Specification of strategies for negotiating agents

René Schumann, Zijad Kurtanovic, and Ingo J. Timm

Information Systems and Simulation,
Goethe University Frankfurt am Main
Robert-Mayer-Str. 10, 60325 Frankfurt am Main, Germany
`[reschu|zijad|timm]@informatik.uni-frankfurt.de`

Abstract. Negotiations are an important part of today's inter-enterprise business processes. Interoperability in negotiations has addressed more technical aspects, so far. Common protocols and shared ontologies can provide sufficient solutions. Actually the interoperability on the process level is more challenging, that is how the multiple attributes of a negotiation issue can be assigned simultaneously respecting all constraints between the attributes in a way that the outcome is acceptable for both human negotiators.

In this paper we present a negotiation model that allows agents to negotiate on behalves of human negotiators. Therefore we formalize negotiation strategies and use a meta model based on ECORE as a framework for specifying negotiation strategies. To allow efficient negotiations among agents, we present a technique for efficient representation and reasoning about these negotiation strategies.

Key words: automated negotiation, tradeoff strategies, MDA

1 Introduction

Negotiations are an important part of today's inter-enterprise business processes. According to Pruitt [1] a negotiation is a process, by which two or more parties try to reach a mutually acceptable agreement on some matter. So far these negotiations are hard to automate. Nevertheless, the field of automated negotiations is an important field in the multiagent system (MAS) research [2, 3].

Thus designing systems that implement business process including negotiations between two or more companies is a hard task. This interoperability problem is neither on the technical level. Common communication protocols and shared ontologies can be applied and thus lead to an interoperability on the system level. Interoperability becomes critical on the process level. Within a negotiation a set of parameters, like price or quality, has to be determined simultaneously. This is complicated by the fact that commonly an agent does not know how the different parameters depend on each other for its opponent and what parameter configurations are acceptable for him.

Within a negotiation encounter there may exist more than one deal satisfying both parties [4, 5]. Thus when the negotiation runs in a conflict because of the parties' distinct interests and preferences, interaction can only proceed by making new proposals with the aim of coming to a mutually acceptable agreement [5]. Those new proposals can be found via tradeoffs, which are an important aspect of negotiations in the human behavior [6, 4] and have been adopted for software agents [7, 5, 8] as well. A tradeoff between two negotiation attributes is a combination of attributes values. The main idea of a tradeoff thereby is to improve one attribute while worsening of other attributes in return [8].

In this paper we encourage the notion of agency that agents act on behalf of their owners, thus the agent should follow the negotiation strategies specified by its owner, the human negotiator. Thus the negotiation may have not an optimal outcome from a game theory point of view, but the agents behave in concern with its owner. The problem of *acquiring* such knowledge from the human negotiator is often left open [3, 8]. But this information is an essential requirement making use of automated negotiation in practical settings. Within a negotiation strategy it is specified how the agent can generate alternative offers based on tradeoffs, and how the agent behaves within a negotiation. This allows automated negotiations within inter-enterprise business processes. Here we present a specification language that allows the human negotiator to specify the negotiation strategy in a comprehensive way, without prerequisite that he has knowledge of the techniques used to implement its agent. Because typically the human negotiator is not a programmer, capable to develop it's agent by himself. The specification language is based on the ECORE meta model which is part of the Eclipse Modeling Framework (EMF) [9]. This allows an easier usage of model driven architecture (MDA) concepts to automatically transform the specified negotiation strategy in software code that can be used by the software agent. Thus the owner of the agent can be ensured, that the agent will act, as he has specified, without a potential erroneous and expensive process of transforming the specification into the agents code by hand.

The rest of this article is structured as follows. In the next section we present the underlying negotiation process. We explain the entire protocol and outline, what role the negotiation strategies have. In section 3 we refine the concept of negotiation strategies by giving a formal definition and show how those strategies can be defined using ECORE . We show that a negotiating agent using tradeoffs has to solve a prioritized fuzzy constraint satisfaction problem (PFCSP) to generate suitable offers or counter offers. As the PFCSP is in general too complex to be solved during a negotiation process that proceed under real time conditions we demonstrate in section 4 how the negotiation process can be implemented efficiently. The entire automated negotiation process is demonstrated in section 5. Finally we draw our conclusion and outline future work.

2 Negotiation model

In this work we use a negotiation model presented by Lou et al. [5]. It is a simple negotiation setting with a bilateral negotiation. Two roles are defined: a buyer and a seller agent. Both agents negotiate a contract with a number of attributes, like price, quality, delivery or payment date. Each of the agents has a global preference function that orders all permutations of all possible outcomes of the negotiations. The agents operate in a semi-competitive environment. This is reflected by their behavior strategies which are based on the *principled negotiation approach* [6]. That is, that they try to weaken their position only minimally e.g. by minimal information disclosure, minimal relaxing their desires [5].

In the following we will outline the negotiation protocol and the negotiation strategies relevant aspects.

2.1 Negotiation protocol and agent's behaviors

The negotiation protocol is based on the alternating offers protocol [10]. Seller agent's behavior protocol is presented in figure 1. The states represent the allowed performatives he can send in a given negotiation context, except the dashed one - this is interpreted as the initial state, in which the seller agent is ready to take proposals from buyer agents. The edges represent the performatives of the buyer agent that can be received during a negotiation encounter. During the negotiation the buyer can specify a set of constraints that each potential offer has to met.

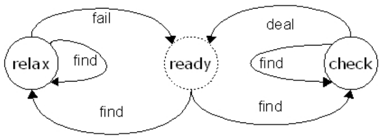


Fig. 1. Seller agent's behavior protocol

When the performative *find* is received the negotiation is initiated and the agent can answer with performatives *check* or *relax*. He uses the performative *check* when he finds an offer satisfying the currently published constraints of the buyer. If there exist no such offer, the seller ask to *relax* at least one of the constraints, so that he can find a suitable offer.

The buyer agent's behavior protocol is presented in figure 2. The states and edges are defined analogous to figure 1. The buyer agent is the initiator of a negotiation and he does so by sending a *find*-performative to the seller agent. While doing so he publishes the constraint with the highest priority to the seller[1]. After he sends a *find*-performative he expects a *check* or *relax* performative from the seller agent. If a *check*-performative is received, he checks the proposed offer and answers with *deal* if it is acceptable, i.e. all constraints are met and the acceptance threshold is exceeded. Otherwise he criticizes the offer by publishing the constraint that is violated. If multiple constraints are violated the one with the highest priority is chosen.

---

[1] If there are different constraints with equal priority one is randomly chosen.

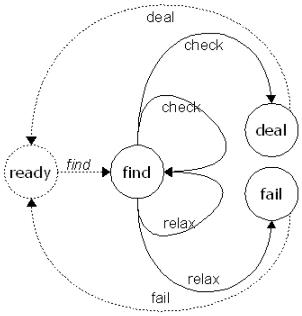


Fig. 2. Buyer agent's behavior protocol

2.2 Negotiation strategy

The decision made in the agents behavior specification, like to decide which counteroffer should be send, or if an offer should be accepted base on data given by the user. These data is encoded in the negotiation strategy. Thus the complexity of the negotiation strategies is independent from the complexity o f the negotiation protocol or the agents behavior.

Of course not every deal is acceptable for the agents. Each agent has a minimal acceptance threshold that must be exceeded. This guarantees that e.g. the seller agent does not agree to a deal with a negative profit. The seller agent has a list of products. For each product a set of value vectors of negotiable attributes is associated. The set of resulting possible deals can be computed by:

- a tradeoff strategy, given by the user,
- the global preference function, and
- user's threshold that specifies what preference value is needed at least.

A tradeoff strategy specifies what combination of attributes form an acceptable deal for the user. It specifies a set of tradeoffs and preferences over attribute values and attributes value combinations. A tradeoff is a relation between two attributes of the negotiations. It defines that in favor for worsening one attribute the other has to improve at a certain rate.

The buyer agent has a requirement model for each product he is interested in, which expresses his specifications about the desired products. This is done in form of fuzzy prioritized constrains. Fuzzy constraints reflect a natural way of modeling user's requirements [5]. Often different priorities can be found in real world settings, thus these fuzzy constraints are prioritized [11]. These requirements are derived from a tradeoff strategy given by the human negotiator, the global preference function and user's thresholds, as well.

So far it becomes clear that the specification of tradeoff strategies is an essential part automating negotiations as outlined in this article. Therefore we detail tradeoff strategies in the next section.

3 Tradeoff strategies

Within a tradeoff strategy all information about tradeoffs for a negotiation are encoded. As sketched previously the tradeoff strategy is a set of pairs[2] of attributes which are in a tradeoff relation and a set of *independent* attributes. A tradeoff function defines how much one attribute can be worsened, in favor for improving the other. According to [8] this can be formalized as follows:

Definition 1
Let the value set of the attribute x be defined with $X = [l_x, r_x]$, and let the value set of y be $Y = [l_y, r_y]$. Then the function is called tradeoff function between X and Y if it is continuous, monotonic *and met the* boundary condition. *The boundary condition assures, that if one attribute is assigned to the best value, the other attribute has to be made worse [8].*
The pair (x, y) is called a tradeoff pair.

For each tradeoff pair a preference function is defined, which specifies the preference over the tradeoff alternatives. Tradeoff alternatives are value combinations of the relevant tradeoff pair [8]. Independent attributes are not in a tradeoff relation with other attributes. To each of them a preference function is associated. A tradeoff strategy represents a *directed forest*, this is formalized in definition 2.

Definition 2
Let the negotiation attributes as nodes and the tradeoff pairs as directed edges be given. The direction of an edge is defined by the tradeoff function (see definition 1). If this function has the form $\tau : X \to Y$, there exists an edge from node X to Y. Then a tradeoff strategy *represents a* directed forest*:*

- *Let a be a negotiation attribute. All values of a's value set A are in a preference ordering relation $\preceq \subseteq A \times A$: the preference direction is* descending *if smaller values are allowed,* ascending *otherwise.*
- *To each tradeoff pair (a, b) the following is associated, with A and B as the value sets of a and b respectively:*
  - *A tradeoff function $\tau : A \to B$ in terms of definition 1.*
  - *A tradeoff preference function $p : A \times B \to [0, 1]$, which assigns to each tradeoff alternative a preference value. It reflects a trapezoid formula of three segments (analogue to the preference function in [8]) to describe the increasing, steady and decreasing preference over tradeoff alternatives.*
- *Independent negotiation attributes are trees with only one node. For each such negotiation attribute a a preference function is associated $p : A \to [0, 1]$, with A: $\forall a, b \in A : \; a \preceq b \Leftrightarrow p(a) \leq p(b)$.*

An example of a tradeoff strategy is shown in figure 3. Hereby the value sets are marked in the lower level of the nodes and user's *best* tradeoff alternatives are labeled at the edges. The edge going from the trapezoid to the first node is labeled with the priority of the tree to which it connects. A more complex

---

[2] Experience shows that tradeoffs between a pair of attributes are the most common [4], hence we focus only on them.

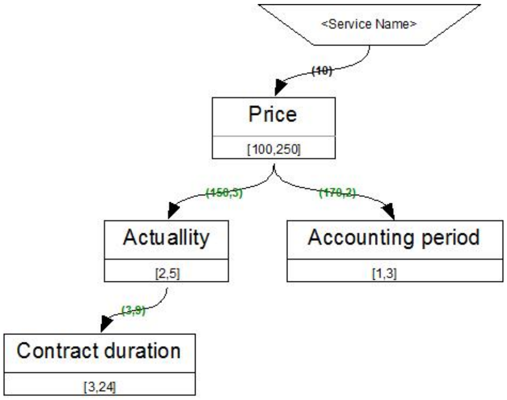


Fig. 3. Exemplified tradeoff strategy

tradeoff strategy can have more complex tradeoff nets, which is sketched in figure 4. In fact it is ensured that all tradeoff strategies can be represented as a forest. This ensures some formal but also informal benefits. A tradeoff strategy can be visualized in a *clear* and *accustomed* way to the users. Due to the acyclic structure there cannot exist inconsistencies, which may be introduced by cycles. This reduces the complexity specifying and validating those strategies. Moreover this forest structure allows the use of efficient algorithms for reasoning about the tradeoff strategies [12].

As already mentioned, a tradeoff strategy can be seen as a set of constraints. In the negotiation scenario it's often the case, that the constraints are *fuzzy* and have different level of importance [13, 5]. In this regard the tradeoff strategy can be modeled as a *prioritised fuzzy CSP* [11] in the following way:

- There is a set of variables with an associated value set; to each negotiable attribute (i.e. variable) a value set is associated.
- From each tree a fuzzy constraint is derived; for negotiation attributes connected as a tree - one fuzzy constraint.

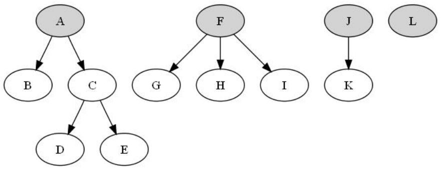


Fig. 4. Tradeoff strategy as directed forest: nodes represent negotiation attributes, edges tradeoff relations. *L* is an independent attribute

- A priority function $\rho : T \rightarrow [0, +\infty)$ assigns to each tree (i.e. fuzzy constraint) a level of importance.

To gain the tradeoff strategies from the user we have developed a metamodel for negotiation strategies, and thus tradeoff strategies in particular. Therefore we used the meta-metamodel Ecore of the Eclipse Modeling Framework (EMF) [9]. Basing on EMF allows us have the ability to integrate a graphical editor for tradeoff strategies in the near future and currently use the concepts of model driven development for a more rapid development. Using EMF the metamodel can be transformed to code. Prior to this a generator-model is needed - a platform-specific model which supplies the EMF-generator with information like the connections between multiple Ecore-models, the name of the generated files, referenced Ecore-models etc. [14]. With use of the generated editor, a user can model his *negostrategy*-based negotiation strategies visually, validate it against the metamodel and save it as XMI-documents. These documents can be transformed into data-objects representing the negotiation strategy used by an agent specifying its negotiation strategy.

4 Efficient reasoning for negotiations

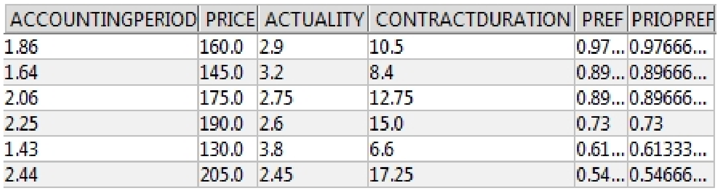
As outlined in the last section a tradeoff strategy forms a PFCSP. Problems of this form are, as all CSPs, hard to solve [15]. To ensure an efficient negotiation process a suitable representation and reasoning technique has to be identified. The main idea to cope with this complexity is to compute a representation set of acceptable deals prior to the negotiation. This representation set can be encoded within a relational representation, as a set of tables. In fact it is necessary for every tree of the tradeoff strategy (see e.g. figure 4) to generate one table. The remaining information not kept in the tables like e.g. the preference directions of attributes' value sets needed for generation of critics are provided through the code model of the *negostrategy*-metamodel. Thus during the negotiation process reasoning about offers and the generation of critics can be done querying the representation set, e.g. using SQL statements.

The computation of the set of acceptable deals is not done at compile time, because context aspects, like the fact with whom a negotiation is done, can be respected in the negotiation strategy. If these aspects have to be integrated in the relational representation the resulting tables would be significantly larger, as all possible contexts would have been respected.

The relational representation is a set of tables where each table represents a fuzzy-constraint, that is equivalent to a set of tradeoffs with different preferences over some negotiable attributes. By joining the tables according to the tree structure the valid combinations of attributes' values can be derived and ordered according to the preference function of the agent. Each row in a table represents an assignment over all negotiation attributes in the tradeoff tree with respect to the approximated tradeoff functions which is called a *tradeoffconsistent* assignment. For each cell in a column (i.e. negotiation attribute), a subset of the value set is derived on the basis of the preference direction of the negotiation attribute and the value contained in that cell. Thus to each row i.e. *tradeoff-consistent* assignment, the agent can derive combinations of interval sets, which represent a spectrum of possible tradeoffs over negotiable attributes represented by that table. This allows the agents to search more efficiently for a mutually acceptable solution i.e. over a set of potential solutions, rather than over single point solutions each round. For each tree the procedure is as follows:

- First a representation set is generated from the root negotiation attribute, containing a list of values from its value set.
- along the directed edges i.e. tradeoff relations the representation lists of all remaining negotiation attributes are derived through the approximated tradeoff functions.
- the preference of each row is computed, with the global preference function.

An example is shown in figure 5, this corresponds to the set of acceptable of the negotiation strategy specified shown in figure 3. In this table all possible deals that satisfy all tradeoff conditions are shown ordered by their preference value.



ACCOUNTINGPERIOD	PRICE	ACTUALITY	CONTRACTDURATION	PREF	PRIOPREF
1.86	160.0	2.9	10.5	0.97...	0.97666...
1.64	145.0	3.2	8.4	0.89...	0.89666...
2.06	175.0	2.75	12.75	0.89...	0.89666...
2.25	190.0	2.6	15.0	0.73	0.73
1.43	130.0	3.8	6.6	0.61...	0.61333...
2.44	205.0	2.45	17.25	0.54...	0.54666...

Fig. 5. Table of all possible attribute combinations of the attributes price, actuality and contract duration

5 Automating Negotiations a case study

In this section we demonstrate the specification of a negotiation. We have designed a simple scenario where the seller offers access to an information service, that the buyer wants to subscribe. Attributes of the contract are

- price (PR)
- actuality of the data (AC)
- contract duration (CD)
- accounting period (AP)

From the seller's perspective these attributes can have the following values: The price can be in a range of [120,270] €, of course a higher price is preferred. The delivered data can have an actuality of 1,2,4 or 6 hours. As more accurate data is more expensive, older data is preferred. The seller assumes its optimal ration between profit and accuracy gaining €170,- for two hour old data. Possible contract durations are 6,12,18 or 24 month, longer durations are preferred. Accounting periods can have a length of 1,3,4 or 6 month, shorter periods are preferred, not given a credit to the customer.

From the buyer's perspective the attributes have other desired values and preferences, of course. The price should be in the interval between [100,200] €, and of course a lower price is preferred. Actuality of the data should be between two and five hours, more accurate data is preferred and a higher price is acceptable. A fair ratio between accuracy and price for the buyer is paying €150,- for three hours old data. The contract duration can be in an interval between [3,24] month, where a shorter duration is preferred being more flexible. For a better (for the buyer a lower) price the buyer might be willing to accept longer contract durations. Acceptable accounting periods can be one to three month. Longer periods are preferred, but for a better price shorter ones can be accepted.

The resulting tradeoff strategies can be specified as described in section 3. In fact the tradeoff strategy of the buyer was shown in figure 3. The tradeoff strategy of the seller is shown here in figure 6. Using the negotiation metamodel we have generated an EMF-editor that allows specifying these negotiation strategies in a comprehensive way. After the negotiation strategies have been specified they can be saved persistently using the XMI format.

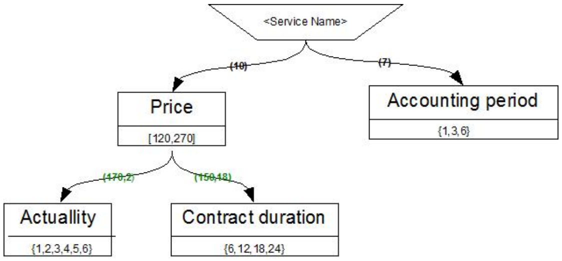
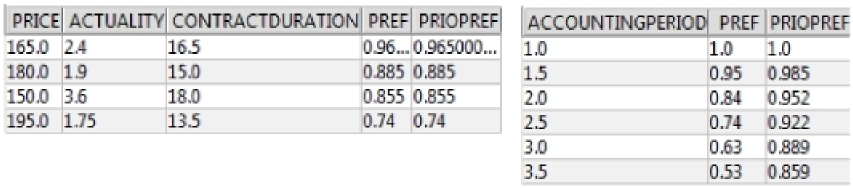


Fig. 6. Seller tradeoff strategy

Before the agents start a negotiation they have to build a relational representation of the set of acceptable deals induced by the tradeoff strategies. This set defined by the buyer's tradeoff strategy has already been presented in figure 5[3]. The resulting tables for the seller are shown in figure 7. If the two agents try to negotiate about subscription terms for the information service the resulting negotiation trace is shown in table 1. The buyer starts the negotiation by selecting the first top row of the table 5 which contains the most preferred combination of attributes' values according to his tradeoff strategy. According to his behavior

---

[3] The directed forest representing buyer's tradeoff strategy contains only *one* tree which corresponds to *one* table



PRICE	ACTUALITY	CONTRACTDURATION	PREF	PRIOPREF
165.0	2.4	16.5	0.96...	0.965000...
180.0	1.9	15.0	0.885	0.885
150.0	3.6	18.0	0.855	0.855
195.0	1.75	13.5	0.74	0.74

ACCOUNTINGPERIOD	PREF	PRIOPREF
1.0	1.0	1.0
1.5	0.95	0.985
2.0	0.84	0.952
2.5	0.74	0.922
3.0	0.63	0.889
3.5	0.53	0.859

Fig. 7. Relational representation of seller's tradeoff strategy. His thresholds are already considered, thus only acceptable tradeoffs are shown

strategy he always tries to minimize the revelation of private information, thus revealing only one constraint to the seller i.e. he requests a deal for a price $\leq$ €160,-. The seller uses the SQL-select command and finds an offer that satisfies buyer's price constraint. He sends these appropriate contract conditions to the buyer and asks him to evaluate his offer. In round 2 the buyer evaluates that the offer is not acceptable because of some violated conditions e.g. for the offered price a better actuality of data and shorter contract duration is expected. In consequence he asks the seller to find another offer satisfying the price and the additional violated actuality constraint. As the seller agent has no fitting offer[4] he asks to relax these constraints. In doing so the buyer lowers his expected satisfaction degree with the deal. Finally, after 2 more unacceptable offers from the seller in rounds 4 and 6, a deal is reached in round 8.
If, for example, the buyer had more strict thresholds which limit his tradeoff ability, then the negotiation could fail.

6 Conclusion

In this article we present a negotiation model that allows complex negotiations among two agents acting on behalves of the human negotiators. This capability allows automating negotiations as part of complex inter-enterprise business processes, thus enabling interoperability on the process level. In doing so, it was pointed out that the formal specification and modeling of negotiation strategies is important. In particular tradeoff strategies are of relevance.

The main contribution of our work is that we have defined a formal representation of negotiation strategies, for agents acting on behalve of human negotiators. Thereby we use the ECORE meta model, to allow the negotiators themselves to describe the negotiation strategy. The transformation into executable code that can be used by the agent is done via automatic transformation gaining the advantages of a formal definition and the ideas of MDA. Thus the expensive and possibly erroneous process of encoding these strategies by hand is avoided.

Due to the lack of space we have omitted more details of the meta-modeling language for negotiations that we have specified. This meta-model includes a

---

[4] In this case the SQL-select statement using buyer's conditions applied on his relational representation shown in figure 7 returns an empty result set.

Round 1	Buyer	Performative: Find Constraint: PR $\leq$ 160
	Seller	Performative: Check (PR:150,AC:4,CD:18,AP:1)
Round 2	Buyer	Performative: Find Constraint: PR $\leq$ 160 $\wedge$ AC $\leq$3
	Seller	Performative: Relax
Round 3	Buyer	Performative: Find Constraint: PR $\leq$ 145 $\wedge$ AC $\leq$3
	Seller	Performative: Relax
Round 4	Buyer	Performative: Find Constraint: PR $\leq$ 175 $\wedge$ AC $\leq$3
	Seller	Performative: Check (PR:165,AC:2,CD:18,AP:1)
Round 5	Buyer	Performative: Find Constraint: PR $\leq$ 175 $\wedge$ AC $\leq$3 $\wedge$ CD $\leq$ 13
	Seller	Performative: Relax
Round 6	Buyer	Performative: Find Constraint: PR $\leq$ 190 $\wedge$ AC $\leq$3 $\wedge$ CD $\leq$ 15
	Seller	Performative: Check (PR:180,AC:2,CD:12,AP:1)
Round 7	Buyer	Performative: Find Constraint: PR $\leq$ 190 $\wedge$ AC $\leq$3 $\wedge$ CD $\leq$ 15 $\wedge$ AP $\geq$ 2
	Seller	Performative: Check (PR:180,AC:2,CD:12,AP:3)
Round 8	Buyer	Performative: Deal

Table 1. Full negotiation trace of buyer and seller (PR: price, AC actuality, CD contract duration, AP accounting period)

graphic specification, too. As we have outlined, the presented specification of tradeoff strategies is very expressive on the one hand, but on the other hand can cause complex computational efforts, as tradeoff strategies can be described as a prioritized fuzzy constraints satisfaction problem. Thus efficiently representations and reasoning is necessary to achieve implementable solutions. We have shown that a relational representation of the set of acceptable deals induced by the negotiations strategy can be computed be used during the negotiation.

An underlying vision of our project is to allow the human negotiator, who is responsible for the negotiations, not for its technical implementation, to specify his negotiation strategy in a form that can be transformed automatically into the reasoning knowledge of the agent. Therefore we will extend our tooling. As the negotiation specification meta model includes a graphical notation, we are going to develop a visual editor for the specification of negotiation strategies. Making it more convenient for the human negotiator. Moreover it is intendet to automate more phases of the specification of multiagent negotiations using MDA principles. So further steps can be the specification and automated transformation of negotiation protocols.

As the presented method follows an engineering approach to specify negotiation strategies it is an open question what expressional power the presented approach for specifying negotiation strategies has, compared to other approaches like bargaining theory.

References

1. Pruitt, D.G.: Negotiation behavior. Organizational and occupational psychology. Acad. Pr., New York, N.Y. (1981)
2. Wooldridge, M., Ciancarini, P.: Agent-oriented software engineering: the state of the art. In: First international workshop, AOSE 2000 on Agent-oriented software engineering, Secaucus, NJ, USA, Springer-Verlag New York, Inc. (2001) 1–28
3. Kraus, S.: Strategic negotiation in multiagent environments. Intelligent robots and autonomous agents. MIT Press, Cambridge, Mass. (2001)
4. Steele, P.T., Beasor, T.: Business negotiation: A practical workbook. Gower, Aldershot (1999)
5. Luo, X., Jennings, N.R., Shadbolt, N., Leung, H., Lee, J.: A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments. Artificial Intelligence Journal **148**(1-2) (2003) 53–102
6. Fisher, R., Ury, W.: Getting to yes: Negotiating agreement without giving in. 2. edn. Mifflin, Boston (1991)
7. Mudgal, C., Vassileva, J.: Bilateral negotiation with incomplete and uncertain information: A decision-theoretic approach using a model of the opponent. In: In Klusch and Kerschberg (Eds.) Cooperative Information Agents IV, LNAI, Springer-Verlag (2000) 107–118
8. Luo, X., Jennings, N.R., Shadbolt, N.: Acquiring user tradeoff strategies and preferences for negotiating agents: A default-then-adjust method. Int. J. Hum.-Comput. Stud. **64**(4) (2006) 304–321
9. Foundation, T.E.: Graphical editing framework (gef). http://www.eclipse.org/modeling/emf, accessable at 08.01.2009
10. Osborne, M.J., Rubinstein, A.: Bargaining and markets. Economic theory, econometrics, and mathematical economics. Acad. Press, San Diego, Calif. (1990)
11. Luo, X., Lee, J., Leung, H., Jennings, N.R.: Prioritised fuzzy constraint satisfaction problems: axioms, instantiation and validation. Int Journal of Fuzzy Sets and Systems **136**(2) (2003) 155–188
12. Russell, S.J., Norvig, P.: Artificial intelligence: A modern approach ; [the intelligent agent book]. 2. ed. edn. Prentice Hall series in artificial intelligence. Prentice Hall, Upper Saddle River, NJ (2003)
13. Dubois, D., Fargier, H., Prade, H.: Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. Applied Intelligence **6** (1996) 287–309
14. Budinsky, F.: Eclipse modeling framework: A developer's guide. The eclipse series. Addison-Wesley, Boston, Mass. (2003)
15. Ghallab, M., Nau, D., Traverso, P.: Automated planning : theory and practice. Elsevier, Kaufmann, Amsterdam et al. (2004)