A System for Analysis of Multi-Issue Negotiation

Tibor Bosse, Catholijn M. Jonker, Lourens van der Meij,
Valentin Robu and Jan Treur

Abstract. This paper presents a System for Analysis of Multi-Issue Negotiation (SAMIN). The agents in this system conduct one-to-one negotiations, in which the values across multiple issues are negotiated on simultaneously. It is demonstrated how the system supports both automated negotiation (i.e., conducted by a software agent) and human negotiation (where humans specify their bids). To analyse such negotiation processes, the user can enter any formal property deemed useful into the system and use the system to automatically check this property in given negotiation traces. Furthermore, it is shown how, compared to fully closed negotiation, the efficiency of the reached agreements may be improved, either by using incomplete preference information revealed by the negotiation partner or by incorporating a heuristic, through which an agent uses the history of the opponent's bids in order to guess his preferences.

1. Introduction

Negotiation is a process by which a joint decision is made by two or more parties [9]. 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 them closer to an agreement. The field of negotiation can be split into different categories, e.g. along the following lines:

- one-to-one versus more than two parties
- single- versus multi-issues
- closed versus open
- mediator-based versus mediator-free

The research reported in this article concerns one-to-one, multi-issue, (partially) closed, mediator-free negotiation. For more information on negotiations between more than two parties (e.g., in auctions), the reader is referred to, e.g., [12]. In single-issue negotiation, the negotiation focuses on one aspect only (typically price) of the concept under negotiation. Multi-issue negotiation (also called multi-attribute negotiation) is often seen a more cooperative form of negotiation, since often an outcome exists that brings joint gains for both parties, see [10].

Closed negotiation means that no information regarding preferences is exchanged between the negotiators. The only information exchanged is formed by the bids. In partially open negotiation some information regarding preferences is exchanged, and in completely open negotiation all information is exchanged. More information about (partially) open negotiations can be found, e.g., in [7] and [10]. However, the trust necessary for open negotiations is not always available.

The use of mediators is a well-recognised tool to help the parties in their negotiations, see e.g., [6, 10]. The mediator aims for 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 than fair with respect to personal gain.

The literature on closed, multi-issue, one-to-one negotiation without mediators covers both systems to (partially) automate the negotiation process, and more analytic research focused on properties of the negotiation process and negotiation space. Based on a literature study and on our own analysis, a number of properties are presented here that focus largely on the dynamics of the negotiation process itself and on the results of the negotiation.

The SAMIN system presented in this paper has been developed to support and formally analyse such negotiation processes, i.e., multi-issue, (partially) closed, one-to-one negotiations without mediators. The system requires three types of input:

(1) a negotiation *trace* (or a set of traces)
(2) a set of *dynamic properties* considered relevant for the negotiation process
(3) the *negotiation profiles* of the participants

A *trace* is a sequence of bids by the negotiators. A *dynamic property* is an (informal, semi-formal or formal) expression that might or might not hold for a certain trace. An example of a simple dynamic property is bid-alternation, i.e., after communicating a bid to another agent, the agent remains silent until it has received a new bid from the other agent. A *negotiation profile* is a description of the preferences of the agent within the particular negotiation domain. The profiles together define the space of possible and efficient outcomes and are, therefore, essential for the creation of a complete analysis of the performance of a negotiator.

The most important measure of efficiency in bilateral negotiations, cf. [21], is

Pareto-efficiency. An outcome is said to be Pareto-efficient if the utility of any party cannot be improved without a loss of utility for another. The set of all Pareto-efficient outcomes form the Pareto-efficient frontier. The distance of an outcome to the Pareto frontier gives a measure of efficiency of a bid.

The SAMIN system consists of three components: an *Acquisition Component*, an *Analysis Component* and a *Presentation Component*. The Acquisition Component is used to acquire the input necessary for analysis. The Analysis Component performs the actual analysis, and the Presentation Component presents the results of the analysis in a user-friendly format.

SAMIN can check automatically whether selected properties hold for the traces under analysis. Such an analysis provides a means to improve bidding strategies and bidding protocols, both for human negotiators and for software agents in automated negotiation systems. Beside introduction of the SAMIN system, a subgoal of this paper is to report some results of such analyses, focusing both on human and automated negotiators. Regarding *human-human negotiation*, the results will be presented of applying SAMIN in the analysis of empirical traces obtained from an experiment in multi-issue negotiation about second hand cars. In the experiment the efforts of 74 humans negotiating against each other have been analysed using SAMIN. Regarding *automated negotiation*, SAMIN has been used to analyse the efficiency of the reached agreements by software agents, in order to improve the strategies of these software agents. To this end, a mechanism has been modelled in which agents are able to use any amount of incomplete preference information revealed by the negotiation partner. It is shown that the outcome of such a negotiation can be further improved by incorporating a "guessing" heuristic, by which an agent uses the history of the opponent's bids to predict his preferences. Experimental evaluation shows that the combination of using incomplete preference information with the guessing heuristic leads to agreement points close to or on the Pareto Efficient Frontier.

In Section 2 formalisation of negotiation process dynamics will be discussed in terms of negotiation states, transitions, and traces. Section 3 explains the formal specification of dynamic properties and presents example dynamic properties relevant for (partially) closed multi-issue one-to-one negotiations. The architecture of the SAMIN system is presented in Section 4. Section 5 illustrates how SAMIN can be used to analyse human negotiation processes. Some experiments in human multi-issue negotiation are described and analysed, and the results of the analysis are discussed. Next, Section 6 shows how SAMIN can be used to analyse automated negotiation processes. It is shown how software negotiators can be improved using two strategies (guessing and limited information sharing) that can be used alone, or in combination. Finally, Section 7 discusses related work, and Section 8 provides conclusions and some planned future work.

2. Formalising Negotiation Process Dynamics

Negotiation is essentially a dynamic process. To analyse those dynamics, it is, therefore, relevant to formalise and study dynamic properties of such processes. For example, how does a bid at a certain point in time compare to bids at previous time points? The formalisation introduced in this section is based on the notion of negotiation process state, negotiation transition and negotiation trace.

2.1 Formalising States of a Negotiation Process

The state of a (one-to-one) negotiation process at a certain time point can be described as a combined state consisting of two states for each of the negotiating agents: S = < S1, S2 > , where S1 is the state of agent A, and S2 is the state of agent B.

Each of these states include,

- the agent's own most recent bid
- its evaluation of its own most recent bid
- its evaluation of the other agent's most recent bid
- the history of bids from both sides and evaluations

To describe negotiation states a state ontology Ont is used. Example elements of this ontology are a sort BID for bids, and relations such as utility(A, b, v) expressing that A's overall evaluation of bid b is a real number v between 0 and 1. Based on this ontology the set of ground atoms At(Ont) can be defined. A state is formalised as any truth assignment: At(Ont) $\rightarrow$ {t, f} to this set of ground atoms. The set of all states described by this ontology is denoted by States(Ont).

2.2 Negotiation Transitions

A particular negotiation process shows a sequence of transitions from one state S from States(Ont) to another (next) state S' from States(Ont). A transition S $\rightarrow$ S' from a state S to S' can be classified according to which agents are involved. During such a transition each of the main state components (S1, S2) of the overall state S may change. The simplest types of transition involve a single component transition. For example, when one agent generates a bid, while the other agents is just waiting: a transition of type S1 $\rightarrow$ S1 or S2 $\rightarrow$ S2. Next come transition types where both components are involved. For example, when a communication from agent A to agent B takes place, changing the state S2 of agent B: a transition of type S1 x S2 $\rightarrow$ S2. Notice that in principle, also more complex transition types are possible, involving changes of both state components at the same time, i.e., S1 x S2 $\rightarrow$ S1 x S2. In organised cooperations between multiple agents the complexity of the types of transitions is often limited by regulation of the organisation. For example, in organised negotiation processes, usually it is assumed in the protocol that after communicating a bid to the other agent, the agent remains silent until it has received a new bid from the other agent (see the dynamic property 'bid alternation' in Sections 3 and further below). Such an assumption about the protocol implies that the transitions involved in the negotiation are only of the simpler types mentioned above.

2.3 Negotiation Traces

Negotiation traces are time-indexed sequences of negotiation states, where each successive pair of states is a negotiation transition. To describe such sequences a fixed *time frame* T is assumed which is linearly ordered. A *trace* $\mathcal{T}$ over a state ontology Ont and time frame T is a mapping $\mathcal{T}$: T → STATES(Ont), i.e., a sequence of states $\mathcal{T}_t$ (t ∈ T) in STATES(Ont). The set of all traces over state ontology Ont is denoted by TRACES(Ont). Depending on the application, the time frame T may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering.

3. Dynamic Properties of Negotiation Processes

This section presents a classification of dynamic properties of negotiation processes along with examples of each class. Before presenting the classification and the specific dynamic properties of negotiation, the formal method for specifying those properties is presented.

3.1 Specification of Dynamic Properties

Specification of dynamic properties of a negotiation process can be done in order to *analyse* its dynamics, for example to find out how certain properties of a negotiation process as a whole relate to properties of a certain subprocess, or to verify or evaluate a negotiation model. To formally specify dynamic properties that express characteristics of dynamic processes (such as negotiation) from a temporal perspective an expressive language is needed. To this end the *Temporal Trace Language* TTL is used as a tool; cf. [5], which is briefly defined as follows.

The set of *dynamic properties* DYNPROP(Ont) is the set of temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace $\mathcal{T}$ over state ontology Ont, a certain state of the agent A during a negotiation process at time point t is indicated by state($\mathcal{T}$, t, A). In the third argument, instead of A also specific parts of A can be used, such as input(A), or output(A), which refer to observations and actions by A, respectively. These state indicators can be related to state properties via the formally defined satisfaction relation |=, comparable to the Holds-predicate in the Situation Calculus: state($\mathcal{T}$, t, A) |= p denotes that state property p holds in trace $\mathcal{T}$ at time t in the state of agent A. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic with sorts T for time points, Traces for traces and F for state formulae, using quantifiers and the usual first-order logical connectives such as ¬, ∧, ∨, ⇒, ∀, ∃. As an example, consider the dynamic property bid alternation, which states that for all two different moments in time t1, t3, that A generates a bid, there is a moment in time t2, with t1 < t2 < t3, such that A received a bid generated by B. In formal TTL-format, this property is expressed as:

bid_alternation(γ:TRACE) ≡
∀ A, B: AGENT, ∀ b1, b3: BID, ∀ t1, t3: time :
t1 < t3 &
state(γ, t1, output(A)) |= to_be_communicated_to_by(b1, B, A) &
state(γ, t3, output(A)) |= to_be_communicated_to_by(b3, B, A)
⇒ ∃b2: BID, ∃t2: time : t1 < t2 < t3 &
state(γ, t2, input(A)) |= communicated_to_by(b2, A, B)

Usually for reasons of presentation dynamic properties are expressed in informal or semi-formal forms.

3.2 Classes and Examples of Dynamic Properties of Negotiation

The properties relevant for analysing the dynamics of (partially) closed multi-issue one-to-one negotiation, can be divided into the following types:

- **Bid properties** give some information about a specific bid. They are usually defined in terms of the negotiation space and the profiles of the negotiators. Bid properties concern, for example, the Pareto efficiency of a bid.
- **Result properties** are a subset of the set of bid properties, concerning only the last bid of a negotiation process (i.e., the final agreement).
- **Bid comparison properties** compare two arbitrary bids with each other. An example is domination: a bid b1 dominates a bid b2 with respect to agents A and B iff both agents prefer bid b1 over bid b2; see below for a formalisation
- **Step properties** are a subset of the set of bid comparison properties, concerning only the transitions between successive bids. Hence, they are restricted to the combinations of bids of one party that directly follow each other.
- **Limited interval properties** concern parts of traces. Basically, they state that each step in a certain interval satisfies a certain step property. For instance: a negotiation process is Pareto-monotonous for the interval [t1, t2] iff for all successive bids b1, b2 in the interval b2 dominates b1 (see below).
- **Trace properties** are a subset of the set of limited interval properties, concerning whole traces.
- **Multi-trace properties** compare the dynamics observed in more than one trace. An example is Better Negotiator: agent A is a better negotiator than agent B iff in more than 60% of the negotiations between A and B, the deal reached is more to the advantage of agent A than of agent B.
- **Protocol properties** specify certain constraints on the negotiation protocol. A specific instance is: over time the bids of negotiators A and B alternate.

Note that the first two types are basically *static properties*, whereas the other types are *dynamic properties*: they specify behaviour over time. In [1] for each of these types a number of properties are described in detail, both in informal and in formal notation. In this paper, only a small selection of relevant properties is presented.

configuration_differs(b1:BID, b2:BID) ≡

∃a: ISSUE, ∃v1, v2: VALUE :
value_of(b1, a, v1) & value_of(b2, a, v2) & v1 ≠ v2

This bid comparison property states that two bids b1 and b2 differ in configuration iff there is an issue that has a different value in both bids. For example, in bid b1 the value of the issue “color” is “red”, whereas in bid b2 this value is “blue”. Similar properties can be defined stating that two bids differ in configuration in at least x issues. This property can also be used as a building block to specify a step property, e.g. “in the view of agent A, agent B varies the configuration, but not the utility”. Such a property are useful to find out what kind of opponent the negotiator is dealing with.

strictly_dominates(b1:BID, b2:BID, A:AGENT, B:AGENT) ≡
∀vA1, vA2, vB1, vB2 : real :
util(A, b1, vA1) & util(A, b2, vA2) & util(B, b1, vB1) & util(B, b2, vB2) ⇒ vA1 > vA2 & vB1 > vB2

This bid comparison property states that a bid b1 dominates a bid b2 with respect to agents A and B iff both agents prefer bid b1 over bid b2. This notion is related to Pareto Efficiency, see e.g., [10]. The property could also be changed to weakly_dominates by changing the > sign into the ≥ sign. Moreover, it can be used as a building block to specify step properties, limited interval properties (see the next property), and trace properties.

strict_pareto_monotony(γ:trace, tb:time, te:time) ≡
∀t1, t2: time, ∀A, B: AGENT, ∀b1, b2: BID :
[tb ≤ t1 < t2 ≤ te & is_followed_by(γ, A, t1, b1, B, t2, b2)]
⇒ state((γ, t2) |= strictly_dominates(b2, b1, A, B)

This limited interval property makes use of the previous property. It states that a negotiation process γ is strictly Pareto-monotonous for the interval [t1, t2] iff for all successive bids b1, b2 in the interval b2 dominates b1. By choosing for tb and te respectively the start and end time of the process, the property can be transformed into a trace property. Generally, traces that satisfy this property are not abundant in (human) real world multi-issue negotiations, since if the profiles of the two parties are strongly opposed (with emphasis on the same issues), even in multi-issue situations a gain for the one often implies a loss for the other. If, however, the profiles are less opposed, Pareto-monotony may occur.

pareto_inefficiency(γ:trace b:BID, A:AGENT, B:AGENT, ε:real) ≡
∀vA, vB : real :
util(A, b, vA) & util(B, b, vB) ⇒ pareto_distance(vA, vB) = ε

This bid property informally states that with respect to agents A and B, the Pareto inefficiency of a bid b is the number ε that indicates the distance to the Pareto Efficient Frontier according to some distance measure d in utilities. Here, d(b1, b2) is the distance between the bids b1 and b2 when viewed as points in the plane of utilities. The function to measure the distance in the plane can still be filled in, e.g., the sum of absolute differences of coordinates, or the square root of the sum of squares of the differences, or the maximum of the differences of the coordinates. The Pareto Efficient Frontier is the set of all bids b for which there is no other bid b’ that dominates b. Hence, in case the Pareto Inefficiency of a bid is 0, there is no other bid that dominates it. By filling in the resulting agreement of a negotiation for bid b, the property is transformed into a result property. In general, determining the number ε for which this property holds is a good measure for checking the success of the negotiation process. In a similar way, the property nash_inefficiency can be formulated, which calculates the distance from a certain bid to the Nash Point. This is the point (on the Pareto Efficient Frontier) for which the product of both utilities is maximal, see e.g., [10].

4. The SAMIN Architecture

SAMIN is a Prolog-based software environment that has been designed at the Vrije Universiteit Amsterdam for the analysis of multi-issue negotiation processes. Section 4.1 describes the role SAMIN can take in an analysis setting of negotiation processes. Next, Section 4.2 presents a top level overview of the SAMIN architecture. Basically, the system consists of three components: an *Acquisition Component*, an *Analysis Component* and a *Presentation Component*. These components are described in more detail in Sections 4.3, 4.4, and 4.5, respectively.

4.1 SAMIN in its Environment

The SAMIN system has been designed to work together in interaction with a human analyst and either human or software agent negotiators. As depicted in Figure 1, the analyst determines the properties that SAMIN is to use in the analysis of negotiation processes. He or she can select (and if necessary adapt) properties from SAMIN's library, or can construct new properties with the help of SAMIN's special dynamic property editor. SAMIN can only analyse a negotiation process if it has access to the profiles used by the different parties, and the bids exchanged between the parties. SAMIN does not influence the negotiation while it is being carried out, it only observes either during the negotiation, or afterwards.

The analysis result of one or more negotiations is presented to the human analyst. The analyst can use that result for purposes within Cognitive Science (e.g., to analyse human negotiation processes and train human negotiators) or Artificial Intelligence (e.g., to improve the strategies of software agents). Interesting for the future might be to present the results directly after the conclusion of the negotiation to a software agent negotiator that is capable of learning so that the agent can use the result to improve its negotiation skill by itself. A negotiation process can be monitored directly by SAMIN (if the agents allow interfacing), or the negotiation trace can be written to a file and be analysed in hindsight by SAMIN. The current version of SAMIN is developed especially for closed multi-issue one-to-one negotiations, entailing that the only information exchanged between the negotiators are the bids.

The input required by SAMIN (see Figure 1) consists of properties, profiles, and traces of bids. Its output consists of an analysis that can be presented in a user-friendly format (see Section 4.4 and 4.5). As mentioned before, SAMIN offers the user both a library of properties to choose from and a dynamic property editor to create new properties. Profiles can be obtained in two ways. Either the negotiator presents a pre-specified profile to SAMIN or the negotiator can use SAMIN's interactive profile editor to create it in SAMIN. Pre-specified profiles have to be in a format recognised by SAMIN. The trace of bids required by SAMIN can be obtained by SAMIN monitoring the bids exchanged between the negotiators during the negotiation process. This only requires the bids to be in a format recognised by SAMIN and the possibility to "over-hear" the communication between the negotiators. Another possibility is that the bids exchanged during a negotiation process are stored in a special file. If the bid-traces are in the right format, SAMIN can perform analysis on one or on a combination of such traces after the negotiation has been completed. If the negotiators wish to do so, they can use SAMIN's bid ontology editor to define what a bid should look like, before entering the negotiation phase. Construction of bid ontology and the profiles is part of the pre-negotiation phase [10].

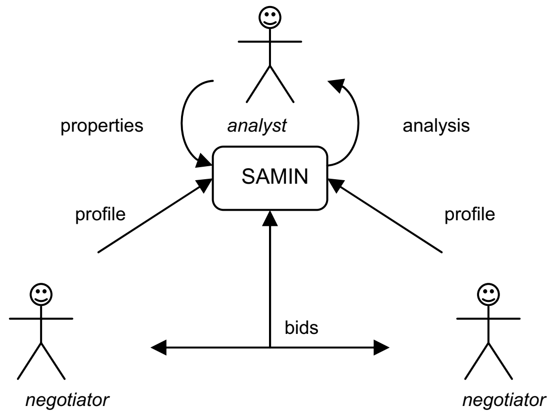


Figure 1. SAMIN in its environment

4.2 Top Level

At the top level, SAMIN consists of three components: an *Acquisition Component*, an *Analysis Component* and a *Presentation Component*, see Figure 2. Here, the solid arrows indicate data flow. The dotted arrows indicate that each component can be controlled separately by the analyst. The Acquisition Component is used to acquire the input necessary for analysis. The Analysis Component is used to perform the actual analysis (i.e., checking which properties hold for the negotiation process under analysis). Finally, the Presentation Component is used to present the results of the analysis in a user-friendly format. Furthermore, SAMIN maintains a library of properties, templates of properties, bid ontologies, and profile ontologies (not shown in Figure 2). The working of the three components will be described in detail in the next sections.

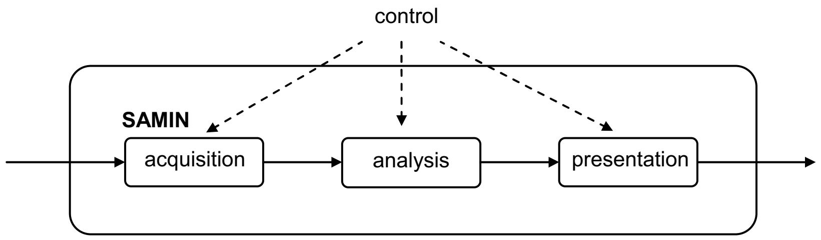


Figure 2. Global Overview of the SAMIN architecture

4.3 The Acquisition Component

The acquisition component is used to obtain the required input for the analysis. It consists of an *ontology editor*, a *dynamic property editor* and a *trace determinator*.

The ontology editor is used for the construction of bid ontologies and profile ontologies necessary to automatically interpret the bids exchanged by the negotiators, and to automatically interpret the profiles of the negotiators. The ontology editor is typically used to construct a bid ontology and a profile ontology, thus allowing the user to identify the issues to be negotiated, the values that each of these issues can take, and the structure of bids, in the bid ontology. Furthermore, in specifying the profile ontology the user identifies the possible evaluations that can be given to values, and the utility functions of bids.

The dynamic property editor supports the gradual formalisation of dynamic properties in TTL format. The editor offers a user interface that allows the analyst to construct dynamic properties, represented in a tree-like format.

The trace determinator can be used interactively with the analyst to determine what traces to use in the analysis. The user can interactively locate the files containing the traces to be checked. The traces themselves can be of three categories: (human) empirical traces, simulated traces, and mixed traces. An empirical trace is the result of an existing human negotiation process. A simulated trace is the result of an automated negotiation process. A mixed trace is the result of a human negotiating with a software agent. To support the acquisition of traces of all three types, a dedicated interface has been created for SAMIN.

4.4 The Analysis Component

The analysis component currently consists of a *logical analyser* that is capable of checking properties against traces. To this end, the tool takes a dynamic property in TTL format and one or more traces as input, and checks whether the dynamic property holds for the traces.

Traces are represented by sets of Prolog facts of the form *holds(state(m1, t(2)), a, true)* where m1 is the trace name, t(2) time point 2, and a is a state property as introduced in Section 3.1. The above example indicates that state formula a is true in trace m1 at time point 2. The Analysis Component basically uses Prolog rules for the predicate sat that reduce the satisfaction of the temporal formula finally to the satisfaction of atomic state formulae at certain time points, which can be read from the trace representation. Examples of such reduction rules are:

```
sat(and(F,G)) :- sat(F), sat(G).
sat(not(and(F,G))) :- sat(or(not(F), not(G))).
sat(or(F,G)) :- sat(F).
sat(or(F,G)) :- sat(G).
sat(not(or(F,G))) :- sat(and(not(F), not(G))).
```

In addition, if a dynamic property does not hold in a trace, then the software reports the places in the trace where the property failed.

4.5 The Presentation Component

The presentation component currently includes a tool that visualises the negotiation space in terms of the utilities of both negotiators. This *visualisation tool* plots the bid trajectory in a 2-dimensional plane, see Figure 3. The utilities are real values that indicate how a particular bid is evaluated by a negotiator. Details about the calculation of utilities are provided in the next Section.

In Figure 3, the seller's utility of a bid is on the horizontal axis, and the buyer's utility is on the vertical axis. The light area corresponds to the space of possible bids. In this area, each curve is a continuous line, corresponding to a different combination of discrete issues. The specific position on the line is determined by the continuous issue 'price'. Since in this particular domain 4 discrete issues with 5 possible values occur (see next section), there are already 625 (= 5^4) different curves. In this Figure, the sequences of actual bids made by both buyer (left) and seller (right) are indicated by the dark points that are connected by the two angular lines. The upper-left point indicates the buyer's first bid, and the lower-right point indicates the seller's first bid. The dotted line indicates the Pareto Efficient Frontier according to the profiles of the negotiating agents, and the short dark lines show the distance from each bid to this frontier. The small dot that is plotted on the Pareto Efficient Frontier (on the right) corresponds to the Nash Point. From this picture, it is clear that both negotiators make more and more concessions (their bids converge towards each other). Eventually, they reach a point that does not lie on the Pareto Efficient Frontier, but is rather close to it anyhow.

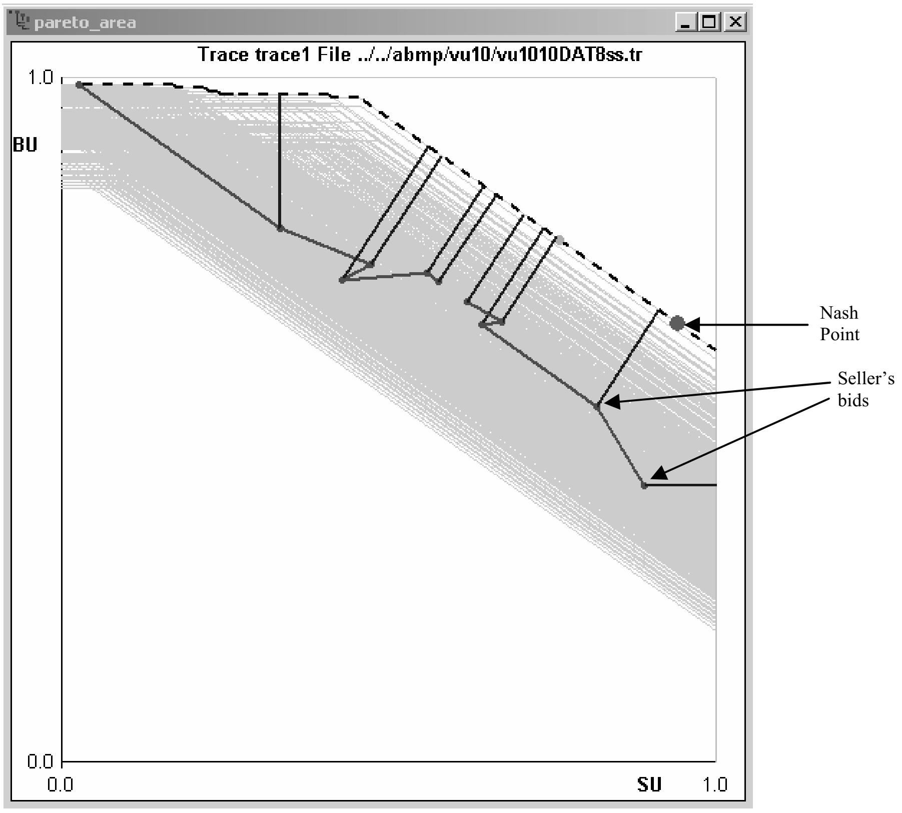


Figure 3. Visualisation Tool

5. Human Multi-Issue Negotiation Experiments

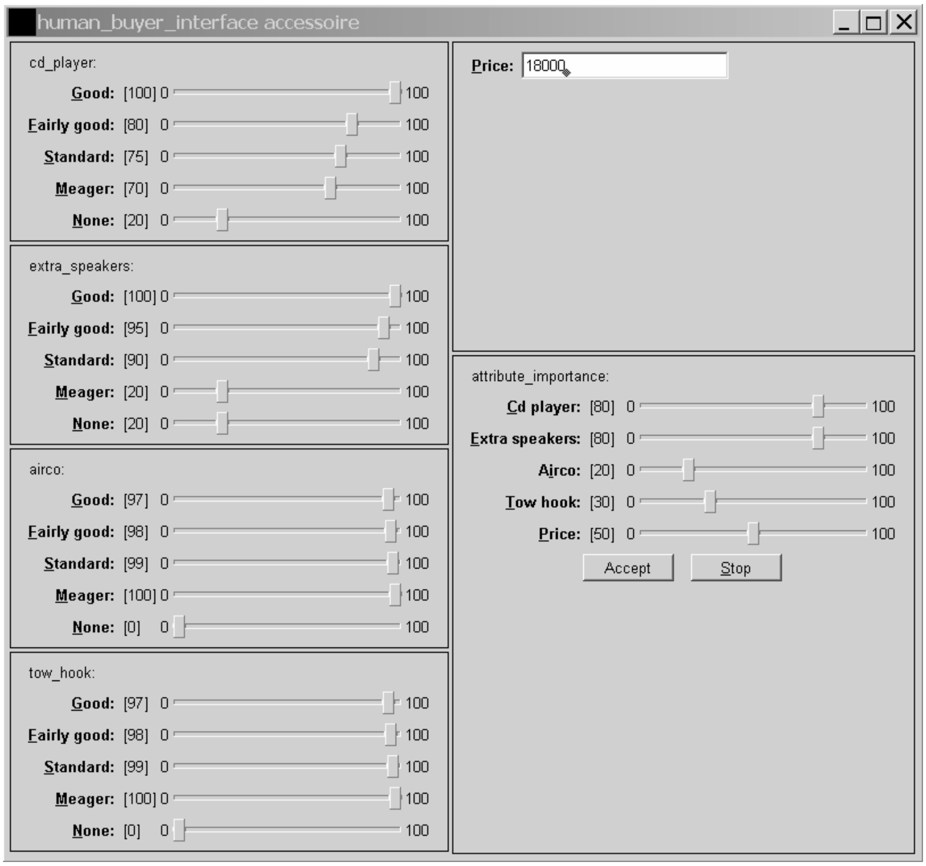
To illustrate the use of analysing human multi-issue negotiation processes, SAMIN has been applied in a case study. As mentioned in Section 4, the analysis component of SAMIN takes traces and formally specified dynamic properties as input and checks whether a property holds for a trace. Using automatic checks of this kind, some of the properties provided in Section 3 have been checked against empirical traces generated by students during practical sessions in multi-issue negotiation. The domain of the case study, a negotiation about second hand cars, is presented in detail in Section 5.1. Section 5.2 describes the setup of the experiments performed in the case study. The analytical results of the acquired traces will be shown in Section 5.3.

5.1 Domain: second hand cars

The protocol used in the experiments is an alternating-offers protocol. In this type of negotiation, a bid has the form of values assigned to a number of issues of the object under negotiation. Here, the object of negotiation is a particular second hand car. Within this domain, the relevant issues are cd_player, extra_speakers, airco, tow_hook and price. Consequently, a bid consists of an indication of which CD player is meant, which extra speakers, airco and tow hook, and what the price of the bid is. The goal of the negotiators is to find agreement upon the values of the four accessories and the price. Here, the price issue has a continuous value, whilst the other four issues have a discrete value from the set {good, fairly_good, standard, meager, none}. These values are assumed to be objective indicators from a consumer organisation, so there can be no discussion about whether a certain CD player is good or fairly good.

Before the negotiation starts, both parties specify their *negotiation profile*: for all issues with discrete values they have to assign a number to each value, indicating how satisfied they would be with that particular value for the issue (e.g. "I would be very happy to buy/sell a good CD player, a bit less happy with a fairly good CD player, …" and so on). The buyer also has to indicate what is the maximum amount of money (s)he would be willing to spend. Moreover, both parties have to assign a number to each of the issues, indicating how important they judge that issue (e.g. "I don't care that much which CD player I will buy/sell"). Notice that this does not conflict with the above statements. An example negotiation profile for a buyer is shown in Figure 4. In addition to this negotiation profile, the seller is provided with a *financial profile*. This is a list of all issues, where for each issue it is indicated how much it costs, both to buy it and to build it into the car. Since we focus on closed negotiation, none of the profiles will be available for the other party. However, SAMIN has access to both profiles.

When both parties have completed their profiles, the negotiation starts. To help human negotiators generating their bids, the system offers a special tool that calculates the utility of a bid before it is passed to the opponent.
The utility U_B of a bid B is defined by the weighted sum over the issue evaluation values $E_{B,j}$ for the different issues denoted by: $U_B = \Sigma_j w_j \ E_{B,j}$. The weight factors w_j are based on the issue importance factors. Here scaling takes place (the sum of weight factors is made 1, and the evaluation values $E_{B,j}$ are between 0 and 1) so that the utility is indeed is between 0 and 1; for more details, see [4]. Since the negotiators have individual negotiation profiles, for each bid the seller's utility of the bid is different from the buyer's utility of the bid.



cd_player:

Good:	[100]	0	100
Fairly good:	[80]	0	100
Standard:	[75]	0	100
Meager:	[70]	0	100
None:	[20]	0	100

extra_speakers:

Good:	[100]	0	100
Fairly good:	[95]	0	100
Standard:	[90]	0	100
Meager:	[20]	0	100
None:	[20]	0	100

airco:

Good:	[97]	0	100
Fairly good:	[98]	0	100
Standard:	[99]	0	100
Meager:	[100]	0	100
None:	[0]	0	100

tow_hook:

Good:	[97]	0	100
Fairly good:	[98]	0	100
Standard:	[99]	0	100
Meager:	[100]	0	100
None:	[0]	0	100

Price: 18000

attribute_importance:

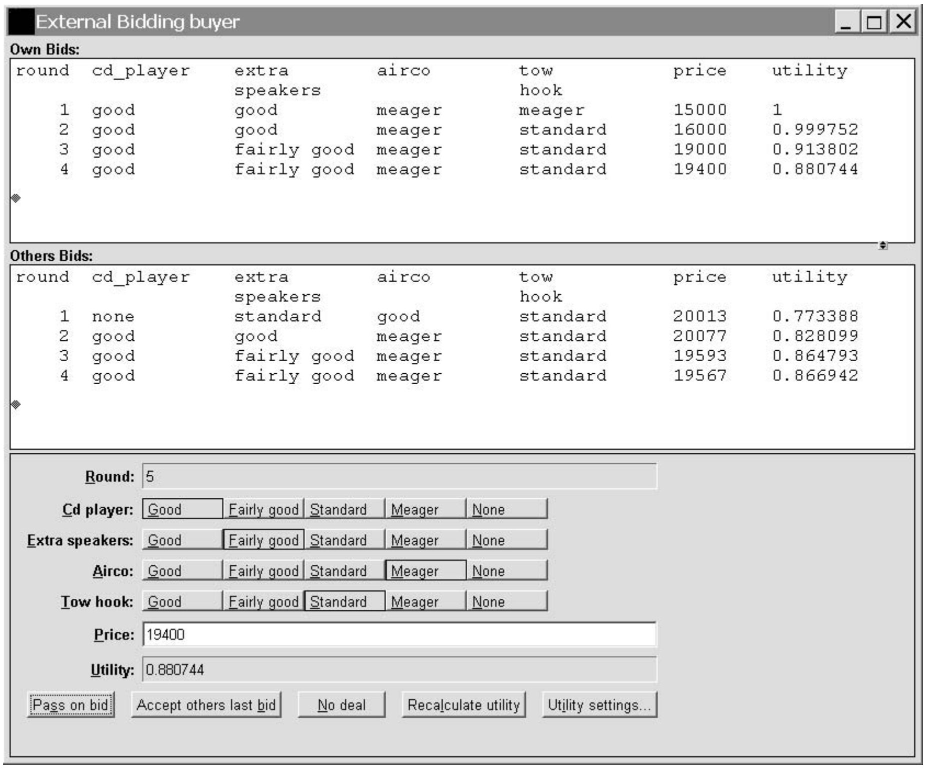
Cd player:	[80]	0	100
Extra speakers:	[80]	0	100
Airco:	[20]	0	100
Tow hook:	[30]	0	100
Price:	[50]	0	100

Figure 4. Example Buyer's Negotiation Profile

Besides for facilitating the bidding process, the profiles are used by SAMIN to analyse the resulting traces. For example, to check whether the property Pareto-Monotony holds (i.e., "For each combination of successive bids b1, b2 in the trace, both agents prefer bid b2 over bid b1"), the software must have a means to determine when an agent "prefers" one bid over another.

5.2 Experimental setup

Participants. 74 subjects participated in the experiment, in three different sessions. All sessions took place during a master class for students of the final classes of the VWO (a particular type of Dutch High School). The age of the students mostly was 17 years, but varied between 14 and 18 years. Most of them were males. In the first session, in March 2002, 30 students participated. In the second, in March 2003, 28 students participated. In the third session, in November 2003, 16 students participated.



Own Bids:

```
round  cd_player    extra         airco        tow           price    utility
                    speakers                   hook
    1  good         good          meager       meager        15000    1
    2  good         good          meager       standard      16000    0.999752
    3  good         fairly good   meager       standard      19000    0.913802
    4  good         fairly good   meager       standard      19400    0.880744
```

Others Bids:

```
round  cd_player    extra         airco        tow           price    utility
                    speakers                   hook
    1  none         standard      good         standard      20013    0.773388
    2  good         good          meager       standard      20077    0.828099
    3  good         fairly good   meager       standard      19593    0.864793
    4  good         fairly good   meager       standard      19567    0.866942
```

Round:	5
Cd player:	[✓ Good] Fairly good Standard Meager None
Extra speakers:	Good [✓ Fairly good] Standard Meager None
Airco:	Good Fairly good Standard [✓ Meager] None
Tow hook:	Good Fairly good [✓ Standard] Meager None
Price:	19400
Utility:	0.880744

Figure 5. Example Negotiation Trace

Method. Before starting the experiment, the participants were provided some background information on negotiation, and in particular about multi-issue negotiation. Some basic negotiation strategies were discussed. In addition, the second hand car example was explained. Then they were asked to start negotiating, thereby taking a profile in mind (that had to be specified first) aiming at obtaining the best possible deal, without showing their own profile to the opponent. The negotiation process was performed using different terminals over a network, which allowed each participant to negotiate with another anonymous participant. All negotiators could input their bids within a special interface. The resulting negotiation traces were logged by the system, so that they could be re-used for the purpose of analysis. A screenshot of an example negotiation trace is depicted in Figure 5. This trace is shown from the perspective of the buyer. In the upper part of the window, the buyer's own bids are displayed, including the buyer's utility for each bid. In the middle part, the bids of the seller are displayed, including the buyer's utility for each bid. The lower part consists of the bidding interface, which allows the buyer to input his bid and pass it to the seller.

5.3 Results of the Human Experiments

Using the SAMIN prototype, a number of relevant dynamic properties for multi-issue negotiation (also see Section 3) have been checked against the traces that resulted from the experiments. The results indicate that humans find it hard to guess where the Pareto Efficient Frontier is located. Due to space limitations, the detailed results are not shown in this paper. The interested reader is referred to [1].

6 Using Incomplete Information

The above sections concentrate on the analysis of human negotiations. In the current chapter it is shown how SAMIN can be used to analyse software negotiations. To this end, a particular software agent for negotiation was implemented and its performance was evaluated using SAMIN.

The software agent uses incomplete information in order to improve the outcome of automated multi-issue negotiations. The rationale for this research line is that in many electronic applications only a limited degree of trust exists between parties. This does not hold for many applications, where only a limited degree of trust exists between parties in sharing preference information. The reasons for this may be endogenous to the negotiation (e.g., fear the other may abuse this information to get a better deal) or exogenous (e.g., privacy concerns).

In classical multi-issue-utility theory ([21]), the solution proposed is the use of an independent mediator, which both parties can trust to reveal their preferences. The problem with this approach in an electronic or open system setting is that it can be difficult to establish whether a mediator is indeed impartial or more trustworthy than the negotiation partner himself. For example, an agent may have no way of knowing if the solutions proposed by the mediator are not biased towards the other or that his preference information will not be stored and used for other purposes. By contrast, our approach is to use a distributed design, in which each agent computes its own bids, using the information available about the preferences of the opponent. We take into account two different types of (incomplete) information:

- Partial profile information, which is communicated by the negotiation partner himself in the beginning of the negotiation.
- Profile information, which can be deduced (learned) from successive bids during the negotiation itself. Here we start from the assumption that the way the negotiation partner is bidding may reveal something about his preferences. For this mechanism we use the term "guessing" to clearly show it is a heuristic.

In our current work we preferred the heuristic approach to designing automated negotiation, since we feel this allows more flexibility. This position is supported, among others, by [17] who clearly show that "what is required are agent architectures that implement different search mechanisms, capable of exploring the set of possible outcomes under both limited information and computation assumptions". However, this does not mean we ignore the results from game theory: they are present in both measuring the efficiency of reached agreements (e.g., Pareto-efficiency) and in analysing some properties of our mechanism (incentive compatibility properties). To analyse the effectiveness of the strategies and the impact of revealing small bits of information in the negotiation, the strategies were implemented and tested within the environment of the SAMIN system.

6.1 Background of the model

As in the human experiments, our negotiation follows an alternating-offers protocol. A bid in such a negotiation has the form of values assigned to a number of issues. If the negotiation is about the sale of a car, then the relevant issues are again: CD player, extra speakers, airco, tow hook, price. Thus, a bid consists of an indication of which CD player is meant, which speakers, airco and tow hook, and what the price of the offer is.

Although the examples given are based on this domain, our negotiation model and its description presented in this chapter are generic. Instantiations in other domains are possible and have been considered – for example an employer and employee negotiating about work shifts and overtime pay (work performed in collaboration with Almende B.V, Rotterdam).

The current model represents an extension of the negotiation model presented in [4]. This paper presents two main directions in which the model was adapted (in [14]), after the publication of the original research:

- A mechanism where the agents are allowed to exchange and take into account partial preference information from the negotiation partner was modelled.
- A novel "guessing" heuristic by which an agent can estimate the preferences of the other using his past bids was proposed and tested.

Both for the original work and the extension, the DESIRE design method and software environment [15] were used to design the agents. Although we also cover some elements of the existing model, we only do so very briefly, to allow more extensive explanations for the parts that were added or adapted from the original research. For further details readers are asked to consult [4, 14, 26].

Our negotiation model works by performing computations on two levels: the overall bid level and the issue level. This involves first evaluating the utility opponent's previous bid, and then planning the target utility for the own next bid. Finally, the configuration of the next bid will be selected such that it fits this target value. In the design of our agent, these steps are modelled as separate components and our presentation follows this structure.

6.2 Bid Utility Determination and Planning

The evaluation for each issue is computed based on an evaluation function, specified by the agent owner (user) in the beginning of the negotiation. This function takes the generic form eval: `VS -> E`, where `VS` is either a finite set of discrete values or an infinite set of continuous values, while $E = [0,1]$. For example, in our domain accessories have discrete values (quality levels, assigned an evaluation by the user), while issues such as mileage or price are continuous, and their utility is computed by a continuous function. Next, the utility of the opponent's previous bid is computed. The *overall utility* U_B of a bid B is taken as a weighted sum of the *issue evaluation values* $E_{B,j}$ for the different issues j: $U_B = \Sigma_j\ w_j\ E_{B,j}$. Here all weights w_j are normalized importance factors based on the raw importance factors p_k for the different issues (provided by the user through an interface in the beginning of the negotiation): $w_j = p_j / \Sigma_k p_k$

Finally a target evaluation is computed for the agent's next bid. For determination of the next bid's *target utility* TU the following formula is used: $TU = U_{BS} + CS$, with U_{BS} the utility of the agent's own last bid, and the concession step CS determined as: $CS = \beta\ (1 - \mu / U_{BS})^* (U_{BO} - U_{BS})$, where U_{BO} is the utility of the opponent's last bid, with respect to the agent's own utility function. Factor β stands for *negotiation speed*, while factor $(1 - \mu / U_{BS})$ expresses that the concession step will decrease to 0 if the U_{BS} approximates a minimal utility μ. The minimal utility is a measure of how far concessions can be made.

6.3 Issue Planning

The *Issue Planning* component is shown in Figure 6. Here, the arrows denote data flow. This component determines the values of issues for the next bid, in such a way that the utility of the next bid equals the target utility. This is done in two steps: first a target evaluation is computed per issue, based on the target evaluation planned for the whole bid. Next, issue values are chosen with the evaluation closest to the target evaluations (for all issues except price). The configuration of the next bid is then completed by selecting a value for price, such that the utility of the final bid fits exactly its target.

In order to make better directed concessions, in planning the target evaluation for each issue we take into account not only the own preference weight of the agent, but also the weight of the opponent. If the opponent is not willing to reveal her preference weight for some (or maybe all) issues, an estimation of these weights is computed in "Estimation of Opponent's Parameters" component. The role of the "Guess Coefficients" component is to analyse the way the opponent is bidding and to provide some extra information to be used for estimating these private preference weights.

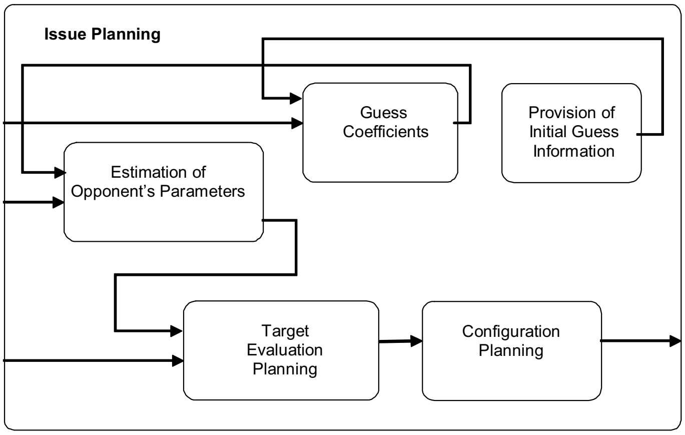

Issue Planning
Guess Coefficients
Provision of Initial Guess Information
Estimation of Opponent's Parameters
Target Evaluation Planning
Configuration Planning

Figure 6. Internal composition of Issue Planning

6.3.1 Target Evaluation Planning

This component outputs a target evaluation for each issue in the next bid, based on the bid target value.

The target issue evaluation is determined in two steps. First a basic target issue evaluation for each issue is computed as:

$$BTE_j = E_{BS,j} + (\alpha_j / N)(TU - U_{BS})$$

In the above formula $E_{BS,j}$ represents the evaluation for issue j in the agent's own previous bid, U_{BS} the overall evaluation of the agent's previous bid, while TU represents the target utility for the next bid (as shown in Section 6.2). The parameter α_j is chosen as $\alpha_j = (1 - w_j)(1 - E_{BS,j})$, where the first parameter expresses the influence of the user's own importance factor, while the second factor assures that the target evaluation values remain scaled in the interval between 0 and 1. Parameter N is a normalization factor, defined as: $N = \Sigma_j w_j \alpha_j$. By this choice we ensure that the following relation always holds: $\Sigma_j w_j BTE_j = TU$ (for a full proof of this property we refer the reader to [4]).

The Basic Target Evaluation, however only takes into accounts the own preference weights of the agent. Using only this value would work, but tests showed that it leads to sub-optimal results, since the preferences of the other are not considered in any way when making concessions. To improve on this, the following solution was implemented. For each issue $j \in I$ (where I denotes the set of all issues) a Preference Difference Coefficient δ_j is computed as:

$$\delta_j = (W_{other,j} - W_{own,j}) / (W_{other,j} + W_{own,j})$$

This coefficient (scaled between -1 and 1) expresses how different the preferences of the two parties for each issue are. Positive values for δ_j denote a stronger preference of the negotiation partner for issue j, while negative values denote a stronger own preference for this issue.

The concession to be made in each issue $j \in I$ depends on a parameter called *configuration tolerance*, denoted as $\tau_j \in [-1,1]$. The tolerance parameter is chosen to be issue-specific, in order to better differentiate the amount of concessions between issues. Therefore, for each issue $j \in I$, the configuration tolerance depends on the preference difference coefficient of that issue, according to the following formula: $\tau_j = \tau_{gen} * (1 + \delta_j)$

Here the parameter τ_{gen} represents the general tolerance, used by the agent for all issues j. The general tolerance is always chosen between 0 and 0.5 and also gives a measure of how fast the agent is willing to make concessions. Values closer to 0 will denote an agent who is less willing to make concessions, while values closer to 0.5 will denote an agent who is interested to reach a deal quickly. Since $\delta_j \in [-1,1]$ the tolerance for any issue j is scaled between 0 and $2*\tau_{gen}$.

Finally, the target evaluation for each issue j is computed. This is done by taking into account both the basic target issue evaluation (as described above) and a concession to the issue evaluation from the previous bid of negotiation partner, as follows: $TE_j = (1 - \tau_j)\, BTE_j + \tau_j\, E_{BO,j}$

Here BTE_j is the basic issue evaluation for issue j and $E_{BO,j}$ is the evaluation for issue j from the opponent's previous bid. From the above formula, one can see that values of the configuration tolerance τ_j close to 0 signify that mostly the user's own importance factors are taken into account, while values close to 1 shows that maximum possible concession is made towards the other's value. And since τ_j depends directly on δ_j, it is the difference in preference for each issue that determines how much concession should be made.

Within the *Target Evaluation Planning* component we have assumed that the opponent's preference weights for an issue are known. However, if the other is not willing to share his weights for some (or all) issues, then they will need to be estimated.

6.3.2 Estimation of Opponent's Parameters and Guessing

The role of these components is to determine, for those issues for which the opponent was not willing to reveal his preference weights, an estimation of those weights (namely of the parameter $W_{other,j}$ needed in the equation above).

We denote by I_{known} the set of issues for which the opponent was willing to reveal his importance weights in the beginning of the negotiation and by $I_{unknown}$ the issues whose preference weights are kept private. Since all preference weights are normalised (see Section 6.2), the sum of weights for the private issues is computed as: $\Sigma_{j\in I_{unknown}} W_j = 1 - \Sigma_{k\in I_{known}} W_k$.

For the issues with private weights, the remaining weight $\Sigma_{k\in I_{known}} W_k$ needs to be divided between them. This is the goal of the guessing mechanism, which is more formally defined in [14]. Intuitively the idea is to divide the remaining weight based on the perceived concessions the opponent makes during the negotiation. Each discrete-valued issue is assigned a qualitative value from the set {good, fairly good, standard, meager, none}. Each party in the negotiation assigns to these values a numerical evaluation from 0 to 100, independent from the issue weight. For example, if "good" has for the buyer an evaluation of 100, and "standard" the evaluation 60, and the weight of the CD player issue is 60 out of a total of 300 (or normalised form 0.2), then a good CD player will bring an evaluation of 0.2*100=20 to the total utility, while a standard one an evaluation of 0.2*60=12. So a buyer that concedes during the negotiation, for the issue CD player, from "good" to "standard" makes a global utility concession of 8.

However, the Seller, in a closed negotiation setting does not know either the weight of the issue CD player, nor does he know the evaluations the Buyer has assigned to values such as "good", "fairly good" or "standard". However, he can compare concessions between issues, based on the fact that, for a certain price level, buyers always prefer better quality issues than worse ones.

For example, suppose two issues: CD player and Tow Hook have both unknown (i.e. not revealed) utility weights. Initially the Buyer asks for both issues the quality "good". However, for one issue, say CD player, she is only willing to concede from "good" to "fairly good", while for the other, say Tow Hook, she is willing to concede from "good" to "standard". The seller in this example can infer that the CD player is more important to this particular buyer than the Tow Hook, because a rational buyer gives in first in the issues which are less important to her.

A formal description of this mechanism, as well as the experimental results from the tests performed to validate our model are given here for reasons of space, but the interested reader is asked to consult [14].

6.4 An example negotiation trace

The model presented above was tested along several dimensions, such as:

- The number of issue weights revealed
- Whether guessing is used or not
- The choice for the issue importance factors
- The evaluations for the issue value levels

Due to space limitation, we do not give a full discussion of these results here. For a full discussion of our experimental results, we ask the reader to consult [14]. In this section we illustrate the functioning of our model through a complete example trace, as produced by our simulation tool. Tables 1 and 2 below describe the evaluations which are given by the 2 parties to their agents, though the input interface.

	Tow hook	Airco	Extra speakers	CD player	Price
BUYER	90	90	15	15	300
SELLER	15	15	90	90	300

Table 1. Importance factors assigned to different issues by the Buyer and Seller

	Good	F. good	Standard	Meager	None
BUYER	100	85	70	50	0
SELLER	30	65	80	65	100

Table 2. Evaluation given by each side for each qualitative value of the 4 issues: Tow hook, CD player, Extra speakers and Airco

As can be seen from Table 2, we assume a business model in which the Seller prefers to sell the car for a standard price – and not have to install extra accessories, but he is willing to do so in order to sell it.

Table 3 provides the complete trace of this negotiation from the perspective of the Buyer, while Table 4 does the same from that of the Seller. The vertical columns show the bids made by the two parties in successive rounds.

BUYER	1	2	3	4	5	Closing
bids						
price	18000	17450	17968	18047	18083	18083
tow hook	good	fairly good	fairly good	fairly good	fairly good	fairly good
airco	good	standard	standard	standard	standard	standard
speakers	good	meager	none	none	none	none
CD player	good	meager	none	none	none	none
utilities						
own bid	1	0.9203	0.9130	0.9094	0.9068	0.9068
seller's bid	0.7407	0.8782	0.8830	0.8864	0.8889	0.8889

Table 3. The negotiation trace: BUYER's perspective

Figure 7 provides a visualization of the negotiation progress in the joint utility space (as automatically produced by the implementation in our software environment). For clarity, only the first 3 bids of the Buyer and the first 2 of the Seller are shown. The rest lie in the straight line between these 2 points. An interesting effect is that, in this example, after establishing mutually agreeable values for the discrete-value issues (accessories), the agents seem to "walk" the Pareto-efficient frontier towards each other's bid. This corresponds to the haggling about the price from rounds 3-5 in Tables 3 and 4.

SELLER	round 1	2	3	4	5	accept:5
bids						
price	16900	18468	18404	18359	18325	18083
tow hook	none	fairly good	fairly good	fairly good	fairly good	fairly good
airco	none	standard	standard	standard	standard	standard
speak-ers	none	none	none	none	none	none
CD player	none	none	none	none	none	none
utilities						
own bid	1	0.9378	0.9296	0.9238	0.9195	0.8884
buyer's bid	0.3167	0.5932	0.8737	0.8838	0.8884	0.8884

Table 4. The negotiation trace: SELLER's perspective

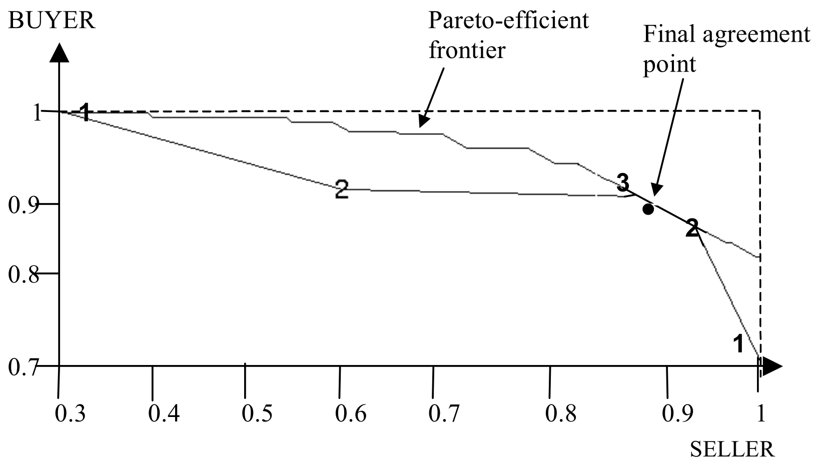


Figure 7. Utility space corresponding to the example trace from Tables 3/4

7 Related Work

This section discusses the literature on the analysis of negotiation processes. Moreover, it reviews automated negotiation systems that use incomplete information described in the literature and compares them to our own.

In the literature on negotiation a number of systems are described. Sometimes it is stated what properties these systems have, sometimes not. If properties are mentioned they can be of different types, and also the justifications of them can be of different degree or type. This section discusses the literature on properties of negotiation and analytical results of implemented systems and of human case studies.

Faratin, Sierra, and Jennings [2] concentrate on many parties, many-issues, single-encounter closed negotiations with an environment of limited resources (time among them). Agents negotiating using the model are guaranteed to converge on a solution in a number of situations. The authors do not compare the solutions found to fair solutions (Nash Equilibrium, Maximal Social Welfare, Maximal Equitability), nor whether the solutions are Pareto Efficient.

Klein, Faratin, Sayama, and Bar-Yam [6] developed a mediator-based negotiation system to show that conceding early (by both parties) often is the key to achieving good solutions. Hyder, Prietula, and Weingart [3] showed that substantiation (providing rationale for your position to persuade the other person to change their mind) interferes with the discovery of optimal agreements.

Weingart et al. [13] found that the Pareto efficiency of agreements between naïve negotiators could be significantly improved by simply providing negotiators with descriptions of both integrative and distributive tactics. Although Pareto *efficiency* was positively influenced by the tactics, Pareto *optimality* was only minimally affected. Compared to [8, 11, 12], the properties identified in this paper are geared towards the analysis of the dynamics of the negotiation process, whereas theirs are more oriented towards the negotiation outcome, rationality and use of resources.

In [19], a model for bilateral multi-issue negotiation is presented, where issues are negotiated sequentially. The issue studied is the optimal agenda for such a negotiation under both incomplete information and time constraints. However a central mediator is used and the issues all have continuous values. The effect of time on the negotiation equilibrium is the main feature studied, from both a game-theoretic and empirical perspective. In earlier research [20] a slightly different model is proposed, but the focus of the research is still on time constraints and the effect of deadlines on the agents' strategies. This contrast with our model, where efficiency of the outcome and not time is the main issue studied. This is because we found that, due to our cooperative assumption, a deal is usually reached in maximum 10-15 steps, if the negotiation speed and tolerance parameters are suitably calibrated.

A direction of work directly related to our guessing heuristic (introduced in Section 6) is represented by [17] and [18]. Like [17] we start from the perspective of distributed negotiation, which eliminates the need of a central planner. As in [17], we also take the heuristic approach and we model agents that are able to jointly explore the space of possible outcomes with a limited (incomplete) information assumption. In [17], this is done through a trade-off mechanism, in which the agent selects the value of its next offer based on a similarity degree with previous bids of the opponent. In our design, we do no explicitly model trade-offs, yet the same effect is achieved through the asymmetric concessions mechanism. An advantage of our model over [17] is that we allow agents to take into account not only their own weights, but also those of the opponent in order to compute the next bid. In this way agents may exchange partial preference information for those issues for which their owners feel this does not violate their privacy. Also the initial domain information for the issues with discrete ("qualitative") values is different. In [17], this consists of fuzzy values, while in our model it is a partial ordering of issue weights.

8 Conclusions and Future Work

The contribution of this work consists of a more systematic approach to the analysis of the negotiation process. Different types of properties are identified and for each class a number of properties are defined. The System for Analysis of Multi-Issue Negotiation (SAMIN) is presented and applied in two ways: to analyse human negotiation in a case study and to analyse the effectiveness of guessing and limited information exchange as implemented in a number of software agents. SAMIN consists of three components: an Acquisition Component to acquire the input necessary for analysis, an Analysis Component to perform the actual analysis, and a Presentation Component to presents the results of the analysis in a user-friendly format.

The system 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.

The strategies tested using software agents showed that the original agents for closed multi-issue negotiation (used in the ABMP system, see [4]), when playing against each other, score better than the human subjects (in human-human negotiations). The software agents were subsequently augmented with a guessing strategy and with the ability to share a bit of information regarding their issue weights. The results show that both strategies increase the effectiveness of the negotiation.

Currently, SAMIN is being used to analyse the dynamics of humans negotiating against software agents of the ABMP system (with and without the guessing strategy, also in setting in which limited preference information is shared). Future research is to analyse the dynamics of other types of (e.g., more experienced) human negotiators. Furthermore, the system needs to allow heterogeneous agents, so that a competition of negotiating agents can be set up and the results of that competition formally analysed. In future, SAMIN will be extended with training facilities for human negotiators, allowing to test the effectiveness of training methods for negotiation. As a simple extension, for example, if a dynamic property checked in a trace turns out to fail, a more detailed analysis can be given of the part(s) of the formula that cause(s) the failure. Finally, we plan to extend SAMIN to provide feedback to a negotiator who is in the middle of a negotiation process, where SAMIN only has access to the same information as the negotiator.

References

[1] Bosse, T., Jonker, C. and Treur, J., Formalisation of Dynamic Properties of Multi-Issue Negotiations. Vrije Universiteit Amsterdam, Department of Artificial Intelligence. Technical Report, 2004.

[2] Faratin, P., Sierra, C., and Jennings, N.R., Negotiation decision functions for autonomous agents. In: International Journal of Robotics and Autonomous Systems, vol. 24(3-4), 1998, pp. 159 – 182.

[3] Hyder, E.B., Prietula, M.J., and Weingart, L.R., Getting to Best: Efficiency versus Optimality in Negotiation, In: Cognitive Science, vol. 24 (2), 2000, pp. 169 – 204.

[4] Jonker, C.M., Treur, J., An Agent Architecture for Multi-Attribute Negotiation. In: B. Nebel (ed.), *Proceedings of the 17th International Joint Conference on AI, IJCAI'01*, 2001, pp. 1195 - 1201.

[5] Jonker, C.M., and Treur, J., Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92.

[6] Klein, M., Faratin, P., Sayama, H., and Bar-Yam, Y., (2001), Negotiating Complex Contracts. Paper 125 of the Center for eBusines@MIT. http://ebusiness.mit.edu.

[7] Kowalczyk, R, Bui, V., On Constraint-Based Reasoning in e-Negotiation Agents. In: Dignum, F, and Cortés, U., (eds.), *Agent-Mediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce Systems*, Lecture Notes in Computer Science, vol. 2003, Springer – Verlag, pp. 31-46.

[8] Lomuscio, A.R., Wooldridge, M., and Jennings. N.R., (2000), A classification scheme for negotiation in electronic commerce, In: *International Journal of Group Decision and Negotiation*, vol. 12(1), January 2003.

[9] Pruitt, D.G., *Negotiation Behavior*, Academic Press, 1981.

[10] Raiffa, H., *Lectures on Negotiation Analysis*, PON Books, Program on Negotiation at Harvard Law School, 513 Pound Hall, Harvard Law School, Cambridge, Mass. 02138, 1996.

[11] Rosenschein, J.S., and Zlotkin, G., *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. The MIT Press, Cambridge, MA, 1994.

[12] Sandholm, T., Distributed rational decision making, In: Weiss, G., *Multi-agent Systems: A Modern Introduction to Distributed Artificial Intelligence*, MIT Press, 1999, pp. 201-258.

[13] Weingart, L.R., Hyder, E.H., and Prietula, M.J., Knowledge matters: The effect of tactical descriptions on negotiation behavior and outcome. In: *Journal of Applied Psychology*, vol. 78, 1996, pp. 504-517.

[14] Jonker, C., Robu, V. – "Automated multi-attribute negotiation with efficient use of incomplete preference information", *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, New York, July 2004.

[15] Brazier, F.M.T., Jonker, C.M., and Treur, J. "Compositional Design and Reuse of a Generic Agent Model", *Applied Artificial Intelligence Journal*, vol. 14, 2000, pp. 491-538.

[16] Byde, A., Kay-Yut Chen – "AutONA: A System for Automated Multiple 1-1 Negotiation", *Fourth ACM Conference on Electronic Commerce*, pp. 198-199.

[17] Faratin, P., Sierra, C. and Jennings, N. Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. *Journal of Artificial Intelligence* vol. 142 (2), 2003, pp. 205-237.

[18] Faratin, P., Sierra, C. and Jennings, N. "Using Similarity Criteria to Make Negotiation Trade-Offs", *Proceedings of ICMAS-2000*, Boston, MA., 119-126.

[19] Fatima, S. S., Wooldridge, M. and Jennings, N. R.. "Optimal Agendas for Multi-Issue Negotiation". In *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, July 2003, pp. 129-136.

[20] Fatima, S., Wooldridge, M. and Jennings, N. R.. "Optimal Negotiation Strategies for Agents with Incomplete Information" . In *Intelligent Agents VIII,* Springer-Verlag LNAI, vol. 2333, pp 377-392, March 2002

[21] Raiffa, H. – "The art and science of negotiation", *Harvard University Press*, Cambridge, Mass., 1982.

Tibor Bosse
Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
e-mail: tbosse@cs.vu.nl

Catholijn M. Jonker
Radboud Universiteit Nijmegen, Nijmegen Institute for Cognition and Information
Montessorilaan 3
6525 HR Nijmegen
The Netherlands
e-mail: C.Jonker@nici.ru.nl

Lourens van der Meij
Vrije Universiteit Amsterdam
The Netherlands
e-mail: lourens@cs.vu.nl

Valentin Robu
CWI, National Center for Mathematics and Computer Science
Kruislaan 403
1098 SJ Amsterdam
The Netherlands
e-mail: robu@cwi.nl

Jan Treur
Vrije Universiteit Amsterdam
The Netherlands
e-mail: treur@cs.vu.nl

Software is available on the Internet as
(X) Demo/trial version

Internet address for download:
http://www.few.vu.nl/~wai/samin
Contact point for questions about the software:
http://www.few.vu.nl/~wai/samin