



## A MODEL OF AUTOMATED NEGOTIATION BASED ON AGENTS PROFILES

SERBAN RADU, EUGENIA KALISZ, ADINA MAGDA FLOREA\*

**Abstract.** In this paper we present a model of self-interested agents acting in an open environment, which captures the most relevant elements of agents' behaviour related to negotiation with other agents. The agent behaviour is mainly motivated by the gain they may obtain while fulfilling their goals and negotiating. Sometimes, the agents are also motivated by the necessity to cooperate with other agents for achieving their goals. The key element in the agent behaviour is their capability to develop a set of negotiation profiles: the preference profile, the partner cooperation profile and the group-of-partners negotiation profile; these profiles help the agents to conduct their negotiation.

**Key words:** agents, multi-agent systems, negotiation, strategy, negotiation profiles

**AMS subject classifications.**

**1. Introduction.** Negotiation is essential in settings where autonomous agents have a desire to cooperate, but also conflicting interests. Automated negotiation among intelligent agents has thus become increasingly important in applications that require computer supported decision making, like e-commerce, distributed resource allocation, or virtual enterprises.

The environments of such applications are inherently open, as they are populated with self-interested agents designed and/or owned by different people and there is no complete information about the preferences or decision-making processes of the participating agents. In order to be really autonomous and achieve performance when conducting a negotiation, an agent should be able to anticipate both the outcome of the negotiation and the best potential partner with which to start a negotiation. Machine learning approaches can contribute to adapt the agent's strategy during negotiation and trading, achieve better outcomes and increased payoffs.

A series of negotiation strategies and studies already exist in the literature, for example [1, 2, 3, 4, 5], in which the agents are able to choose different strategies. The automated negotiation process is usually taking place in open environments. These environments don't have a way to control the agents' behaviour and it is also possible to have humans in these environments, whose behaviour is unpredictable [4]. Some negotiation strategies are based on agent profiles, which can define statically or develop dynamically agents' preferences. There are advantages for creating agents negotiation profiles. Using these profiles, agents obtain better results than those in case of fixed negotiation strategies, i.e. increase the agents' gain from negotiation [6, 7, 8].

In this paper we present a framework for automated negotiation, based on negotiation profiles, and rules that encode the agents' negotiation strategy.

The set of negotiation profiles the agents are able to evolve consists of: the preference profile, which specifies the agent negotiation strategy, the cooperation profile, which keeps track of the agent interaction with the other agents in the system, and the group-of-partners negotiation profile, which clusters the profiles of several agents.

The outcome of negotiation is evaluated for different strategies, encoded in the preference profile. In the system there are different agents and a facilitator, which can be used by the agents for registering their capabilities and optionally for facilitating the negotiation.

The three types of negotiation profiles are described and discussed, as well as the uninformed and informed negotiation strategies. Negotiation strategies are implemented in the form of production rules. In our approach, preference coefficients can be assigned to these rules and dynamically modified, according to the negotiation situation.

The current system is under development and has been tested on some simple cases of electronic transactions. When completed, the system is aimed to support general e-commerce transactions.

The paper is organized as follows. Section 2 presents the agent framework for automated negotiation, Section 3 presents the agent structure, and Section 4 details the preference profile and different agent strategies for negotiation. Section 5 presents related work and Section 6 contains conclusions and several future developments.

**2. Framework for Automated Negotiation.** The negotiation agent behaviour that we propose is defined in a framework of a multi-agent system in an open environment. The system includes the facilitator,

---

\*Department of Computer Science, University Politehnica of Bucharest, Bucharest, Romania ([serban.radu](mailto:serban.radu), [eugenia.kalisz](mailto:eugenia.kalisz), [adina.florea@cs.pub.ro](mailto:adina.florea@cs.pub.ro)).

which can be used or not by the agents during negotiation, depending on the preference profile. Because the environment is open, other agents can enter or exit the system dynamically, cf. Fig. 2.1. When entering the system, the agent has to register with the facilitator, and the facilitator will inform other agents about the new agent arrival.

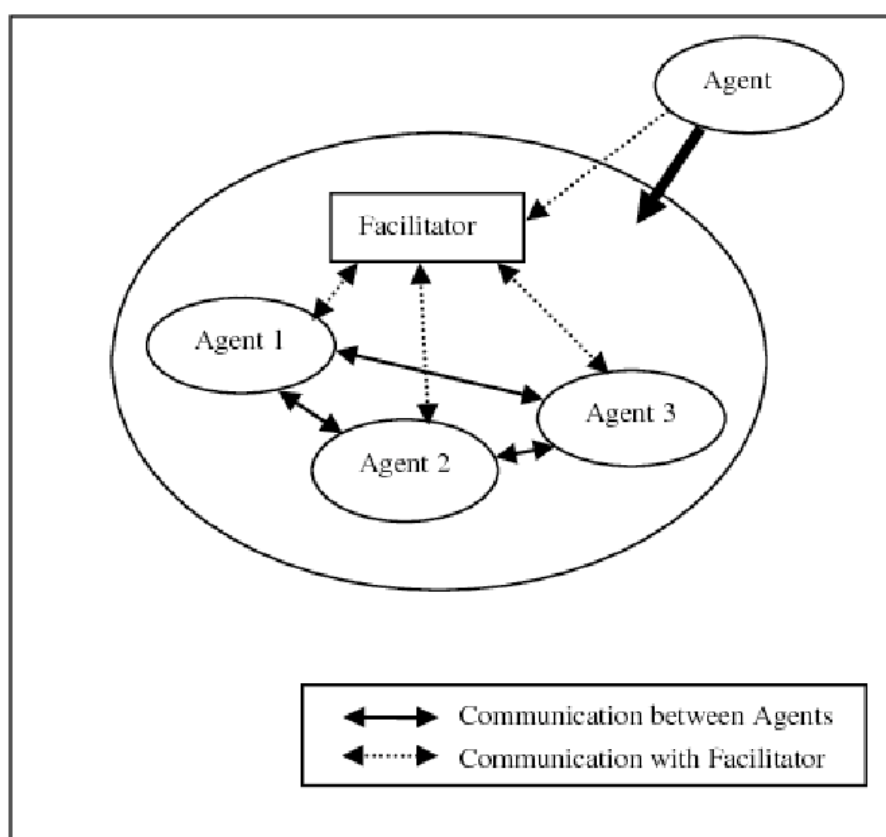


FIGURE 2.1. *The MAS Open Environment.*

When an agent wants to negotiate, it sends a message to the facilitator, asking it to find other agents appropriate to negotiate with. In this message, the agent sends to the facilitator two types of information: a part of the message presenting its abilities, for example which tasks it can do, which objects it possesses, etc., and the other part of the message asking help for a negotiation, e.g., to buy or sell an item. It is the facilitator which will query the agents with the required abilities, and will inform the agent about the outcome. The advantage of this approach is that not all the agents are queried, but the disadvantage is that the facilitator has a lot of messages to process and can become a bottleneck in the system.

The negotiation between agents is a single issue negotiation, where the price of the products is bargained. The type of negotiation implemented is a heuristic negotiation, in which the agent computes the gain, as compared to its private value for an object. The agents negotiate following the Contract Net protocol. Each agent takes into account a private value for any item (product or service) to trade (sell or buy). The Contract Net protocol may be applied conveniently in B2B models, where one company wants to acquire a product or service (and becomes the initiator of the protocol) and also in some specific cases of B2C where the initiator is the seller of a service he wants to be promoted.

There is an upper limit for the number of negotiation rounds. In a buying negotiation, an agent will look for a lower value than its private one, while in a selling negotiation, its main goal is to obtain more than the items private value. The private value is established by the internal reasoning mechanism of the agent and depends on the specific domain.

The agent behaviour is mainly motivated by the gain, but also, depending on a specific context, by the desire to achieve cooperation with other agents. For example, as a general rule, it will not accept a price lower than its private value. However, when the agent wants to cooperate, it can accept a lower price. The agent behaviour is set up by the negotiation strategy.

The negotiation model we propose in this paper can be easily extended to multi-issue negotiation. In this case, the agent will trade the negotiation object (NO). A negotiation object is the range of issues over which agreements must be reached, as defined in [5]. The object of a negotiation may be an action which the negotiator agent A asks another agent B to perform for its benefit, a service that agent A asks to B, or, alternatively, an offer of a service agent A is willing to perform for B, provided B agrees to the conditions of A. An agent A may have a plan to achieve one of its desires, but may be unable to carry out some of the necessary actions, therefore it will ask B to execute these actions for it, if it believes the actions belongs to B's abilities [7]. The agent A may ask agent B to perform a service for him, for example to paint its house, or A may be a communication company offering to an agent B a competitive long distance calls service.

**3. Agent Structure.** The agents have different reasoning capabilities, as specified above, designed to conduct successful negotiation and the fulfillment of agents' goals. Any agent from the system uses a set of behaviour rules, which defines how the agent fulfills the goals and carries on the tasks assigned to it, and a set of rules, which defines its negotiation strategies, cf. Fig. 3.1. During negotiation, the agents gather information about the partner agents, and store it in the associated cooperation profiles.

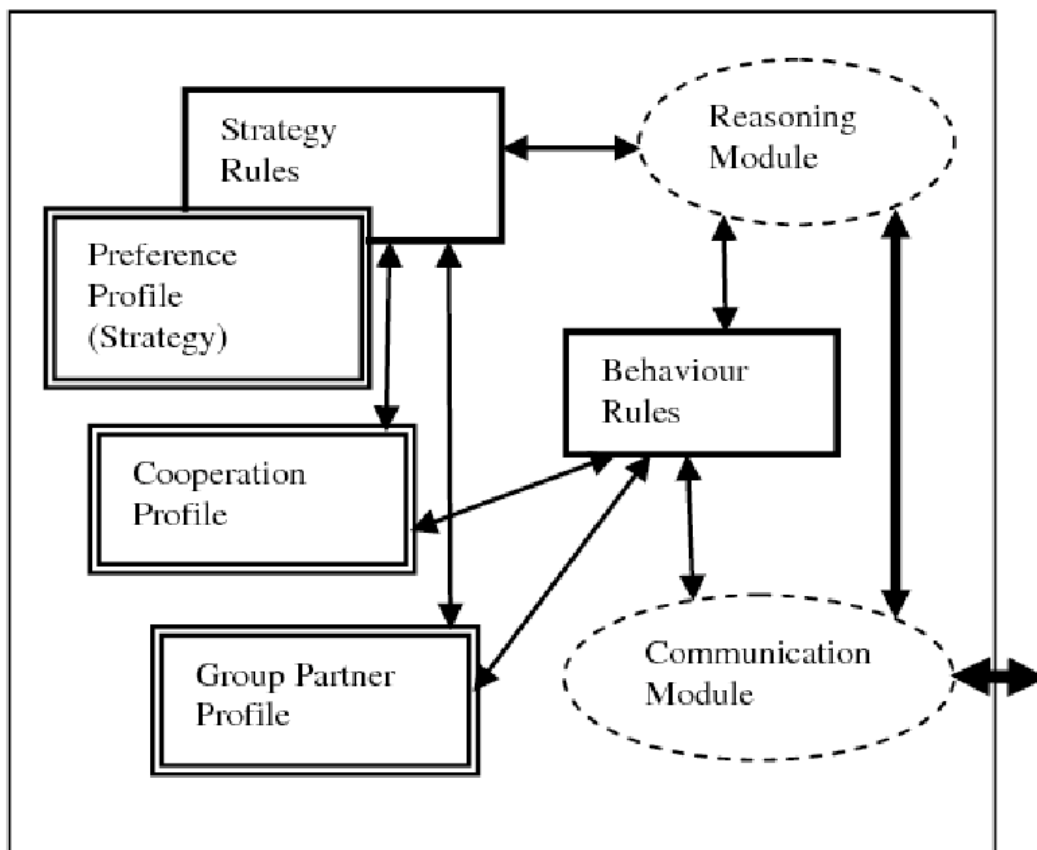


FIGURE 3.1. *The Agent Main Components.*

Each agent is endowed with a set of negotiation profiles, cf. Fig. 3.1:

- **the preference profile**, which specifies the agent negotiation strategy;
- **the partner cooperation profile**, which keeps track of the agent interactions with the other agents in the system;

TABLE 3.1  
*Agent Cooperation Profile.*

Agent Name	No. Negotiations with Partner	No. Successful Negotiations	Gain
John	5	3	2
Gain Percent	No. Negotiation Rounds	Interest Degree of Partner	Classification
40	6	3	cooperative

- **the group-of-partners negotiation profile**, which deals with a group of negotiation partners.

The partner cooperation profile describes the preferences regarding the agents with which an agent prefers to cooperate. The cooperation profile of an agent is implemented as a structure which evolves dynamically. For each agent encountered during the negotiation process, knowledge about the outcome of different negotiations is stored in this structure.

The partner cooperation profile is a characteristic of each agent and is stored as a matrix. Each row contains the agent name and a set of associated attributes, upon which the agent dynamically changes its preferences. The partner cooperation profile is updated during the negotiation process, at the end of each negotiation. Table 3.1 presents an example of partner cooperation profile for an agent.

An entry in the matrix, defining the cooperation profile for an agent, has the fields presented below, representing the cooperation attributes.

1. The first field represents the name of the partner agent, stored as a string of characters.
2. The second field contains how many times the agent negotiated with its partner, and is represented as a natural number.
3. The third field stores the number of successful negotiations, as an integer number.
4. The fourth field points out the connection between the outcome of negotiation and the private value of the agent for the negotiation object, showing the total gain obtained. This is stored as a positive or negative integer number.

Based on heuristic criteria, an agent can accept a price greater than its private value in order to buy a product, or can sell a product at a price lower than its private value. These decisions are based on the information stored in the agent cooperation profile and are specified by the strategy rules.

5. The fifth field shows the gain percentage, namely how much is the gain obtained while negotiating with the partners, as a percentage of the agent total gain.
6. The sixth field is the number of negotiation rounds, during the last negotiation, and is an integer number.
7. The seventh field tries to capture the agents' beliefs about the partner abilities and/or credentials. We call this attribute the interest degree of the partner, which is quantified as: very interesting, interesting, moderately interesting, and not interesting (4...1). For example, if the partner has an ability to perform a task, which is lacking to the agent, then the partner is interesting or very interesting to the agent. Moreover, if the negotiation is successfully concluded in a small number of steps, and the gain is positive, then the partner is very interesting. This field is either set in the beginning by the designer if the partner agent is known, or it is computed based on the agent abilities using a function provided by the designer.
8. The eighth field contains the classification of the partner agent, which represents the current agent belief about the cooperation potential of the partner. The partners are classified in six cooperation classes: highly cooperative, very cooperative, cooperative, slightly cooperative, non cooperative and unknown.

The attributes 1 to 7 described above are updated by the agent after each negotiation with a specific agent. The last attribute (8) will be filled in by a more elaborate process, to be described further on.

The representation of the agents' cooperation profiles as a set of attributes is able to characterize the cooperation potential of a partner, with a high degree of granularity. We would like to be able to characterize the cooperation potential of a partner agent in a broader sense; to this aim, we proposed the clustering of partners into cooperation classes, specified in field 8. We achieve this classification by using a C4.5 learning algorithm [9], in which the fields from 2 to 7 are used as classification attributes and field 8 represents the class. We can have two approaches to generate training examples for the learning algorithm. In the first approach, we let the system run and collect gathered information. In the second approach, we generate a training set of virtual agents, with which an agent has virtual negotiations. After these virtual negotiations are performed, the matrix is filled with the results of these negotiations, according to the fields described before.

The information stored in this way in the matrix will allow the classification of real negotiations. The

classification is improved, as more negotiations take place and the matrix is filled with the correct real values. After that, the C4.5 algorithm will classify correctly the negotiation instances.

Some premises guide the C4.5 algorithm, such as the following:

- if all cases are of the same class, the tree is a leaf and so the leaf is returned labeled with this class;
- for each attribute, compute the potential information provided by a test on the attribute, based on the probabilities of each case having a particular value for the attribute. Also, compute the gain in information that would result from a test on the attribute, based on the probabilities of each case with a particular value for the attribute being of a particular class;
- depending on the current selection criterion, find the best attribute to branch.

**4. Agent Behaviour.** The preference profile of an agent describes its strategy. This is represented as a set of negotiation rules. These rules can be heuristic negotiation rules, which refer to cooperation profiles, or rules which encode certain negotiation strategies.

The negotiation strategy of an agent may show how much and how quickly the prices are decreased and if the price is lower than the agent's private value. An agent can use a tradeoff. In the case of cooperative agents, an agent can gain more once, and then can sell cheaper. This is a kind of global evaluation on previous deals.

The negotiation strategy is implemented in the form of production rules, each agent keeping a history of its interactions. If in a given situation several rules are eligible, then the negotiation strategy decides which rule from the conflicting set to apply. A possible approach to solve the conflicts is to assign priorities between rules. The solution is to apply the rule with the highest priority. Using a feedback, it is possible to apply the same rule or another rule.

The rule's priorities are dynamically modified, and the preference coefficients are not built-in, they are dynamically changed, according to the negotiation situation. The preference coefficients are updated, using the Q-learning algorithm. Each preference coefficient is indexed on a state and an action. The states from the Q-learning algorithm represent the internal states of an agent and the actions are described by the rules applied by an agent when negotiating.

The group of partner negotiation profile is defined by grouping into classes the partner agents, with which the agent interacted in the system. For each of the six values of the cooperation classes, the group negotiation profile contains the list of all agents that belong to a given class. The agents, for which no class was found out yet, belong to the unknown class. During the interaction with other agents, using the C4.5 algorithm, the agents are classified in the appropriate class; moreover, following a negotiation, an agent may migrate from one class into another.

There is a tradeoff between how often we update the agent classes, which may be costly, and the accuracy of classification.

The preference profile is aimed to express the negotiation strategy, more specifically to establish the priority among the negotiation rules.

Each rule has an associated preference coefficient (**PC**), which indicates, in case two or more rules apply in a given situation, which rule should be preferred. In this way, the strategy can be explicitly set up by the system designer by specifying the values of PC when writing the rules, and can be changed from one use of the system to another, simply by changing these coefficients. Another use of these coefficients is that they can be dynamically modified by a reinforcement learning algorithm, in a similar manner to a learning classifier system [10]. Learning classifier systems are a machine learning technique that may be categorized in between symbolic production systems and sub-symbolic connectionist systems.

There are two categories of rules used in the strategy. The first category of rules is used in the beginning of the negotiation and represents **uninformed strategy rules**. The second category of strategy rules is used when there is enough information about the negotiation partner and represents **informed strategy rules**. In this part of the negotiation, it is also possible to apply the uninformed strategy rules. Depending upon the success or failure of the negotiation, the preference coefficients of the strategy rules are changed accordingly.

It is possible to have more than one strategy in the system, for instance, at each negotiation step, the price is decreased by 1, or is decreased by 3. Another strategy rule tells what happens when the price increases with 10 % above the private value, if the offer is instantly accepted or not.

In order to show an example of strategy rules, we consider the following Contract Net protocol:

- a)  $cfp(A, X, NO, P)$  is the communication primitive, which represents a call for proposals from agent  $A$  to all the acquaintances  $X$ , regarding a negotiation object  $NO$ , with an associated cost  $P$

- b) *propose*(*X*, *A*, *NO*, *P*, *Step*) is the communication primitive, which represents the response of agent *X* to the *cfp*, with the negotiation object *NO*, price *P* and negotiation step *Step*
- c) *accept*(*X*, *NO*) indicates the acceptance of a proposal issued by *X*, for the *NO*
- d) *reject*(*X*, *NO*) indicates the rejection of a proposal issued by *X*, for the *NO*
- e) *counterpropose*(*A*, *X*, *NO<sub>i</sub>*, *P<sub>i</sub>*, *Step*) defines a communication primitive, which represents the counterproposal of agent *A* to the proposal of agent *X*, with the negotiation object *NO<sub>i</sub>*, price *P<sub>i</sub>* and negotiation step *Step*.

As in any other Contract Net protocol, agents may be either buyer or sellers, and the roles can be interchanged.

We consider a Prolog-like language and a set of predicates, defined as follows:

- *propose*(*X*, *A*, *NO*, *P*, *Step*) defines a predicate, which is true when agent *A* receives the response (b) of agent *X* to the *cfp*, with the negotiation object *NO*, price *P* and negotiation step *Step*
- *accept*(*X*, *NO*) defines a predicate which, when true, triggers an acceptance message (c) of a proposal issued by *X*, for the *NO*
- *reject*(*X*, *NO*) defines a predicate which, when true, triggers a rejection message of a proposal issued by *X*, for the *NO*
- *counterpropose*(*A*, *X*, *NO<sub>i</sub>*, *P<sub>i</sub>*, *Step*) defines a predicate which, when true, represents the counterproposal (e) of agent *A* to the proposal of agent *X*, with the negotiation object *NO<sub>i</sub>*, price *P<sub>i</sub>* and negotiation step *Step*
- *tp*(*Ag\_Name*, *Atr\_Name*, *Value*) is a predicate which selects from the cooperation profile, for a given agent name (*Ag\_Name*), the value (*Value*) of the attribute (*Atr\_Name*) in the associated field.

Considering the above described predicates, an example of an uninformed strategy rule is:

*propose*(*John*, *Tom*, *House*, *1000*, *1*),  
*private\_value*(*House*)=*p*,  $p \geq 1000 \rightarrow \text{accept}(\text{John}, \text{House})$

and some examples of informed strategy rules of an agent are given below:

*r*<sub>1</sub> :

- *propose*(*John*, *Tom*, *House*, *1000*, *S*),  
*tp*(*John*, *No\_Successful\_Negotiations*, *v*<sub>1</sub>),  
*tp*(*John*, *No\_Negotiations\_with\_Partner*, *v*<sub>2</sub>),  $v_1 > v_2 - 2$ ,  
*tp*(*John*, *Interest\_Degree\_of\_Partner*, *v*<sub>3</sub>),  $v_3 > 3$ ,  
*tp*(*John*, *Gain\_Percent*, *v*<sub>4</sub>),  $v_4 > 20$   
 $\rightarrow \text{accept}(\text{John}, \text{House}) \quad PC_1$

*r*<sub>2</sub> :

- *propose*(*John*, *Tom*, *House*, *1000*, *S*),  
*tp*(*John*, *No\_Successful\_Negotiations*, *v*<sub>1</sub>),  $v_1 > 5$ ,  
*tp*(*John*, *Gain*, *v*<sub>2</sub>),  $v_2 > 100$ ,  
*tp*(*John*, *Interest\_Degree\_of\_Partner*, *v*<sub>3</sub>),  $v_3 = 4$   
 $\rightarrow \text{accept}(\text{John}, \text{House}) \quad PC_2$

*r*<sub>3</sub> :

- *propose*(*John*, *Tom*, *House*, *1000*, *S*),  
*tp*(*John*, *No\_Successful\_Negotiations*, *v*<sub>1</sub>),  
*tp*(*John*, *No\_Negotiations\_with\_Partner*, *v*<sub>2</sub>),  $v_1 < 0.5 * v_2$ ,  
*tp*(*John*, *Interest\_Degree\_of\_Partner*, *v*<sub>3</sub>),  $v_3 < 2$   
 $\rightarrow \text{reject}(\text{John}, \text{House}) \quad PC_3$

*r*<sub>4</sub> :

- *propose*(*John*, *Tom*, *House*, *1000*, *S*),  
*tp*(*John*, *No\_Successful\_Negotiations*, *v*<sub>1</sub>),  
*tp*(*John*, *No\_Negotiations\_with\_Partner*, *v*<sub>2</sub>),  $v_1 < 0.5 * v_2$  ,  
*tp*(*John*, *Interest\_Degree\_of\_Partner*, *v*<sub>3</sub>),  $v_3 > 3$   
 $\rightarrow \text{counterpropose}(\text{Tom}, \text{John}, \text{Car}, 500, S+1) \quad PC_4$

*r*<sub>5</sub> :

- *propose*(*John*, *Tom*, *House*, *1000*, *1*),  
*tp*(*John*, *Classification*, 'highly cooperative')  
 $\rightarrow \text{accept}(\text{John}, \text{House}) \quad PC_5$

*r*<sub>6</sub> :

- *propose(John, Tom, House, 1000, S),*  
*tp(John, Classification, v<sub>1</sub>), v<sub>1</sub> = 'unknown',*  
*Price ≥ private\_value(House)*  
*→ reject(John, House) PC<sub>6</sub>*

*r<sub>7</sub> :*

- *propose(John, Tom, House, 5000, 1),*  
*tp(John, Classification, v<sub>1</sub>), v<sub>1</sub> = 'unknown',*  
*private\_value(House) = p, p ≤ 5000,*  
*tp(John, Gain, v<sub>2</sub>), v<sub>2</sub> = 5000 - p*  
*→ accept(John, House),*  
*tp(John, No\_Negotiations\_with\_Partner, 1),*  
*tp(John, No\_Successful\_Negotiations, 1) PC<sub>7</sub>*

During negotiation steps, the C4.5 learning algorithm can classify the partner in another cooperation class, if the criteria used as attribute in the algorithm are changed.

So, instead of having a negotiation strategy which applies all the rules that match at a certain moment, the classification with the C4.5 learning algorithm decreases the number of eligible rules.

**5. Related Work.** The work reported here is connected to our previous developments of a multi-agent system framework for negotiation. In [7], we have presented a system of self-interested agents that are endowed, besides the widely accepted beliefs-desires-intentions notions, with goals, preferences, obligations, and norms. The inference rules that guide negotiation are based both on cost and gain, and on the cooperation profile the agent develops during previous interactions with other agents in the system. However, the cooperation profile is unique and considers another set of attributes, than the ones in our current model. In [7], we have defined several types of agents, by varying their behaviour according to obligation compliance, self-interestedness, desire to develop good cooperation relations with other agents in the system or simply desire to obtain the maximum gain.

In [8], we have reported on a model of heuristic negotiation between self-interested agents, which allows negotiation over multiple issues of the negotiation object, comprises different types of negotiation primitives, including argument based ones, and a set of rules to conduct negotiation. In order to negotiate strategically and to adapt negotiation to different partners, the agents use rewards associated to negotiation objects and the notion of regret to compare the achieved outcomes with the best possible results that could have been obtained both in a particular negotiation and in selecting the partner agent.

The decision process during the negotiation is modeled as an adversarial bandit problem with partial information and uses the computed probabilities of negotiation rules to select the best rule to be used at a certain moment during negotiation. Rewards were defined depending on the attributes of the negotiation object at a given negotiation round or, in case adaptation of partner selection is sought, depending on the negotiation object with which the negotiation is concluded. Moreover, we have shown how the problem can be modeled if not all rules can be selected at a given decision point (equivalent with not all experts being available for consultation).

In [11], the authors apply the Q-learning algorithm to analyze and learn customer behaviours and then recommend appropriate products. As compared to our approach, the user profile is not used for negotiation, but to personalise the information to the user interests. The authors use weighting features to recommend products to the user; we use weights to represent the preference coefficients.

In [12], the authors propose a software framework for negotiation, in which the negotiation mechanism is represented by a set of rules, as in our case. The rules are organized in a taxonomy, and can be used in conjunction with a simple interaction protocol; the negotiation language is based on OWL-Lite. Although the rules allow flexible definition of several negotiation strategies, there are no negotiation profiles and the possibility to modify the negotiation, according to these profiles, as in our case.

An implementation of automated negotiations in an e-commerce modeling multi-agent system is described in [13]. A specific set of rules is used for enforcing negotiation mechanisms. An experiment involving multiple English auctions performed in parallel is discussed.

A system for automated agent negotiation, based on a formal and executable approach to capture the behaviour of parties involved in a negotiation is shown in [14]. The negotiation strategies are expressed in a declarative rules language, defeasible logic, and are applied using the implemented system Dr-Device.

In [4], the authors present a system, called GENIUS (General Environment for Negotiation with Intelligent multi-purpose Usage Simulation), that supports the design of different strategies for agent negotiation, and the evaluation of these strategies in a simulated environment. The system allows the negotiation between automated agents, but also between agents and humans. The designer of a strategy can select from a repository a negotiation domain and a preference profile for the agent. Both are represented in a tree-like structure, which enables to specify priorities related to outcomes of negotiation. As compared to this system, in our approach, there are several negotiation profiles, which are evolved during interactions, and the negotiation domain is specified by the agent rules.

**6. Conclusions and Future Work.** In this paper, we have presented a model of negotiating agents that aims to combine the agents' beliefs about the other agents in the system, with the possibility to explicitly represent and modify the negotiation strategy, expressed in a set of rules. In order to achieve this, we have defined three negotiation profiles: the preference profile, the partner cooperation profile and the group-of-partners negotiation profile. The last two, partner and group-of-partners, are profiles developed during interactions and they are gradually built, as the agent is taking part to more and more negotiation rounds. The group profile is obtained by applying C4.5 algorithm, and allows the classification of negotiation partners in different classes. Once these classes are available, the agent can decide much quicker on the behaviour to adopt, regarding a partner agent, than in case of the single preference profile.

The negotiation strategy is explicitly represented as a set of rules, together with the preference coefficients, associated to the rules. The preference profile is formed of these coefficients, which can be fixed by the system designer or can be evolved, using a reinforcement-like algorithm.

The behaviour of the agents is motivated by the gain they obtain when realizing their goals or by the necessity to cooperate with other agents, in order to achieve these goals. During negotiation, the agents beliefs on the other agents are updated, as the agent comes to know more about the others.

Because the agents' preferences are based on their interests, the preference coefficients can be modified in time. The agents can modify their preferences over negotiation outcomes, when receiving new information.

The current system is under development and has been tested on some simple cases of electronic transactions. The system prototype has been implemented in Jade; the preliminary results have shown good performances.

The work reported here is an extension of our previous research presented in the Workshop on Agents for Complex Systems (ACSys 2012), Timisoara, Romania.

A future research direction is to use an alternate approach to develop group profiles. Namely, the k-means clustering algorithm, which will remove the necessity to determine in advance the cooperation classes.

Another future line of work is to investigate an alternate approach to update the preference coefficients. This may be done using genetic algorithms. Moreover, genetic algorithms that use rule-specific genetic operators can be used to evolve new strategy rules, based on the existing ones.

Testing the system with a large number of agents and in a real-world environment is the next future work.

**Acknowledgments.** This work was supported by the project ERRIC No. 264207, FP7-REGPOT-2010-1.

## REFERENCES

- [1] T. ITO, H. HATTORI, AND M. KLEIN, *Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces*, in Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 1347–1352.
- [2] C. JONKER, V. ROBU, AND J. TREUR, *An agent architecture for multi-attribute negotiation using incomplete preference information*, Autonomous Agents and Multi-Agent Systems, 15 (2007), pp. 221–252.
- [3] R. LIN, Y. OSHRAT, AND S. KRAUS, *Investigating the benefits of automated negotiations in enhancing peoples negotiation skills*, Proceedings of the 8th International Conference on Automated Agents and Multi-agent Systems, 2009, pp. 345–352.
- [4] R. LIN, S. KRAUS, J. WILKENFELD, AND J. BARRY, *Negotiation with bounded rational agents in environments with incomplete information using an automated agent*, Journal of Artificial Intelligence, 172 (2008), pp. 823–851.
- [5] N. R. JENNINGS, P. FARATIN, A. R. LOMUSCIO, S. PARSONS, C. SIERRA, AND M. WOOLDRIDGE, *Automated negotiation: prospects, methods and challenges*, International Journal of Group Decision and Negotiation, 10 (2001), pp. 199–215.
- [6] S. RADU AND A. M. FLOREA, *An adaptive multi-agent system for e-commerce*, Proceedings of Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems, 2012, pp. 297–300.
- [7] A. M. FLOREA AND E. KALISZ, *Conceptual models of multi-agent systems*, Proceedings of the 14th International Conference on Control Systems and Computer Science, 2003, pp. 394–399.
- [8] A. M. FLOREA AND E. KALISZ, *Adaptive Negotiation Based on Rewards and Regret in a Multi-agent Environment*, IEEE Computer Society Press (CPS), 2007, pp. 254–259.
- [9] J. QUINLAN, *C 4.5: Programs for machine learning*, Morgan Kaufmann, 1992.



- [10] M. V. BUTZ, *Learning classifier systems*, Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, 2010, pp. 2331–2352.
- [11] A. SRIVIHOK AND P. SUKONMANEE, *E-commerce intelligent agent: personalization travel support agent using Q Learning*, Proceedings of the 7th International Conference on Electronic Commerce, 2005, pp. 287–292.
- [12] C. BARTOLINI, C. PREIST, AND N. R. JENNINGS, *A software framework for automated negotiation*, Proceedings of the 3rd International Workshop on Software Engineering for Large-Scale Multi-Agent System, 2005, pp. 213–235.
- [13] C. BADICA, A. BADITA, AND M. GANZHA, *Implementing rule-based mechanisms for agent-based price negotiation*, Proceedings of the 2006 ACM Symposium on Applied Computing, 2006, pp. 96–100.
- [14] T. SKYLOGIANNIS, G. ANTONIOU, N. BASSILIADES, G. GOVERNATORI, AND A. BIKALIS, *Dr-Negotiate A system for automated agent negotiation with defeasible logic-based strategies*, Journal Data and Knowledge Engineering, 63 (2007), pp. 362–380.

*Edited by:* Marc Eduard Frîncu

*Received:* Jan 13, 2013

*Accepted:* Apr 6, 2013