Demonstration of a Software System for Automated Multi-Attribute Negotiation

Catholijn Jonker[1] Lourens van der Meij[1] Valentin Robu [2] Jan Treur[1]

[1]Vrije Universiteit Amsterdam,
Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam,
The Netherlands {jonker, treur}@cs.vu.nl

[2]CWI, National Center for Mathematics and
Computer Science, Applied Algorithmics Group
Kruislaan 403, 1098 SJ Amsterdam
robu@cwi.nl

Abstract

This paper presents the demonstration of a software system for integrative negotiation. The agents in this system conduct one-to-one negotiations, in which the values across multiple attributes 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). 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

Negotiations have been identified as a key form of interaction in multi-agent systems and, therefore, have attracted considerable attention from the multi-agent research community. Especially multi-attribute negotiations ([4]) are of particular interest, since in such cases outcomes that bring utility gains for both parties are possible. The reported work aims at bridging the gap between negotiation theory, human negotiation practice, and the implementation of egotiation mechanisms in open systems and e-commerce settings, by constructing answers to open questions (e.g., how to deal with incomplete information, how to integrate human negotiation with that mediated by software agents).

The theoretical foundations of the research, the agent models proposed, empirical studies, as well as the relations with other work in this field have been presented in [1], [3], and [4]. In this paper an overview of the technical content of the software demonstration and present a high-level view of its main elements are shown. Section 2 describes the negotiation model, while Section 3 summarizes the aims of the proposed demonstration.

2. The Negotiation Model

The considered type of negotiation follows an alternating-offers protocol. A bid has the form of values assigned to a number of attributes. If the negotiation is about a car, for example, the relevant attributes considered are CD player, Extra Speakers, Airco, Tow Hedge, Price, and a bid consists of an indication of which CD player is meant, which extra speakers, airco and tow hedge, and what the price of the offer is.

The proposed demonstration is based on this domain, and was originally developed in collaboration with Dutch Telecom KPN. However, the negotiation model presented in [3] and [4] is a generic one. 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). In both cases, the DESIRE agent system design method and software environment (cf. [2]) were used to design and (automatically) implement the agents. The system supports 3 types of negotiation:

- *Human vs. human negotiation*. In this case a human Buyer and Seller negotiate (possibly through a computer network) by manually specifying, at each step, their bids.
- *Human vs. software agent*. In this case one of the parties (either the Buyer or the Seller) negotiates by specifying the bids herself, while the other is represented by a software agent. In this case, the human only specifies his preference weights and value levels for the attributes, and the software agent conducts the bidding on his/her behalf.
- *Software agent vs. software agent*. This is the fully automated case, in which both parties are represented by software agents and the humans only specify their initial preference parameters through a graphical interface.

The attributes (issues) considered are of two types: discrete-valued and continuous-valued attributes. In the example domain the *discrete-valued attributes* are the accessories that a car dealer can install to meet the preferences of his customer: Tow Hedge (or Drawing Hook), Air Conditioning, CD Player and Extra Speakers. Each of these can take 5 discrete value levels: good, fairly good, standard, meager and none. These values are assigned an evaluation by each party in the beginning of the negotiation, in the form of a value from 0 to 100. For example, the Buyer may assign, for the attribute CD player, the evaluations: good = 100, standard = 70, none = 20, etc.

In the current model for the example domain there is only one *continuous-valued attribute*, the price. The evaluation for price is computed through an evaluation function, which can have different, parametrised types (e.g., linear uphill or downhill, normal distribution function).

The evaluation of the whole bid is computed as a weighed sum of the evaluations of individual attributes. The weights (which are also specified by each party through an interface) play a crucial role in the negotiation, because the way each party makes concessions in successive bids depends on his/her preference weights (possibly also involving the known, or estimated, preference weights of the opponent). The full mathematical model for computing the target evaluations and determining the actual values of the attributes in the next bid is too complex to be presented in a demonstration paper. Basically the model involves planning the target utility first at the overall level and then for each attribute separately. Finally, the configuration of the next bid will be selected such that it fits these target values (for the full details, interested readers are referred to [3] and [4]). The demonstration also shows how the model incorporates the possibility that the agents use partial preference information in order to achieve better outcomes. Two types of preference information can be taken into account:

- 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 both cases, the profile information is in the form of preference weights. Next section, highlights the features of the software environment to be demonstrated.

3. Purpose of the Demonstration

This section describes more explicitly the aims for the proposed demonstration and how it is planned to achieve them. Otherwise stated, it is discussed which features would be the most interesting to show to a larger audience and their importance is discussed.

The first such feature (introduced in Section 2) is the way incomplete preference information and opponent modeling can be used to increase the efficiency of the joint exploration of the utility space. The method used to achieve this is to compare the traces produced by two negotiations: a perfectly closed negotiation with no guessing and one where some profile info (in the form of one or several preference weights) and/or guessing are used (see [6]).

It is important to note that these negotiations should start from exactly the same configurations of attribute weights and evaluation levels for the values. Otherwise it becomes difficult to say whether any positive effect observed is due to the choice of these parameters or due to the guessing mechanism, for example.

The second important aim of the demo is to show how humans can use such a system to negotiate both against other humans or software agents. This is significant for several reasons:

- It gives us the possibility to use an experimental economics approach to analyse the behaviour of humans in complex negotiations over multiple attributes and in the presence of uncertain information. This may hold important clues for the design of future automated trading mechanisms.
- It provides an alternative to game theory in testing the “incentive-compatibility” properties of the system (i.e. if the system encourages truthful revelation of preferences and if there are any ways in which the strategy used by the software agent can be abused by a human negotiator).

In [1] such empirical work is reported, where the system has been used as a set up for human negotiation experiments and the analysis of the outcomes of these experiments.

Finally, the system can also be used as a training tool for introducing human negotiators into the complexities of multi-attribute utility theory (described in the classical work by H. Raiffa - [5]). In this educational capacity, our software may be useful both to students, as well as professionals outside the academic field.

4. References

[1] Bosse, T., Jonker, C.M., and Treur, J., Experiments in Human Multi-Issue Negotiation: Analysis and Support. Technical Report, Vrije Universiteit Amsterdam, Department of Artificial Intelligence, 2004.

[2] Brazier, F.M.T., Jonker, C.M., and Treur, J., Principles of Component-Based Design of Intelligent Agents. *Data and Knowledge Engineering*, vol. 41, 2002, pp. 1-28.

[3] 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.

[4] Jonker, C., Robu, V. Automated Multi-Attribute Negotiation with Efficient Use of Incomplete Preference Information. Technical Report, Vrije Universiteit Amsterdam, Department of Artificial Intelligence, 2004.

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

[6] http://homepages.cwi.nl/~robu/aamas/aamas_demo.html