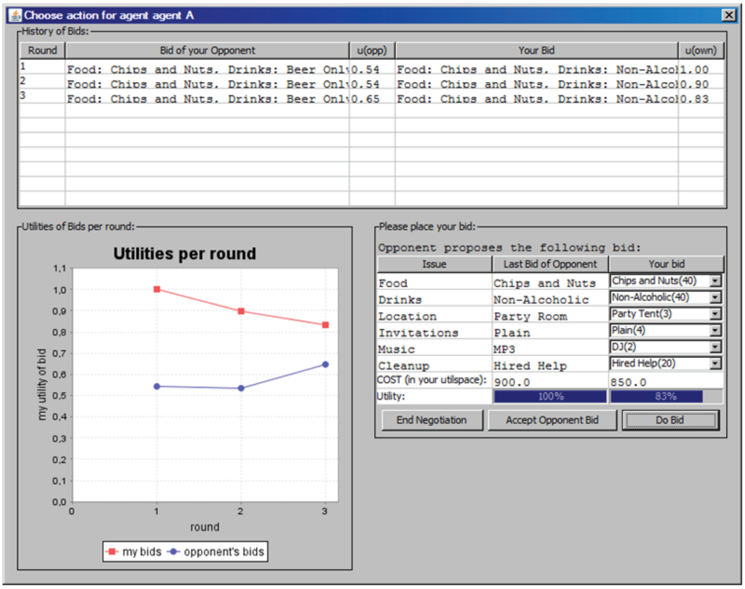

Fig. 10.3 Human negotiator graphical user interface.

The bidding interface has two main components: a table showing the last bid and a possible next bid and a row of buttons representing possible actions for the humans negotiator's. The table has three columns:

- the left column shows the names of the issues in the domain;
- the center column shows the values for the issues as proposed in the last bid of the opponent;
- the right column shows the current selected values for the issues. A user can edit the current bid by clicking on the fields, which will open the drop-down boxes in the fields.

The last two rows of the table show the cost and utility of the last opponent's bid and your current bid. The cost field will turn red if the bid exceeds the maximum cost. The utility is shown as a percentage and also as a bar of matching size. These values are computed according to the user's utility space because a user has no access to the opponent's utility space. The lower three buttons allow a user to submit the next bid as it is set in the right column, or to accept the opponent's last bid.

10.3 Agents

In this section we present an agent architecture in SAMIN and explain the state-of-the-art negotiation agents that are available in SAMIN.

10.3.1 Agent Architecture

The *software agent* component highlighted with the darker area in Figure 10.1 is a generic component that can be instantiated by a variety heterogeneous software agents. The components that are specified as part of a software agent in Figure 10.1 are the parts of the *conceptual design* of such agents but do not need to be actually present or identifiable as such in any particular software agent. These components are not introduced here to specify a requirements that need to be satisfied when developing an agent (although it could be used as such [1, 18, 21]). Here these components are introduced to identify integration points of agents with the system architecture. Five of such integration points, also referred to as *adapters*, were identified above.
In the reminder of this section we discuss every component of the proposed agent architecture.

Preference Model

The component models the agent's preferences with respect to the set of possible negotiation outcomes. The model can be based on various structures: utility functions, rankings, etc. This component can require additional processing depending on the complexity of the agent's preferences and the types of inquiries that can be made by other components, see e.g. [19]. Typically, the preferences model must be able to evaluate an outcome on a given scale, compare two or more outcomes, give a single or a set of outcomes that satisfy some constraints on the negotiation domain and preferences.

Negotiation Strategy

This is the core component of any negotiation agent. It makes decisions about acceptance of the opponent's offer, ending the negotiation, and sending a counter-offer. To propose a counter-offer the negotiation strategy can use various tactics [5]. Depending on the negotiation tactics used in the negotiation strategy the component can use information about the model of the agent's own preferences, the opponent's preferences and strategy (as far as known to or guessed by the agent), and, the previous offers made during the current, or even previous negotiation sessions.

Negotiation History

The negotiation history component keeps track of the bids exchanged between the agents in a negotiation. It can also have a history about earlier negotiations, the outcomes, identities of the opponents, and even opponent models. It can be used by the negotiation strategy component as an additional information source to improve its negotiation performance. For example, in repetitive negotiations with the same opponents this information can be used as a priori knowledge about the opponent to shorten the learning time.

Opponent Model

In the negotiation games we consider here, the preferences of negotiation parties are private [30]. Efficiency of a negotiation strategy can be significantly improved with information about the preferences of the opponent [33]. In the literature a number of learning techniques have been proposed to learn the opponent's preferences model from the offers exchanged in a single-shot negotiation, see e.g., [34, 13]. In [33] it was show that a successful negotiation strategy should make use of an opponent model.

Our generic component consists of three main subcomponents: *preferences*, *negotiation strategy*, and *update mechanism*.

The component *Preferences* contains specifications of the preferences of the current and previous negotiation opponents. As the opponent's preferences are typically private, the preference information has a certain degree of uncertainty. Depending on the agent developed on the base of the generic components information about the certainty of the preferences can be maintained or not.

The aim of the model of *opponent's strategy* is to predict negotiation moves that will be made by the opponent. It is important to know for an agent what the next move of the opponent would be. This knowledge can be used in the negotiation strategy to increase the efficiency of the agent's own offers and increase the chance of acceptance of its offer by the opponent.

Models of the opponent's preferences and strategy are typically learned by the agent from the evidence, such as negotiation agreements achieved in the previous negotiations [33], and offers sent by the opponent in multiple sessions of single-shot negotiations [13, 34, 18]. The learning techniques used in the agent can depend on the types of the models chosen to represent the opponent's preferences and strategy.

10.3.2 State of the Art Negotiating Agents

Interfaces and adapters have been developed to make it easy to integrate agents developed by others into SAMIN, see [13]. A number of the state-of-the-art agents have found a place in SAMIN: ABMP [17], Bayesian agent [14], Bayesian Tit-for-Tat [12], FBM [29], Trade-off agent [6], QO agent [24], Random Walker [11]. As they were developed by different teams, their design, architecture, and implementation varies.

Random Walker

The Random Walker strategy introduced in [11], also known as Zero Intelligence (ZI) strategy [8], randomly jumps through the negotiation space. It does not use own preferences or a model of opponent's preferences to make an offer. Random Walker accepts the opponent's offer if it has higher utility than the agent's own last offer. The Random Walker strategy can be run with a break-off point to avoid making offers below that utility and, thus, introduces some limited rationality in its behaviour.

It is difficult for the Random Walker strategy to achieve a better agreement than its break-off point as there is only a very low probability that it will be able to find bids close to Pareto frontier. Any efficient negotiation strategy that is capable of learning an opponent model and is able to use it efficiently would be expected to outperform the Random Walker strategy. For this reason, the Random Walker strategy may be used as a "baseline" strategy. In addition, as the Random Walker strategy does not derive its moves from its preference profile but only uses an acceptance strategy to avoid outcomes with a utility below its break-off point, it also provides a good test case to evaluate of robustness of a negotiation strategy.

ABMP Agent

The ABMP strategy is a concession-oriented negotiation strategy, see [17]. It selects counter-offers without taking domain or opponent knowledge into account. The ABMP strategy decides on a negotiation move based on considerations derived from the agent's own utility space only. It calculates a utility of a next offer, called *target utility*, based on the current utility gap between the last opponent's offer and the last own offer. To determine the next offer the target utility is propagated to the individual issues taking into account the weights of the issues in the agent's preferences profile. The ABMP strategy can be fine-tuned with a number of parameters, such as the negotiation speed, concession factor, configuration tolerance and others.

The original ABMP strategy was not capable of learning. A heuristic for adapting the ABMP strategy to the opponent's issue priorities was introduced in [18]. The results showed improvement of the negotiation outcome compared to the original version of the ABMP strategy.

The ABMP strategy was implemented in an ad hoc environment using the DESIRE method [3]. The environment facilitated negotiation about a Second-hand car domain [17] that was hard-coded in the implementation. Later, when the second Java-based version of the SAMIN was available the ABMP strategy was re-implemented in SAMIN. The results of the DESIRE-based ABMP implementation were reproduced in SAMIN.

Trade-off Agent

The effectiveness of using knowledge about the negotiation domain has been demonstrated in the Trade-off strategy introduced in [6]. In particular, this paper shows that domain knowledge (coded as so-called similarity functions) can be used to select bids that are close to the opponent's bids, thus increasing the likelihood of acceptance of a proposed bid by that opponent. In this approach, the knowledge represented by similarity functions is assumed to be public.

In [6], the Trade-off strategy is combined with several so called *meta strategies* that control the concession behaviour of the agent. The most interesting meta strategy, the *smart* strategy, consists of deploying a Trade-off mechanism until the agent observes a deadlock in the average closeness of own offers compared to that of the opponent as measured by the similarity function. In a case of the deadlock, the value of the previous offer is reduced by a predetermined amount (0.05), thereby lowering the input value of the Trade-off mechanism.

The Trade-off strategy was originally evaluated on the Service-Oriented Negotiation (SON) domain. The SON domain has four quantitative continuous issues, the price, quality, time, and penalty. Both, buyer and seller use linear functions to evaluate individual issues and combine them in a linear additive utility function using a vector of weights. It is assumed that the buyer and the seller have opposite preferences for every issue, that is, if buyer wants to maximize the quality then the seller wants to minimize it. Therefore, in this domain the differences in the weights are the key elements to consider for joint improvements of the offers.

The Trade-off strategy combined with the smart meta strategy showed good performance on the SON in the experimental setup of [6]. It was demonstrated that the Trade-off strategy is capable of producing very efficient offers resulting in agreements that are very close to the Pareto efficient frontier. Interestingly, the best performance the Trade-off strategy showed in negotiation against itself, while in negotiations against agents that used other meta strategies the utility of agreement was somewhat lower. This phenomenon will be discussed in details in Section 10.6.

Unfortunately, no implementation of the Trade-Off strategy was available. The strategy was implemented in the SAMIN from scratch. The results reported in [6] were reproduced for the Service-Oriented Negotiation domain.

Bayesian Agent

One way to approach the problem of incomplete information in closed negotiation is to learn an opponent's preferences given the negotiation moves that an opponent makes during the negotiation. A learning technique based on Bayesian learning algorithm was proposed in [14]. The opponent model in [14] is based on learning probability over a set of hypothesis about evaluation functions and weights of the issues. The probability distribution is defined over the set of hypothesis that represent agent's belief about opponent's preferences. Structural as-

sumptions about the evaluation functions and weights are made to decrease the number of parameters to be learned and simplify the learning task.
The set of hypotheses about the evaluation function is defined using three types of shapes of the functions: (a) downhill shape: minimal issue values are preferred over other issue values, and the evaluation of issue values decreases linearly when the value of the issue increases; (b) uphill shape: maximal issue values are preferred over other issue values, and the evaluation of issue values increases linearly when the value of the issue increases; (c) triangular shape: a specific issue value somewhere in the issue range is valued most and evaluations associated with issues to the left ("smaller") and right ("bigger") of this issue value linearly decrease (think, e.g., of an amount of goods).
During a negotiation every time when a new bid is received from the opponent the probability of each hypothesis is updated using Bayes' rule. This requires a conditional probability that represents the probability that the bid might have been proposed given a hypothesis. Therefore the utility of bid is calculated according to this hypothesis and compared with the predicted utility according to the rationality assumption. To estimate the predicted utility value an assumption about the opponent concession tactics is used based on a linear function.
Authors propose two versions of the learning algorithm. In the first version of the algorithm each hypotheses represents a complete utility space as a combination of weights ranking and shapes of the issue evaluation functions. The size of the hypothesis space growth exponentially with respect to the number of issue and thus is intractable for negotiation domains with high number of issues.
The second version of the algorithm is a scalable variant for the first one. This version of the agent tries to learn probability distribution over the individual hypothesis about the value of the weight and shape of the issue evaluation function independently of other issues. The computational tractability of the learning is achieved by approximating the conditional distributions of the hypotheses using the expected values of the dependent hypotheses.

QO Agent

In [24] the authors propose a negotiation agent, called QO agent, that is based on qualitative decision making. The QO agent is designed for automated negotiations with multiple issues. The internal structure of the QO agent is similar to the agent architecture proposed in this article. The underlying assumption in the QO agent is that the opponent uses one of three preference profiles. The preference profiles of the opponent are represented in same way as QO agent's own preference profile. A probability is associated with each of the possible opponent profiles. An update mechanism interprets the observed offers from the opponent and updates the probability distribution according to the opponent strategy model. The opponent profiles have the same structure as the own preferences profile and the same preference profile adapter is used to load them from files.

The original implementation of the QO agent uses Java programming language. The interaction protocol, however, is more complex then the alternating offers protocol currently used by the SAMIN. The QO agent environment implements a rather complex interaction protocol that extends the alternating offers protocol. It does not have a clear turn taking flow and allows agents to exchange pre-defined textual messages between the agents, such as threats of breaking negotiation if the last offer is not accepted. It was decided to simplify it in the interaction protocol adapter. Only those functions of the agent were used that represent the core functionality: interpret the opponent's offer, generate next action of the agent, generate a counter-offer.

Fuzzy-based Model Agent

The other agent integrated into the negotiation system is the Fuzzy-based model (FBM) agent introduced in [29]. The Fuzzy-based agent is designed for negotiation where agents can exchange fuzzy proposals. The original FBM agent is designed for negotiations where agents can exchange fuzzy proposals. The original implementation of the FBM agent works only for one-issue negotiations but can be extended for multi-issue negotiations. As a result, the negotiation domain is defined using one issue that takes real values from a give interval. The agent adopts time dependent negotiation tactics from [5] and, thus, always makes concession towards opponent. The offers are defined using two values: the peak value and the stretch of the offer.

The FBM agent is implemented in an experimental setup using Java programming language. The experimental setup uses the alternating offers protocol [27]. The preference profile is hard-coded in the agent and based on a linear function. The experimental setup consists two agents that have opposed preferences over the issues.

Bayesian Tit-for-Tat Agent

In [12] a negotiation strategy is proposed that uses a model of the opponent's preferences not only to increase the efficiency of the negotiated agreement but also to avoid exploitation by the other party in a sophisticated Tit-for-Tat manner.Authors in [12] try to show that two important goals in any negotiation can be realized when a reasonable estimate of the preferences of an opponent is available.

For that purpose they combine the Bayesian learning technique as proposed in [14] with a Tit-for-Tat tactic, see e.g., [5], and the classification of negotiation moves as described in, e.g., [11]. As is typical for Tit-for-Tat, it avoids exploitation by a form of mirroring of the bids of the opponent. Bayesian learning is used to learn the opponent's preferences. The opponent profile together with the classi-

fication scheme is used to develop a sophisticated Tit-for-Tat Bayesian negotiation strategy.
Bidding of the proposed strategy can be understood by the opponent as signalling whether a move is appreciated or not (which is not as easy as it seems). Tit-for-Tat Bayesian negotiation strategy does not punish the opponent for making a move that can be understood as an honest mistake. The strategy is based on a rationality assumption, i.e., that an opponent would tend to accept more preferred offers over less preferred. In line with this assumption the strategy searches for Pareto efficient offers, i.e., offers that cannot be improved for both parties simultaneously. Pareto efficient offers increase the chances that an opponent accepts an offer, while protecting the agent's own preferences as best as possible. Finding such offers requires that the Pareto efficient frontier can be approximated which is only feasible if a reasonable model of the opponent's preferences is available.

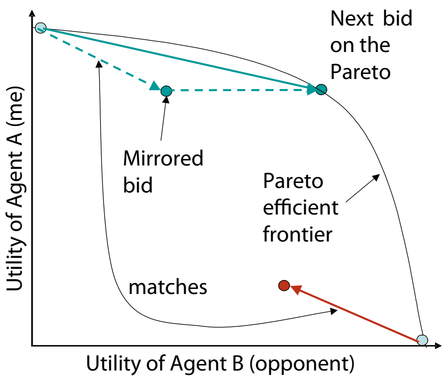

Fig. 10.4 Bayesian Tit-for-Tat Strategy

The basic idea of Tit-for-Tat in multi-issue negotiation is to respond to an opponent move with a symmetrical one, as depicted in Figure 10.4. Typically, a rational negotiation strategy would try to make concession moves at some points during the negotiation. The most reasonable response to a concession move would be a concession move of approximately the same concession size. This is called "mirroring" the move of the opponent.
Mirroring simply in this manner would imply that an unfortunate move (an offer that decreases utility for both parties compared to the agent's previous offer) of the opponent would be answered with an unfortunate step. However, it is not rational to consciously make unfortunate steps. Therefore, authors conclude that the pure tactic by mirroring the opponent moves is too simplistic. Instead they use an approximation of the Pareto frontier computed using the learned opponent model and the agent's own preference profile to add an additional step.
The Bayesian Tit-for-Tat strategy is constructed on the basis of the assumption that by maximizing the opponent's utility in every offer, the chance of acceptance increases as well. Therefore, if after mirroring the opponent's move the efficiency

of the agent's own next move can be increased by selecting an equivalent offer (with respect to the agent's preference profile) on the Pareto frontier the strategy will choose to make that offer. Important is that this approach makes the Bayesian Tit-for-Tat negotiation strategy less dependent on the efficiency of the opponent's strategy. The opponent might intend to make a concession but in fact make an unfortunate move. By selecting a bid on the approximated Pareto frontier, while mirroring the concession intent of the opponent, the strategy is able to maintain a high efficiency of the outcome, no matter what mistakes the opponent makes.

10.4 Multi-Agent System

The organisation of SAMIN as a multi-agent system and as research environment is introduced in [13].

10.4.1 Organisation

Negotiation, in fact, can take place in a distributed environment. To support distributed negotiation a Web-based interface to the system will be introduced in the next version. This will enable negotiations between humans that are physically distributed. In addition, the Web interface will allow researchers to upload their code from different locations and participate in a tournament.
To setup a negotiation a negotiation template is created. Negotiation template specifies all details of the negotiation: number of agents (currently only bilateral negotiations are supported), names of the agent's classes that implement negotiation strategies, negotiation domain and preference profiles of the parties. This setup is static through single negotiation session.
The structure of the multi-agent system and organisation of the negotiating agents in SAMIN is determined by the negotiation protocol that is used. The interaction of agents is also fully controlled by the environment and negotiation protocol used. All agents are required to comply with the protocol, which is enforced by the environment.

10.4.2 Interaction

The interaction layer manages the rules of encounter or protocol that regulate the agent interaction in a negotiation. Any agent that wants to participate in such a negotiation protocol must accept and agree to conform to these rules. An interaction protocol specifies which negotiation moves and what information exchange between agents is allowed during a negotiation.

The current version of SAMIN focuses on bilateral negotiation. A centralized interaction engine is used, which facilitates the control over the negotiation flow and the enforcement of rules on the negotiation process. The interaction engine also feeds information to the advanced logging capabilities of SAMIN. Logs are used by the analytical toolbox to assess the performance of negotiation strategies and algorithms, see [11, 13]. Interaction protocols are implemented in the negotiation environment as a separate component to allow the use of a variety of protocols. Implementation of a new interaction protocol in the negotiation environment is a relatively easy task and has no or minimal effect on the agent code.

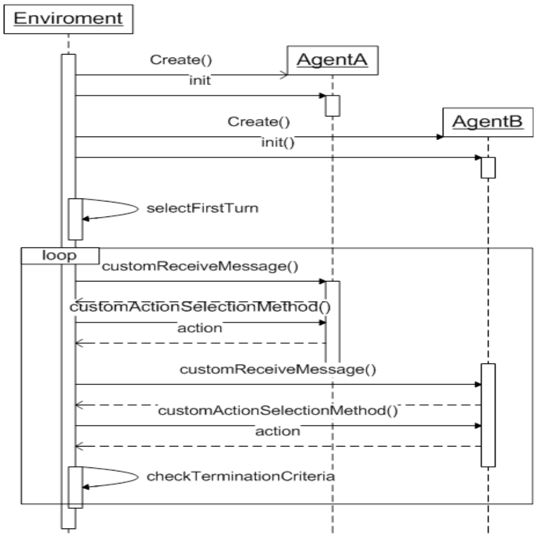

Fig. 10.5 A sequence diagram of the interaction protocol

An example of one of the best known negotiation protocols, the alternating offer protocol [27], is illustrated in Figure 10.5. The alternating offers protocol in a bilateral setting dictates a simple turntaking scheme where each agent is allowed to make a single negotiation move when it is its turn. Apart from turntaking a protocol may also dictate whether exchange of complete package deals is required or that alternatively the exchange of partial bids is allowed. In addition a protocol may manage deadlines, or timeouts that are fixed by the environment.
The interaction protocol is initialized with the information provided by the user. There is no need for a yellow pages mechanism as the agents are made aware about the identity of each other and thus are able to keep track of previous negotiations with the same partner if multiple negotiation sessions are played.
In [16] an alternative protocol involving multiple agents is introduced that is also available in SAMIN. The motivation for introducing this protocol is that it can be used to simulate an auction mechanism. [16] shows that a particular auction

mechanism, called the Qualitative Vickrey Auction (QVA) [10], can be simulated with the protocol.[2]

The QVA mechanism can be thought of as consisting of two rounds. In the first round, the buyer publicly announces her preferences, potential service providers (sellers) submit offers in response, and a winner is selected by the buyer. The winner is the seller who has submitted the best offer from the point of view of the buyer. After establishing the winner, in a second round, the buyer determines the second-best offer (from its perspective again) it received from another seller, announces this publicly, and then the winner is allowed to select any agreement that has at least the same utility to the buyer as the second-best offer (which can be determined by the winner since the preferences of the buyer are publicly announced). It is assumed that the bids proposed in the first round are all monitored by a trusted third party.

The negotiation protocol of [16] provides an alternative to the QVA mechanism. An advantage of using a negotiation setup instead of the QVA is that in that case the buyer does not have to publicly announce its preferences. The negotiation protocol is structured in two rounds to match the structure of the mechanism. In the first round negotiation sessions are performed between the buyer and every potential seller using the Alternating offers protocol (see Figure 10.5). Moreover, the negotiation sessions are assumed to be independent. At the end of the first round, a winner (one of the sellers) is determined. Before starting the second round, the agreement between the seller and buyer that is second-best from the perspective of the buyer is revealed to all sellers, in particular to the winner. In the second round an agreement between the winner and the buyer is established. In section 10.6 we present some experimental results received for the proposed negotiation mechanism.

10.4.3 MAS Environment

The MAS environment in SAMIN is a negotiation environment that controls some aspects of the agent's behaviour, such as the setup and initialization of a negotiation session(s), compliance of the agents with a selected negotiation protocol, etc. The layers with corresponding components of the negotiation environment are shown in Figure 10.1 and have a lighter background. First of all, the negotiation environment provides a negotiation ontology to the agents. The ontology specifies concepts, such as a negotiation domain, a preference profile, and shared knowledge.

A *negotiation domain* is a specification of the objectives and issues to be resolved by means of negotiation. It specifies the structure and content of bids or offers exchanged, and of any final outcome or agreement. An outcome determines a specific value for each issue, or, alternatively, only for a subset of the issues. Ob-

[2] The QVA is a generalization of the well-known Vickrey auction to a multi-issue setting where payments are not essential. In QVA a buyer has complex preferences over a set of issues.

jectives allow to define a tree-like structure with either other objectives again or issues as children, in line with [30]. Various types of issues are allowed, including discrete enumerated value sets, integer-valued sets, real-valued sets, as well as a special type of issue called *price* issue. Additionally, a specification of a negotiation domain may introduce constraints on acceptable outcomes. For example, costs associated with a particular outcome may not exceed the available budget of the agent.

A *preference profile* specifies the preferences regarding possible outcomes of an agent. It can be thought of as a function mapping outcomes of a negotiation domain onto the level of satisfaction an agent associates with that outcome. The structure of a preference profile for obvious reasons resembles that of a domain specification. The tree-like structure allows to specify relative priorities of parts of the tree. This allows, for example, to ensure that all issues relating to travelling combined are weighted equally as all issues relating to the actual stay at a particular location.

In a *closed* negotiation an agent is not informed about the preferences of its negotiating partner. In that case an agent can at best use a reconstruction (using e.g. machine learning techniques) of these preferences to decide on the negotiation move it should do next. It is typical, however, that with a domain comes certain public knowledge that is shared and can be used to obtain a better negotiation outcome. For example, common preferences such as preferring early delivery over later (though not always the case) may be common knowledge in a given domain. Such knowledge allows agents to compute the preferences of their negotiation partner e.g. using the time interval between two dates. This type of knowledge, labelled *shared domain knowledge*, is modelled explicitly as a separate component that can be accessed by all negotiating agents.

The *analytical toolbox* layer of the negotiation environment a set of statistical analysis methods to perform an outcome analysis on negotiation sessions as introduced and discussed in e.g., [11, 30]. Furthermore, the toolbox contains methods for the analysis of dynamic properties of negotiation sessions as discussed in e.g., [11]. The methods for both outcome and dynamics analysis were used to produce a number of performance benchmarks for negotiation behaviour and for the agent components [13]. The analytical toolbox uses the optimal solutions [30], such as the Pareto efficient frontier, Nash product and Kalai-Smorodinsky solution for the negotiation outcome benchmarking. The benchmarks in the negotiation system can be used to analyze the performance of opponent modelling techniques, the efficiency of negotiation strategies, and the negotiation behaviour of the agent. The result of the analysis can help researchers to improve their agents. The output of the analytical toolbox is presented graphically (see e.g., Figures 10.6 and 10.8).

10.5 Execution Platform

The system is implemented as a stand-alone application running on a single computer. The negotiation settings, such as role and types of the agents, negotiation domain, and preference profiles are predefined by an script. A tournament is a typical experimental setup for negotiating agents [11]. Therefore, the system has a utility to generate scripts for a tournament setup and can automatically run a sequence of negotiation.
SAMIN is currently focused on the closed negotiations, where negotiating parties have no access to the preference profiles of each other. In addition, agent's own preference profile is supposed to be static during negotiation and cannot be changed during the negotiation. Few security precautions were implemented in SAMIN to meet these requirements and avoid situations where agents would improve their performance by means of software hacks. This is especially important when SAMIN is used as a testbed for negotiating agents or in an educational setup. Negotiating agents in SAMIN as any imperfect software product can fail. All errors and exception raised by the agent's code are properly logged by the SAMIN to allow the agent's developer to improve it. SAMIN uses multi-threading mechanism to assure responsiveness of the SAMIN's GUI during negotiation sessions. Agents running into a deadlock can be stopped by the user by means of the GUI without fatal consequences for the negotiation environment.
The algorithms used in the negotiation strategies can have high computational complexity [19] and, thus, require significant computational power from the execution platform and essential time slot to perform necessary computations to process opponent's offer or select the next action. Negotiation typically, take place under time constraints [5]. Therefore, a timeout mechanism is implemented in SAMIN.
The agents are notified by the negotiation environment about the time left until the deadline using the real-time clock. The timeout mechanism can be switched off by the user when SAMIN is used as a research tool.

10.6 Results

The main advantage of the proposed MAS architecture is to allow for integration of heterogeneous agents and to facilitate comparison of their negotiation. SAMIN can be used as a testbed to perform experiments with various negotiation domains, preference profiles and negotiating agents. Thus, it contributes to automated negotiating agents research by providing a tool that is able to show new insights about such agents. Here we shortly present the most interesting results received with SAMIN for negotiating agents that have been implemented and/or integrated in it.

10.6.1 Experimental Setup

A tournament is a typical experimental setup for evaluation of negotiating agents It enables analysis if the behaviour and effectiveness of an agent compared to that of others. Multiple negotiation domains and preferences profiles can be selected for a tournament. To test sensitivity of a strategy to its internal parameter the value of the parameter can be varied in a tournament. Every session can be repeated a number of times to build a representative sample of negotiation results for a statistical analysis in case of non-deterministic negotiation strategies.

A number of negotiation factors influencing negotiation behaviour have been reported in [11]. We reuse these factors in our method.

Size of the negotiation domain. Complexity of the negotiation domain and preference profiles is determined by the size of the negotiation domain. Size of the domain can influence learning performance of the negotiation strategy and, thus, the outcome reached by the strategy [14]. The size of the domain is exponential with respect to the number of issues. Therefore, to be able to test scalability of a negotiation strategy the experimental setup should have a set of domains ranging from low number of issues to higher number of issues.

Predictability of the preferences. Negotiation strategies can try to exploit the internal structure of the preferences in order to improve one's own efficiency. I.e., the Trade-off strategy assumes that distance measures can be defined using domain knowledge for the preferences of the opponent. These measures combined with the opponent's offers allow the Trade-off strategy to predict opponent preferences and as a result improve efficiency of the bidding. In [11], however, it has been shown that in case of a mismatch of the domain knowledge and the actual structure of the opponent's preferences the performance of a strategy can drastically drop. Therefore, we introduce the notion of the predictability of the preferences into our method.

Issues are called predictable when even though the actual evaluation function for the issue is unknown, it is possible to guess some of its global properties. For example, a price issue typically is rather predictable, where more is better for the seller, and less is better for the buyer, and the normal ordering of the real numbers is maintained; an issue concerning colour, however, is typically less predictable.

Opposition of the preferences. The results of analyzing negotiation dynamics presented in [11] revealed that some negotiation strategies are sensitive to preference profiles with compatible issues. Issues are compatible if the issue preferences of both negotiating parties are such that they both prefer the same alternatives for the given issue. Negotiation strategies may more or less depend on whether preferences of the negotiating parties are opposed or not on every issue. That is, using some strategies it is harder or even impossible to exploit such common ground and agree on the most preferred option by both parties for compatible issues (humans are reported to have difficulty with this as well; cf. [32]). A selection of preference profiles should therefore take into account that both preference profiles with and without compatible issues are included.

To measure the opposition between two preference profile we use ranking distance measure proposed [16]. The measure is based on the conflict indicator proposed in [9]. The conflict indicator function yields 1 when the ranking relation of two arbitrary outcomes based on the utility space of one agent is not the same as the ranking relation based on the utility space of the opponent; if the rankings based on both utility functions match the conflict indicator takes the value of 0. The conflict indicator is calculated for all permutations in the negotiation domain and normalized over the domain. The higher the value of the ranking distance the stronger opposition between the preference profiles.
Another measure for the opposition of preferences proposed in [15] uses Pearson's correlation coefficient for that purpose. This coefficient represents the degree of linear relationship between two variables. The Pearson's correlation coefficient takes a real value from the interval $[-1; 1]$. A value of $+1$ means that there is a perfect positive linear relationship between variables, whereas a value of -1 means that there is a perfect negative linear relationship between variables. A value of 0 means that there is no linear relationship between the two variables.
The following negotiation domains and preference profiles are available in SAMIN (see Table 10.1 for summary):

- The *Second hand car selling* domain, taken from [17], includes 5 issues. Only the buyer's preferences and the price issue are predictable, in the sense that an agent can reliably predict the other agent's preferences associated with an issue.
- The *Party* domain is created for negotiation experiments with humans. It is a rather small domain with 5 discrete issues with 5 possible values each. All of the issues are unpredictable. In this domain, the preference profiles used are not as opposed to each other as in the other domains.
- The *Employment contract negotiation* domain, taken from [26] with 5 discrete issues. All issues have predictable values. The preference profiles are strongly opposed, i.e. both negotiators dislike outcomes that the other prefers most.
- The *Service-Oriented Negotiation* domain, taken from [6], includes 4 issues. All issues are predictable, i.e. based on available "domain knowledge" preferences can be reliably predicated.
- The *AMPO vs City* domain, taken from [30], includes 10 issues, of which 8 are predictable. Information about the opponent's issue priorities, i.e. the weights agents associate with issues. This is a large domain with more than $7,000,000$ possible outcomes.

Domain	Utility spaces		Weights		Domain size	Number of predictable
	Ranking	Pearson	Ranking	Pearson		
AMPO vs. City	0.662	-0.482	0.422	-0.139	7,128,000	3 (10)
Party	0.540	-0.126	0.467	-0.276	3,125	0 (5)
SON	0.669	-0.453	0.833	-0.751	810,000	4 (4)
2nd hand car	0.635	-0.387	0.600	-0.147	18,750	1 (5)
Employment contract	0.698	-0.584	0.600	-0.241	3,125	5 (5)

Table 10.1 Summary of the negotiation domains and preference profiles

10.6.2 Experimental Results

Here we present the most interesting results we received for the state-of-the-art agents described in Section 10.3.2.

Trade-off and ABMP Agents

Figure 10.6 shows typical runs in the AMPO vs City domain. Figure 10.6a shows a run of Trade-Off, representing the City, versus Random Walker (with break-off set to 0.6), playing AMPO. The Random Walker strategy is insensitive with respect to its own preferences. This fact, combined with the lack of information of relative importance of issues (weights) causes the unfortunate moves (an offer that decreases utility for both parties compared to the agent's previous offer, see [11]) produced by the Trade-off strategy.
Figure 10.6b shows Trade-off (as City) vs ABMP (as AMPO) in which ABMP is rather insensitive to the behaviour of the opponent, and Trade-off is sensitive. In this domain Trade-off really exploits the available domain knowledge. Figure 10.6c shows Random Walker (City) vs ABMP (AMPO). ABMP always concedes on all issues, determining the size of the concession on the difference between the utilities of its own bid and that of its opponent. It does not use previous opponent bids to get insight into the opponent's preferences and, as a result, does not adapt much to the strategy of the opponent.

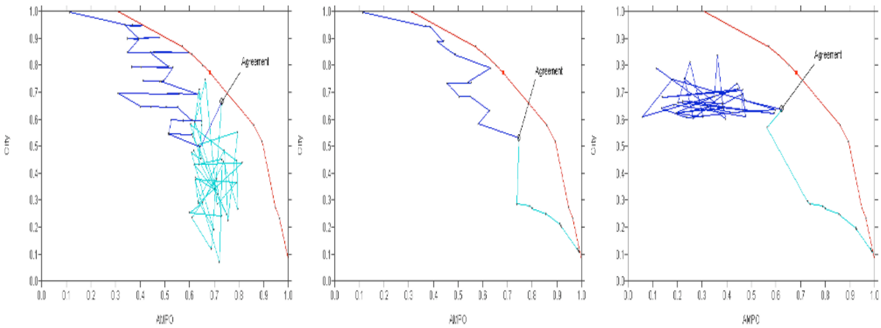

Fig. 10.6 Dynamics of negotiation process for: a) Trade-off (City) vs Random Walker strategy (AMPO), b) Trade-off (City) vs ABMP strategy (AMPO), c) Random Walker (City) vs ABMP strategy (AMPO).

This analysis shows a direct link between the correctness and/or completeness of the domain knowledge and opponent preferences sensitivity. The Trade-off strategy is very sensitive to opponent preferences given complete information. In that case, the similarity functions exactly match the opponent's preferences and the weights exactly represent the issue importance factors of the opponent.

The SON domain does not have information about weights of the similarity functions and thus opponent preferences sensitivity of the Trade-off strategy decreases but it is still more sensitive to the opponent preferences than ABMP. Similarity functions for the Second hand car domain were defined in such a way that they often do not match the preferences of the negotiation opponents. In addition, the weights of the similarity function do not match the opponent's importance factors of the negotiation issues. This leads to under performance of the Trade-off strategy while ABMP shows more robust negotiation behavior. The experiments show that if less domain knowledge is available, Trade-off makes more unfortunate steps.
In general, when issues are predictable, the chance of making an unfortunate step becomes small. This aspect becomes clear in the car domain, where the seller's preferences are rather predictable, but the buyer's preferences vary a lot.
We conclude that it is impossible to avoid unfortunate steps without sufficient domain knowledge or opponent knowledge. Indeed, the similarity criteria functions used in the Trade-off Strategy provide general information about the negotiation problem, but do not take into account the specific attributes of the negotiating parties. In any particular case, a negotiator may deviate from the generalized domain model in various ways. Approaches as reported in [4, 23, 32] apply techniques to learn more about the opponent.

Bayesian Agent

In small domains such as the SON domain, the Bayesian agent is very efficient in learning issue weights and evaluation functions of the issues that is indicated by the fact that the negotiation trace almost coincides with the Pareto frontier, see [14] for the details. Here we demonstrate the effectiveness of the scalable version of the Bayesian Agent on larger domains. The results on the AMPO vs City domain presented in Figure 10.7 show, as is only to be expected, that it becomes harder to stay close to the Pareto efficient frontier. The performance of the Bayesian learning agents is now similar to that of the agent based on the Trade-off strategy and both stay close to the Pareto frontier. The ABMP strategy shows similar behaviour as on the other negotiation domains, and is outperformed by the other strategies. The results thus are still very good. Also, note that the agreement reached by the Bayesian agents has a higher utility than that reached by the other strategies and that both the Bayesian agent without domain knowledge as well as the Trade-off agent make quite big unfortunate steps.

QO Agent

Figure 10.8 presents the results of the negotiation experiment. A small and simple negotiation problem, called "Party" [14], is used to analyze the performance of the QO gent within our negotiation framework. This domain has been created

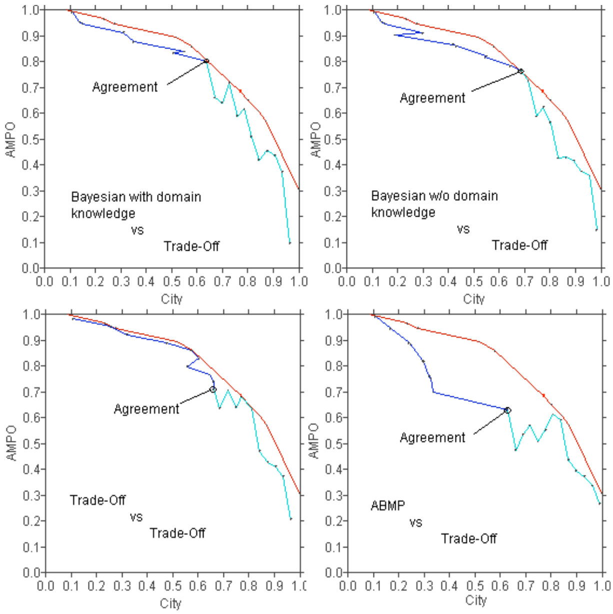

Fig. 10.7 Negotiation dynamics for the Bayesian agent on the AMPO vs. City domain

for negotiation experiments with humans, which also explains its rather limited size. The charts show the space of all possible negotiation outcomes. The axis represent the utilities of the outcomes with respect to the utility functions of the negotiating agents. The charts show the negotiation paths of the agents marked by arrows with the names of the agents.

The Bayesian agent starts with an offer that has maximum utility. It tries to learn the opponent preferences from the offers it receives and uses this model when it makes a concession towards the opponent. As a result, it stays close to the Pareto Efficient frontier. The QO agent in this domain has more difficulty to propose efficient offers. This is a result of limitation of the opponent model of the agent. The QO agent accepts an offer of the Bayesian agent as soon as such an offer has a utility level for the QO agent that is higher then utility of the QO agent's own offer.

Fuzzy-based Model Agent

The other agent integrated into SAMIN is the FBM agent introduced in [29]. The FBM agent was tested in a setup where it has to negotiate against the Bayesian agent about a single issue defined on real values ranging from 10 to 30. The original FBM agent is designed for negotiations where agents can exchange fuzzy proposals. The implementation of the FBM agent we used is able to negotiate about one-issue negotiations but can be extended for multi-issue negotiations. The agent

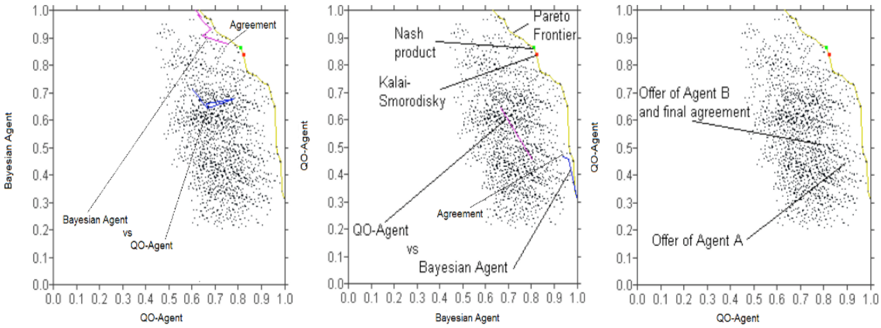

Fig. 10.8 Negotiation dynamics for the QO agent on the Party domain

adopts time dependent negotiation tactics from [5] and, thus, always makes concessions towards opponent. The offers are defined using two values: the peak value and the stretch of the offer. The preference profiles of the agents used were in complete opposition: the FBM agent wants to minimize the value of the issues and the Bayesian agent tries of maximize it. In the experiments we performed, the β parameter that defines whether an agent makes bigger concessions in the beginning of the negotiation (Conceder) or at the end (Boulware) was varied, see Table 10.2.

Agents	Utility						
	$\beta=0.02$	$\beta=0.1$	$\beta=0.5$	$\beta=1$	$\beta=2$	$\beta=10$	$\beta=50$
FBM Agent	0.898	0.897	0.734	0.585	0.449	0.193	0.060
Bayesian Agent	0.102	0.103	0.266	0.415	0.551	0.807	0.940

Table 10.2 Utility values of the FBM and Bayesian agents

In a single issue negotiation there is no possibility for a “win-win" outcome and all negotiation outcomes are Pareto efficient. One of the more important aspects of a negotiation strategy for a single issue negotiation is how fast it concedes to the opponent. As a result, for $\beta > 1$ the FBM agent implements a Conceder tactic and the FBM agent under performs with respect to the Bayesian agent that makes linear concessions in this case because no moves towards the Pareto frontier are possible. When the FBM agent employs a Boulware tactic ($\beta < 1$) the Bayesian agent starts conceding significantly and the result is a much lower utility for the Bayesian agent.

Bayesian Tit-for-Tat Agent

As discussed, the main objective associated with a negotiation strategy is to gain the best agreement possible in a negotiation. Utility of an agreement, therefore, measures the efficiency of a strategy. For every negotiation domain and preference profile the utility of agreements achieved by a strategy were averaged over

Negotiation Domain	Negotiation Strategy			
	ABMP	Trade-Off	Bayesian Smart	Bayesian Tit-for-Tat
Car	16%	12%	13%	14%
Party domain	13%	9%	13%	14%
Service-Oriented	14%	17%	25%	38%
Employment contr.	11%	40%	44%	47%
AMPO vs City	10%	13%	14%	20%

Table 10.3 Increase in utility for the Bayesian Tit-for-Tat strategy relative to the Random Walker strategy

all opponent strategies in the tournament. We assume that an efficient negotiation strategy should perform better than the Random Walker strategy. Therefore, we calculate the percentage of the utility increase compared to the utility of the Random Walker strategy (see Table 10.3).

The results show that on all domains the Bayesian Tit-for-Tat strategy performs better than all other strategies currently available in the negotiation repository. Only on the 2nd hand car negotiation domain the Bayesian Tit-for-Tat strategy is outperformed by the ABMP strategy. As in this domain a concession-based strategy is very efficient, and ABMP aims to concede on all issues, this strategy does particularly well in this domain.

The most significant increase in the efficiency of the reached agreement is shown on the Employment contract negotiation domain. This negotiation domain is rather small and evaluations of the issue alternatives are predictable in this domain. Learning in such a domain is relatively simple and, as a result, the Bayesian Tit-for-Tat strategy shows excellent performance. The Trade-off strategy shows good performance as well, however, it does not perform as well as the Bayesian Tit-for-Tat strategy. The ABMP strategy is significantly less efficient than the Bayesian Tit-for-Tat and the Trade-off strategies due to presence of issues with compatible preferences.

Similar results are obtained for the Service-Oriented Negotiation domain. This domain is much bigger than the Employment contract domain in terms of the possible agreements but has less issues. In addition, weights of the issues in the SON domain have bigger variation then in the Employment Contract domain where importance of the issues is more uniform. This explains the much lower efficiency of the Trade-Off strategy that is not capable of dealing with the weights of the issues. The Bayesian Tit-for-Tat strategy learns weights of the issues in the opponent preference profile and therefore shows a better performance.

AMPO vs City domain is the biggest domain in the repository. As is to be expected, the performance of the learning technique used in the Bayesian Tit-for-Tat strategy degrades in such bigger negotiation domains. This explains the lower relative increase in Table 10.3.

10.6.3 Approximating Auction Mechanism with Negotiation

In Section 10.4.2 we introduced a one-to-many negotiation protocol that approximates an auction mechanism. Here we present experimental results received for the proposed negotiation protocol. Figure 10.9 shows the histograms of the differences in utilities between the outcomes received with the original auction mechanism and the negotiation protocol.

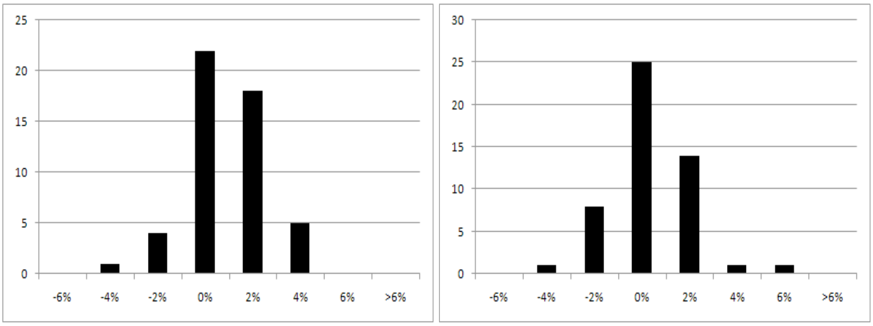

Fig. 10.9 Histograms of the differences in the utilities of experimental and theoretical outcomes for the buyer (left) and the seller (right).

The winner predicted by the mechanism and the negotiation protocol coincide 100%. This means that the negotiation protocol does not change the results of the first round in which a seller is selected as winner. Moreover, in the second round, in general the outcomes obtained by negotiation are also quite close to those determined by the mechanism. That is, in 78% of the experiments the deviation is less than 5%. The standard deviation of the difference between the mechanism outcome and the experimental results is 4%, and in 94% of the experiment the deviation did not differ with more than 10%, indicating that overall outcomes were reasonably close to the mechanism outcome with a few exceptions. This means that the negotiating agents that can learn are able to approximate the outcome determined by the mechanism quite well.

10.7 Conclusion

SAMIN, the system for analysis of multi-issue negotiation introduced here, has proved to be a valuable tool to analyse the dynamics of human-human closed negotiation against a number of dynamic properties. Our analysis shows that humans find it difficult to guess where the Pareto Efficient Frontier is located, making it difficult for them to accept a proposal. Although humans apparently do not negotiate in a strictly Pareto-monotonous way, when considering larger intervals, a weak monotony can be discovered. Such analysis results can be useful in two different ways: to train human negotiators, or to improve the strategies of software agents. Clear from our research so far, is that five key factors shape the outcome of a bilateral negotiation with incomplete information: (i) knowledge about the negotiation domain (e.g. the market value of a product or service), (ii) oneŠs own and oneŠs opponentŠs preferences, (iii) process attributes (e.g. deadlines), (iv) the negotiation strategies, and (v) the negotiation protocol.

The use of agent technology for negotiation systems has been a big help in both the design and the implementation of the SAMIN system. Principled design methods for agents and multi-agent systems such as DESIRE ensured a transparent design that properly reflects the interests of the stakeholders (researchers) and negotiators (human and software agent). The organization makes it easy to run tournaments with any number of agents, and over a number of negotiation domains. The interface and adapters to connect agents to the negotiation environment have been clearly specified which enable an easy integration of heterogeneous negotiating agents. The graphical user interfaces support both researchers and human subjects participating in experiments.

A good start has been made in the development of a toolkit for analysis in SAMIN, but more work needs to be done. Additional research on ontologies for negotiation is required to make this feasible; for example, we cannot currently formulate associated constraints on the domain of negotiation that must be satisfied for an agreement to be acceptable. More technically, components for web integration as well as extensions of adapters need to be developed, e.g., in order to handle more generic ontologies.

Acknowledgements This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VIVI-project 08075.

References

1. Ashri, R., Rahwan, I., Luck, M.: Architectures for negotiating agents. In: The 3rd Int. Central And Eastern European Conf. on Multi-Agent Systems (2003)
2. Bosse, T., Jonker, C., Treur, J.: Experiments in human multi-issue negotiation: Analysis and support. In: Jennings, N., Sierra, C., Sonenberg, L., Tambe, M. (eds.) Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'04, p. 672 Ű 679. IEEE Computer Society Press (2004)
3. Brazier, F., Dunin-Keplicz, B., Jennings, N., Treur, J.: Formal specification of multi-agent systems: a real world case. International Journal of Co-operative Information Systems, IJCIS **6(1)**, 67–94 (1997)
4. Coehoorn, R., Jennings, N.: Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In: Proceedings of the 6th International Conference on E-Commerce, pp. 59–68 (2004)
5. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Int. Journal of Robotics and Autonomous Systems **24**(3-4), 159–182 (1998)

6. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make negotiation tradeoffs. Journal of Artificial Intelligence **142**(2), 205–237 (2003)
7. Fisher, R., (and for the latest edition B. Patton), W.U.: Getting to Yes: Negotiating Agreement Without Giving In. Penguin Books (1981, 1992, 2003)
8. Gode, D.K., Sunder, S.: Allocative efficiency in markets with zero intelligence (zi) traders: Market as a partial substitute for individual rationality. Journal of Political Economy **101**(1), 119–137 (1993)
9. Ha, V., Haddawy, P.: Similarity of personal preferences: Theoretical foundations and empirical analysis. Artificial Intelligence **146**(2), 149–173 (2003)
10. Harrenstein, P., Mahr, T., de Weerdt, M.M.: A qualitative vickrey auction. In: Endriss, U., Paul W, G. (eds.) Proceedings of the 2nd International Workshop on Computational Social Choice, pp. 289–301. University of Liverpool (2008). URL http://www.st.ewi.tudelft.nl/.pdf
11. Hindriks, K., Jonker, C., Tykhonov, D.: Negotiation dynamics: Analysis, concession tactics, and outcomes. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07), pp. 427–433 (2007)
12. Hindriks, K., Jonker, C., Tykhonov, D.: Using opponent models for efficient negotiation (extended abstract). In: Decker, Sichman, Sierra, Castelfranchi (eds.) Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009) (2009)
13. Hindriks, K., Jonker, C.M., Tykhonov, D.: Towards an open negotiation architecture for heterogeneous agents. In: Proceedings of 12th International Workshop CIA 2007 on Cooperative Information Agents, Lecture Notes in AI. Springer-Verlag (2008)
14. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: Proceedings of the AAMAS 2008 (2008)
15. Hindriks, K., Tykhonov, D.: Towards a quality assessment method for learning preference profiles in negotiation. In: Proceedings of the AMEC 2008 (2008)
16. Hindriks, K.V., Tykhonov, D., de Weerdt, M.: Approximating an auction mechanism by multi-issue negotiation. In: Hindriks, K.V., Brinkman, W.P. (eds.) Proceedings of the First International Working Conference on Human Factors and Computational Models in Negotiation (HuCom2008), pp. 33–38 (2008)
17. Jonker, C., Treur, J.: An agent architecture for multi-attribute negotiation. In: Nebel, B. (ed.) Proceedings of the 17th International Joint Conference on AI, IJCAI'01, pp. 1195 – 1201 (2001)
18. Jonker, C.M., Robu, V., Treur, J.: An agent architecture for multi-attribute negotiation using incomplete preference information. Journal of Autonomous Agents and Multi-Agent Systems **15**(2), 221–252 (2007). DOI http://dx.doi.org/10.1007/s10458-006-9009-y
19. Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Negotiating complex contracts. Paper 125 of the Center for eBusines@MIT. http://ebusiness.mit.edu. (2001)
20. Kowalczyk, R., Bui, V.: On constraint-based reasoning in e-negotiation agents. In: Dignum, F., Cortés, U. (eds.) Agent-Mediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce Systems, Lecture Notes in Computer Science, pp. 31–46. Springer Ű Verlag (2003)
21. Lai, G., Sycara, K.: A generic framework for automated multi-attribute negotiation. Group Decision and Negotiation **18**(2), 169–187 (2009)
22. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3 edn. Prentice Hall PTR (2004)
23. Lin, R., Kraus, S., Wilkenfeld, J., Barry, J.: An automated agent for bilateral negotiation with bounded rational agents with incomplete information. In: Proc. of the 17th European Conference on Artificial Intelligence (ECAIŠ06), pp. 270–274 (2006)
24. Lin, R., Kraus, S., Wilkenfeld, J., Barry, J.: Negotiating with bounded rational agents in environments with incomplete information using an automated agent. Artificial Intelligence Journal **172**(6-7), 823–851 (2008)
25. Mnookin, R., Peppet, S., Tulumello, A.S.: Beyond Winning: Negotiating to Create Value in Deals and Disputes. Harvard University Press (2000)

26. Nadler, J., Thompson, L., van Boven, L.: Learning negotiation skills: Four models of knowledge creation and transfer. Journal of Management Science **49**(4), 529–540 (2003)
27. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press (1994)
28. Pruitt, D.: Negotiation Behavior. Academic Press (1981)
29. Raeesy, Z., Brzostwoski, J., Kowalczyk, R.: Towards a fuzzy-based model for human-like multi-agent negotiation. In: Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology, pp. 515–519 (2007)
30. Raiffa, H., Richardson, J., Metcalfe, D.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Harvard University Press (2003)
31. Sandholm, T.: Multi-agent Systems: A Modern Introduction to Distributed Artificial Intelligence, chap. Distributed rational decision making. MIT Press (1999)
32. Thompson, L.: The Heart and Mind of the Negotiator. Pearson Prentice Hall (2005)
33. Zeng, D., Sycara, K.: Benefits of learning in negotiation. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97) (1997)
34. Zeng, D., Sycara, K.: Bayesian learning in negotiation. International Journal of Human Computer Systems **48**, 125–141 (1998)