Experiments in Human Multi-Issue Negotiation: Analysis and Support*

Tibor Bosse[1], Catholijn M. Jonker[1] and Jan Treur[1,2]

[1] Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, jonker, treur}@few.vu.nl ; http://www.few.vu.nl/~{tbosse, jonker, treur}

[2] Utrecht University, Department of Philosophy
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

Abstract

The purpose of this paper is to report on experiments in (human) multi-issue negotiation and their analysis, and to present a generic software environment supporting such an analysis. First, the paper presents a System for Analysis of Multi-Issue Negotiation (SAMIN). SAMIN is designed to analyse negotiation processes between human negotiators, between human and software agents, and between software agents. 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. Second, the paper presents the results 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.

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

* In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'04. IEEE Computer Society Press, 2004.

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, 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. More information about (partially) open negotiations can be found, e.g., in [7] and [10]. 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., [6,10]. 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 than fair.

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, see Section 9. Based on that 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 formally analyse such negotiation processes, i.e., multi-issue, closed, one-to-one negotiations without mediators. Basically, the system needs three different inputs:

(1) a negotiation *trace* (or a set of traces)

(2) a set of *dynamic properties* that are 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 can or cannot 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. 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.

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. Section 4 provides example dynamic properties relevant for closed multi-issue one-to-one negotiations. SAMIN's architecture is presented in Section 5, and some details of the current prototype are presented in Section 6. The set-up of some experiments in human multi-issue negotiation is described in Section 7, and the results are provided in Section 8. Section 9 compares our work with the literature. Finally, Section 10 provides conclusions and some planned future work.

2. Formalising Negotiation Process Dynamics

Negotiation is essentially a dynamic process. To analyse those dymamics, 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 > with:

- S1 state of agent A
- S2 state of agent B

Each of these states include, for example,

- 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 v. Based on this ontology the set of ground atoms At(Ont) can be defined. A state is formalised as any truth assignment: At(Ont) $\longrightarrow$ {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 → 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 → S1 or S2 → 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 → 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 → 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. Specifying Dynamic Properties of a Negotiation Process

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 Language TTL for Dynamic Properties

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). 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:
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, ∃t2: 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.

4. Dynamic Properties of Closed Multi-Issue One-to-One Negotiation Processes

The properties relevant for analysing the dynamics of 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. 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 could be 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, ∀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 tb and te in an appropriate way it 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].

5. Design of the SAMIN architecture

SAMIN is a software environment that has been designed at the Vrije Universiteit for the analysis of multi-issue negotiation processes. This section describes the role SAMIN can take in an analysis setting of negotiation processes, and presents the global design of the architecture chosen for SAMIN. In Section 6 the parts of this design that have been implemented in the current SAMIN prototype are described in more detail.

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 for cognitive scientific purposes, to train human negotiators, or 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.

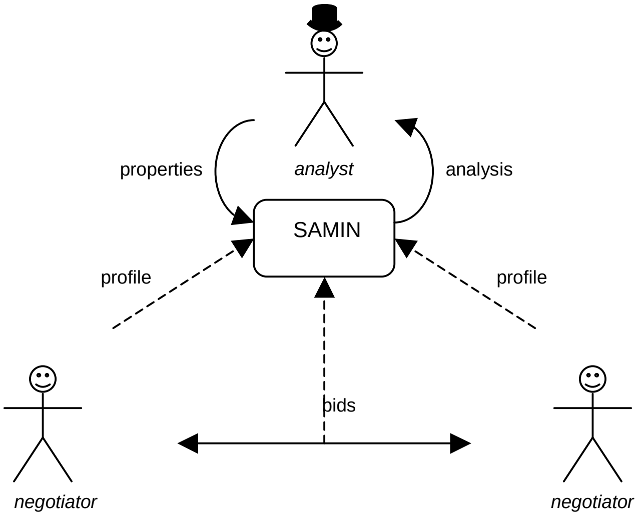


Figure 1. SAMIN in its environment

The input required by SAMIN, see Figure 1, consists of properties, profiles, and traces of bids. Its output consists of an analysis. 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 (see also Section 6.1). 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 "overhear" 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 a bid ontology and the profiles is part of the pre-negotiation phase [10].

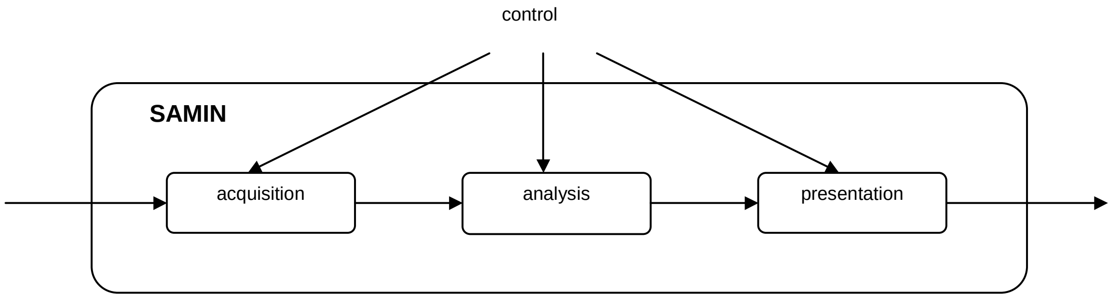


Figure 2. Global Design of the SAMIN architecture

For a global overview of the design of the SAMIN architecture, see Figure 2. It consists of components to acquire the input necessary for analysis, to perform the analysis, and to present the results of the analysis. Furthermore, SAMIN maintains a library of properties, templates of properties, bid ontologies, and profile ontologies.

6. The SAMIN Prototype

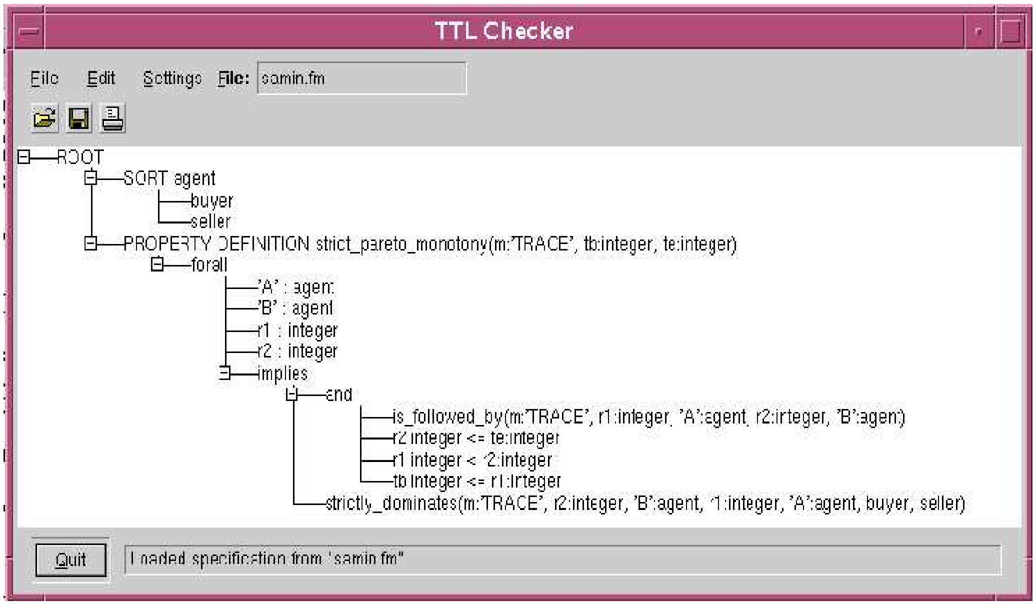
Within the current SAMIN prototype a number of the components of the SAMIN architecture have been implemented. These implemented components will be briefly described below.

6.1. The acquisition component

The acquisition component is used to obtain the required input to perform 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 (that makes use of the bid ontology) the user identifies the possible evaluations that can be given to issue-value combinations, the possible interdependencies between issues, and the utility functions of bids.

The dynamic property editor based on TTL supports the gradual formalisation of dynamic properties that are initially entered in natural language (informal). It is also possible to directly enter formal properties, see Figure 3.



```
ROOT
    SORT agent
        buyer
        seller
    PROPERTY DEFINITION strict_pareto_monotony(m:'TRACE', tb:integer, te:integer)
        forall
            'A' : agent
            'B' : agent
            r1 : integer
            r2 : integer
            implies
                and
                    is_followed_by(m:'TRACE', r1:integer, 'A':agent, r2:integer, 'B':agent)
                    r2:integer <= te:integer
                    r1:integer < r2:integer
                    tb:integer <= r1:integer
                strictly_dominates(m:'TRACE', r2:integer, 'B':agent, r1:integer, 'A':agent, buyer, seller)
```

Loaded specification from 'samin.fm'

Figure 3. Dynamic Property Editor

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 system. 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.

6.2. The analysis component

The analysis component currently consists of a *logical analyser* that is capable of checking whether a dynamic property holds for a trace, or for a number of traces. If a dynamic property does not hold in a trace, then the software reports the places in the trace where the property failed.

6.3. 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 4.

In this Figure, 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 Figure 4, the sequences of actual bids made by both buyer (left) and seller (right) are indicated by the two dark angular lines. 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.

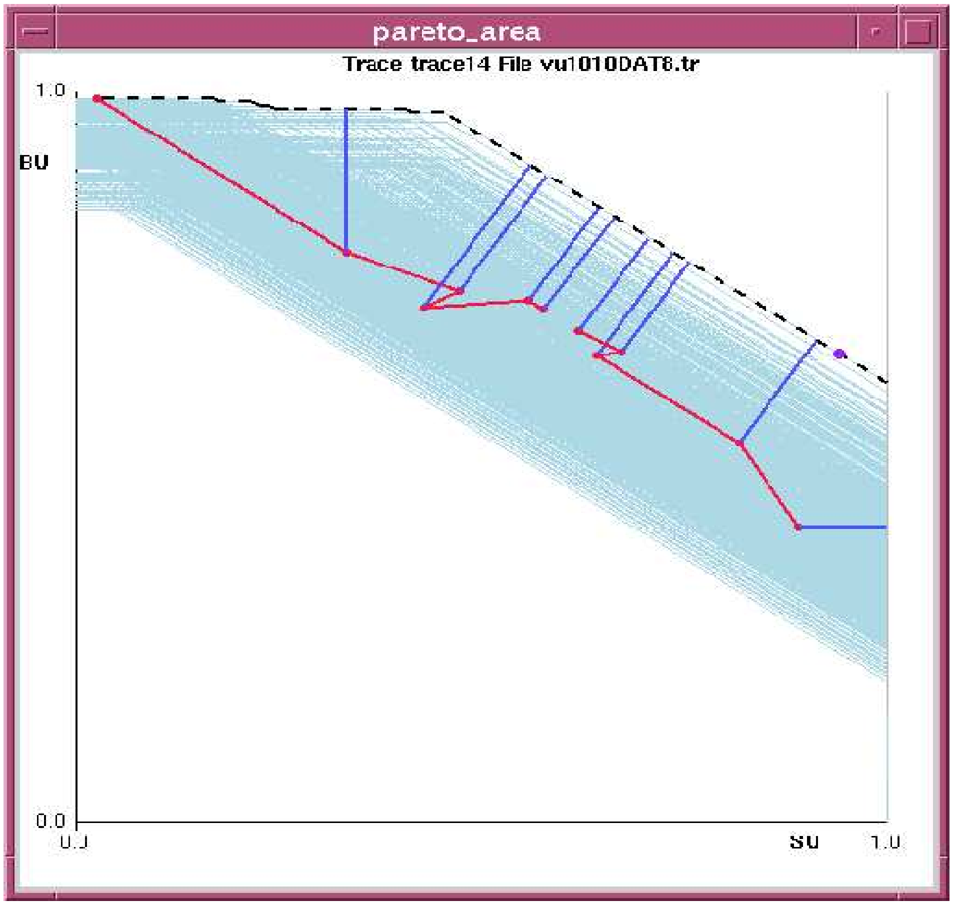


Figure 4. Visualisation Tool

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 over time. Eventually, they reach a point that does not lie on the Pareto Efficient Frontier, but is rather close to it anyhow.

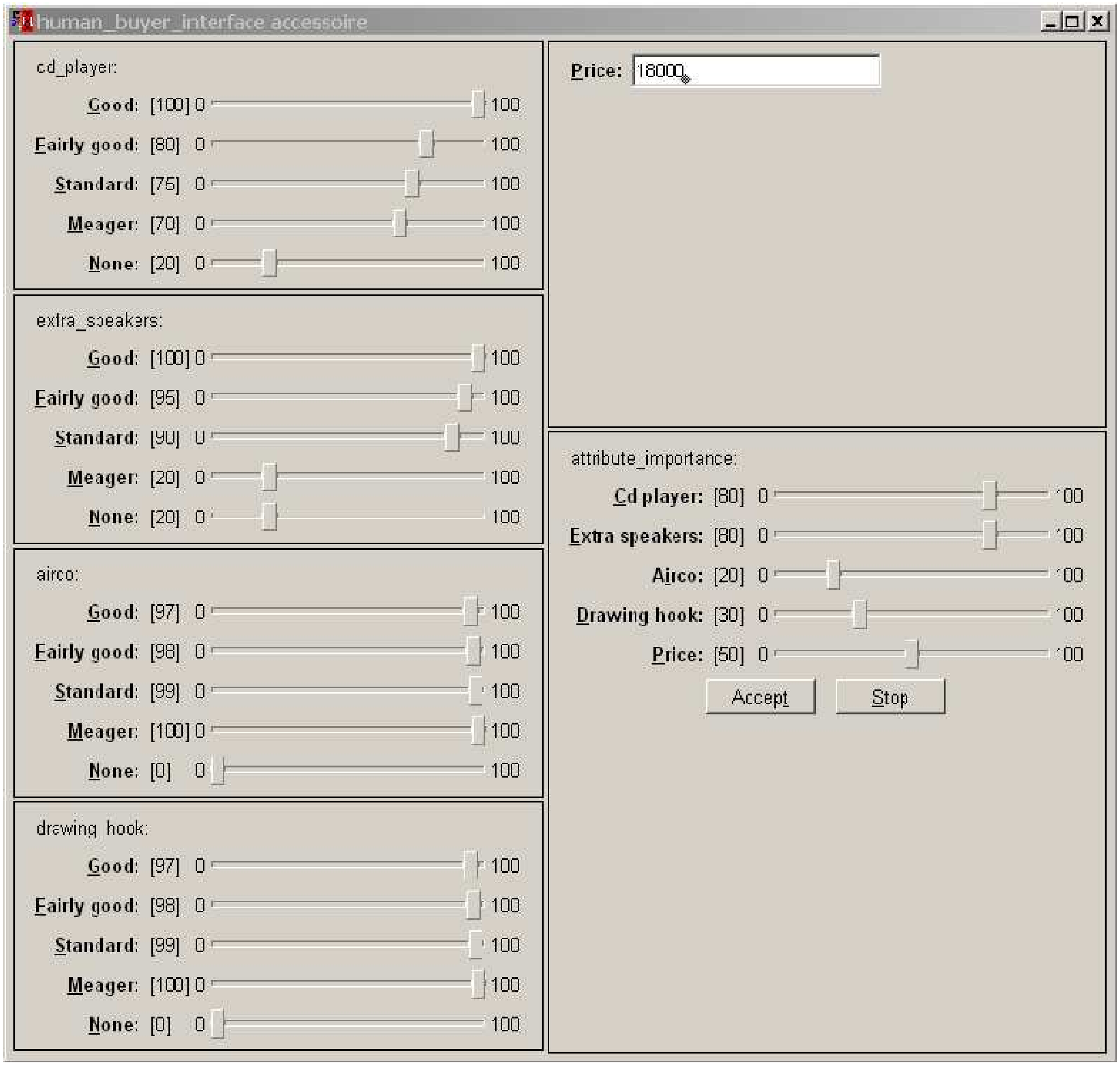
7. Design of the Human Multi-Issue Negotiation Experiments

To illustrate the use of analysing multi-issue negotiation processes, SAMIN has been applied in a case study. As mentioned in Section 6.2, 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 4 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, will be presented in detail in Section 7.1. Section 7.2 describes the setup of the experiments performed in the case study. The results of the analysis of the acquired traces will be shown in Section 8.

7.1. Domain: second hand cars

The object of negotiation is a particular second hand car. In multi-issue negotiation, a bid has the form of values assigned to a number of issues of the object under negotiation. Within this domain, the relevant issues are cd_player, extra_speakers, airco, drawing_hook and price. Consequently, a bid consists of an indication of which CD player is meant, which extra speakers, airco and drawing 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 5. In addition to this negotiation profile, the seller is also provided with a *financial profile*. This is a list of all issues, in which 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 negotiator. However, SAMIN has access to both profiles.



human_buyer_interface accessoire

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

drawing 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 '00
Extra speakers: [80] 0 '00
Airco: [20] 0 '00
Drawing hook: [30] 0 '00
Price: [50] 0 '00

Figure 5. Example Buyer's Negotiation Profile

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:

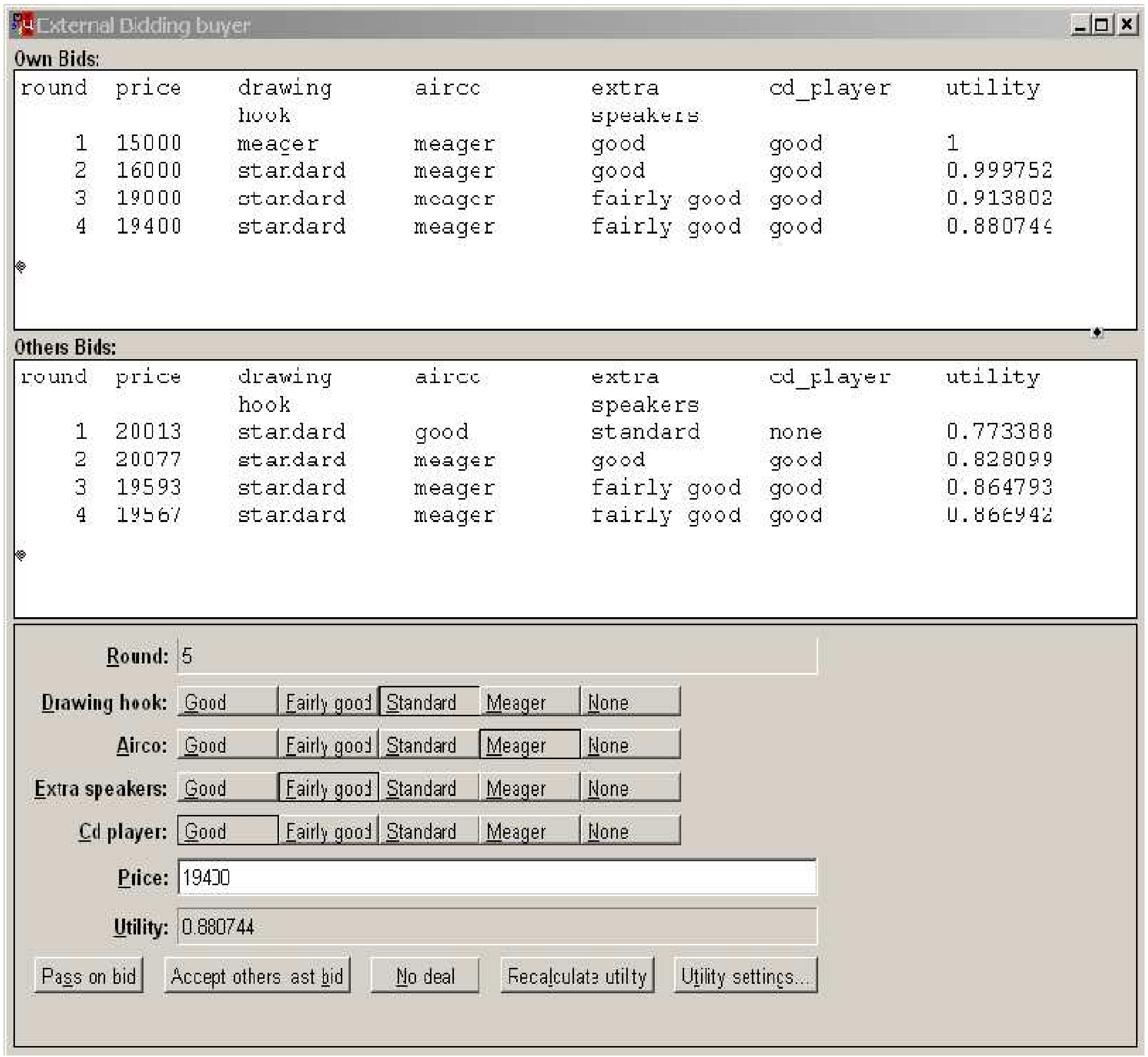
$$U_B = \Sigma_j w_j \ E_{B,j}$$

The weight factors w_j are based on the attribute 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.

Besides for facilitating the bidding process, the profiles are used by SAMIN to analyse the resulting traces (see Section 8). 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.

7.2. Experimental setup

Participants. Seventy-four 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 about 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 session, in March 2003, 28 students participated. In the third session, in November 2003, 16 students participated.



Own Bids:

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

Others Bids:

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

Round: 5

Drawing hook: Good | Fairly good | [✓ Standard] | Meager | None

Airco: Good | Fairly good | Standard | [✓ Meager] | None

Extra speakers: Good | [✓ Fairly good] | Standard | Meager | None

Cd player: [✓ Good] | Fairly good | Standard | Meager | None

Price: 19430

Utility: 0.880744

Figure 6. 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 6. 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.

8. Results of the Human Experiments

Using the SAMIN prototype, a number of relevant dynamic properties for multi-issue negotiation (also see Section 4) have been checked against the traces that resulted from the experiments. In this Section, a selection of interesting results is reported:

Obviously, the property bid_alternation (Section 3) holds for all traces. This means that all participants have committed to the protocol, which prescribes that as long as the negotiation lasts, a bid from A to B should be followed by a bid from B to A.

In none of the traces, the Pareto_inefficiency (Section 4) of the resulting deal was equal to 0. In several cases, during the negotiation some bids made by one of both parties temporarily lay on the Pareto Efficient Frontier, but the resulting bids never did. On average, the negotiating agents performed only slightly above halfway, i.e., the resulting bids lay somewhat above the middle of the space of possible bids (the light area of Figure 4). Apparently, it is difficult for human negotiators to guess the Pareto Inefficiency. As a result, they find it hard to decide what is the right moment to accept a proposal.

As can be derived from the previous conclusion, also the Nash Point was never reached in any final agreement, nor was it reached during any of the negotiations.

When used as a trace property, the property strict_Pareto_monotony did not hold in any of the traces. When used as a limited interval property, it sometimes held during a very short interval, but hardly ever during more than three steps. Apparently, the profiles of the negotiating parties were often strongly opposed, meaning that a gain for one party implies a loss for the other. However, when changing the criterion of strict domination into weak domination, the property often held for larger intervals. Most of the time, these intervals corresponded to the “end phase” of the negotiation: the phase in which the only issue on which no agreement has yet been reached, is the price.

9. Discussion

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. Examples of such literature are the following.

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.

A previous version of SAMIN was first developed as an analysis environment for the multi-issue negotiation system ABMP, see [4]. However, the scope of SAMIN as it has been set up as a generic environment is much broader. SAMIN can be seen as a logical next step, given the existing negotiation-related systems and the existing literature on the formalisation and analysis of negotiation, to provide a bridge between such negotiation systems and the analysis of their properties.

10. Conclusions and Future work

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.

Currently, SAMIN is being used to analyse the dynamics of humans negotiating against software agents of the ABMP system. Future research is to analyse the dynamics of other types of (e.g., more experienced) human negotiators and of automated negotiation systems and 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 (see Section 6.2 above), 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., (ed.), *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.