IADIS International Journal on Computer Science and Information Systems
Vol. 5, No.2, pp. 86-97
ISSN: 1646-3692

AGENTS THAT LEARN WHAT ARGUMENT TO SELECT IN ARGUMENTATION-BASED NEGOTIATIONS

Ariel Monteserin. *ISISTAN, Fac. Cs. Ex., UNCPBA – CONICET. Campus Universitario, Pje. Arroyo Seco, Tandil, Argentina*
amontese@exa.unicen.edu.ar

Analía Amandi. *ISISTAN, Fac. Cs. Ex., UNCPBA – CONICET. Campus Universitario, Pje. Arroyo Seco, Tandil, Argentina*
amandi@exa.unicen.edu.ar

ABSTRACT

Argument selection is considered the essence of the strategy in argumentation-based negotiation. An agent, which is arguing during a negotiation, has to decide what arguments are the best to persuade the opponent. In fact, in each negotiation step, the agent must select an argument from a set of candidate arguments by applying some selection criterion. For this task, the agent observes some factors of the negotiation context, for instance trust in the opponent, expected utility, among others. Usually, argument selection mechanisms are defined statically. However, as the negotiation context varies from a negotiation to another, defining a static selection mechanism it is not useful. For this reason, we present in this paper a novel approach to personalize argument selection mechanisms in the context of argumentation-based negotiation. The selection mechanism defines a set of preferences that determine how preferable it is to utter an argument in a given context. Our approach maintains a hierarchy of preferences in order to learn new preferences and update the existing ones as the agent experience increases. We tested this approach in a simulated multiagent system and obtained promising results.

KEYWORDS

Argument selection, argumentation-based negotiation, autonomous agents.

1. INTRODUCTION

In multi-agent systems, autonomous agents need to interact with one another to achieve their goals because reciprocal dependencies exist among them. In this context, negotiation is a fundamental tool to reach an agreement among agents with conflicting goals. The essence of the negotiation process is the exchange of proposals. Agents make proposals and respond to proposals in order to converge on a mutually acceptable agreement. However, not all approaches are restricted to that exchange of proposals. Several approaches to automated negotiation have been developed. One of them is the argumentation-based approach (see e.g. Kraus et al., 1998; Sierra et al., 1998; Ramchurn et al., 2003; Rahwan et al., 2004; Amgoud et al., 2007; Geipel and Weiss, 2007). In argumentation-based approaches, agents are allowed to exchange some additional information as arguments, besides the information uttered on the proposals. Thus, in the context of the negotiation, an argument is seen as a piece of information that supports a proposal and allows an agent (a) to justify its position of negotiation, or (b) to influence the position of negotiation of other agents (Jennings et al., 1998).

In contrast to agents without an argumentative ability, an argumentative agent, in addition to evaluating and generating proposals, must be able to evaluate, generate and select arguments (Ashri et al., 2003; Rahwan et al., 2004). Argument evaluation processes incoming arguments and updates the agent's mental state as a result. Argument generation and selection are related to the production of outgoing arguments. When the agent has to argue during a negotiation, it generates first a set of candidate arguments, for example by using explicit rules (Kraus et al., 1998; Rahwan et al., 2004), and then the agent selects what argument utter by applying a selection mechanism. This selection mechanism usually observes the context of the negotiation and decides which type of argument the agent has to utter. Several factors of the negotiation context are taken into account in the argument selection mechanism: trust in the opponent (Rahwan et al., 2004), agreement urgency, authority relation with the opponent (Sierra et al., 1998), expected utility, argument strength (Kraus et al., 1998), among others. Generally, the selection mechanism is composed of a set of explicit rules that determines which factors have to be observed and which type of argument the agent has to utter. Nevertheless, these mechanisms do not take into account the process of learning new rules or updating existent ones. Because of the constant appearance of new factors, opponents and types of agreements in the negotiation context, learning is essential. In addition to that, opponents are heterogeneous, thereby, we cannot think that all opponents, in the same context, will respond to the same arguments in the same way.

Argument selection is considered as the essence of the strategy in argumentation-based negotiation (Rahwan et al., 2004). Therefore, the success of the negotiation will depend on the accuracy of this mechanism. In this work, we propose a novel approach to learn argument selection preferences in the context of argumentation-based negotiation. These preferences determine how suitable it is to utter an argument in a given context. Each preference is composed by the argument, the set of factors that describe the negotiation context (trust, authority role, urgency, utility, among others), and a preference level described by two values: support and confidence. The preferences are structured in a hierarchy. At the top levels of the hierarchy are situated the most general preferences and in the low levels, the most particular ones. Initially, this hierarchy is empty, but new preferences are added as the agent gains experience by arguing in different negotiations. This allows us to add new factors to the preferences dynamically and make them more specific. In addition, we update the support and confidence values of each preference taking into account the success or failure of the argument uttered (for example, an argument is successful when it is accepted by the opponent).

We have tested our proposal in a simulated multiagent system in which the agents have to negotiate with other agents to reach an agreement. We have obtained promising results. We compared the argument success rate between an agent choosing the arguments randomly and an agent using preferences for argument selection. This comparison was made in a static context as well as in a dynamic one. In a static context, we found that the success rate of the first agent was 45%, whereas the second agent started at 40%, increased logarithmically and reached a final success rate of 70% after finishing the experiments. In the dynamic context, the agent that uses selection preferences obtained a better success rate, too. Initially, the success rate increased during the first interactions, but then it stayed the same due to the fact that the context changed dynamically.

The paper is organized in the following way. Section 2 introduces concepts and related work in the area of argumentation-based negotiation. Section 3 presents the approach to learn argument selection preferences. Section 4 presents the experimental results. Section 5 discusses how to use the proposed approach to assist users that argue in CSCW. Finally, in Section 6, concluding remarks and future work are described.

2. ARGUMENT SELECTION IN ARGUMENTATION-BASED NEGOTIATION

In accordance with the work of Rahwan et al. (2005), there are two major strands in the literature on argumentation-based negotiation: (a) attempts to adapt dialectical logics for defeasible argumentation by embedding negotiation concepts within these (Amgoud et al., 2000, Parsons et al., 1998); and (b) attempts to extend bargaining-based frameworks by allowing agents to exchange rhetorical arguments, such as promises and threats (Kraus et al., 1998; Sierra et al., 1998; Amgoud and Prade, 2005). Our work is situated in the second strand.

As we have introduced above, in an argumentation-based negotiation approach, agents can exchange arguments in order to justify their proposals, to persuade their opponent, and to reach an expected agreement. In contrast to agents without this argumentative ability, an argumentative agent must be able to (a) evaluate incoming arguments and update its mental state as a result; (b) generate candidate outgoing arguments; and (c) select an argument from the set of candidate arguments (Ashri et al., 2003). An argument is a set of one or more meaningful declarative sentences known as the premises along with another meaningful declarative sentence known as the conclusion. There are several types of rhetoric arguments that an agent can generate in the argumentation-based negotiation context. Three general argument types are defined in the literature on argumentation-based negotiation: appeals (Amgoud and Prade, 2004, define them as explanatory arguments), rewards and threats (Kraus et al., 1998; Sierra et al., 1998). Appeals are used to justify a proposal; rewards to promise a future recompense; and threats to warn of negative consequences if the counterpart does not accept a proposal. Moreover, varying the premises of the appeals, we can define several subtypes: past promise, counterexample, prevailing practice, self-interest, among others.

We will focus on the selection of arguments. Rahwan et al. (2004) consider argument selection as the essence of the strategy in argumentation-based negotiation. Argument selection is concerned with selecting the argument that should be uttered to a counterpart from the set of candidate arguments generated by the argument generation process. Once the candidate arguments have been generated, the argument selection mechanism must apply some criteria, in accordance with the agent's mental state, to select the best argument. Argument selection mechanisms are diverse. Kraus et al. (1998) define that the candidate arguments are ordered by their severity, then they select the weakest, taking into account appeals as the weakest argument and threats as the strongest argument. Ramchurn et al. (2003) define rules for argument selection by observing the trust in the opponent and the expected utility of the proposal. For example, they state that if the trust is low and the utility is high then the agent should send a strong argument, but if the trust is high and the utility low, then it should utter a weak one. In the work of Sierra et al. (1998), several authority roles among agents are taken into account to generate and evaluate arguments. Moreover, other factors influence the negotiation the process and they should be taken into account during the argument selection. For instance, the time available to reach the agreement influences directly the negotiation process, affecting the agent behaviour in different ways: the agent can be patient or impatient. Thus when the agent is patient, it gains utility with time and when the agent is impatient, it loses utility with time (Fatima et al., 2004). Other works analyse the information that composes each argument. Schroeder (1999) chooses the shortest argument in order to reduce the target to counter-argue. Amgoud and Prade (2003) assign a strength to each argument in accordance with the beliefs with which it was built. All these works establish different factors and rules to select the best argument. However, they define static mechanisms for argument selection. That is, they do not define how to learn and update the selection criteria nor how to integrate different factors or incorporate new ones.

Additionally, the design of negotiation strategies has been studied. Rahwan et al. (2003) determine that a negotiation strategy may be defined as a rule or algorithm which specifies what the agent should utter and when, in a particular negotiation interaction. In that direction, Rahwan et al. identify some factors that may influence the design of the strategy. Among these factors, we can stress: goals (what goals the agent wants to achieve from undertaking a negotiation), counterparts (the nature of the other participants), resources (the time and the resources available for the agent), among others. Therefore, the argumentation selection process, as an essential part of the argumentation-based negotiation strategy, may take into consideration these factors too.

In the next section, we are going to present an approach to learn and update argument selection preferences, which are the base of a dynamic argument selection mechanism. In contrast to the approaches presented above, our approach allows the agent to incorporate dynamically new factors and to improve the accuracy of the selection mechanism as the agent's experience increases.

3. LEARNING ARGUMENT SELECTION PREFERENCES

As we have shown, several works in argumentation-based negotiation establish rules to decide which argument an agent should utter in a given situation. However, these rules are static and do not contemplate learning. Moreover, we have remarked that several factors influence the argument selection, in particular, factors related to goals, counterparts and resources, which have a strategic bearing on this process (Rahwan et al., 2003).

Learning is an essential ability if we want the agent to improve its performance as it gains experience. Specially, learning how to argue is a promising idea (Emele et al., 2006). As we stated previously, argument selection mechanism has direct influence in the final result of the negotiation. So, it is important for this mechanism to be effective. For this goal, we think that the agent must capture all information available after a negotiation and update the criteria applied by the selection mechanism. In this direction, we propose an approach to learn and update argument selection preferences. Thus, our selection mechanism defines a set of preferences that determine how preferable is to utter an argument in a given context.

3.1 Preference and Context Format

First, we define a structured format to represent the preferences about argument selection. This format is the following: *preference(argument(TYPE, SENDER, RECEIVER, CONC, [PREM]),[CONTEXT], S, C)*[1], where *TYPE* is the kind of rhetoric argument; *SENDER* is the agent that is uttering it; *RECEIVER* is the agent that will receive it; *CONC* is its conclusion; *PREM* is the set of premises that compose the argument; *CONTEXT* is a set of factors in which the argument will be uttered; *S* is the support of the preference, and *C* the confidence value. The support *S* is defined as:

$$Support\ (arg) = \frac{count_{arg}}{count_{tot}}$$

where *arg* represents *argument(TYPE, SENDER, RECEIVER, CONC, [PREM])*, $count_{arg}$ is the number of times that an argument that matches with *arg* was uttered by the agent, and $count_{tot}$ is the total number of arguments uttered by the agent. On the other hand, the confidence *C* indicates the success rate of the preferences and it is defined as:

$$Confidence\,(arg) = \frac{success_{arg}}{count_{arg}}$$

where $success_{arg}$ is the number of times in which an argument that matches with *arg* was successful. To determine the preference level, we multiply *S* by *C*.

The context is represented by a set of variables. These variables depict the factors that influence the negotiation and they can change based on the negotiation domain. For example, these variables can be:

- *utility(Ut)*: it represents the utility associated to a proposal that motivated the negotiation. *Ut* can take three values: low, medium and high.
- *urgency(Ur)*: it corresponds to the urgency of the sender to reach the agreement. *Ur* can take three values: patient, medium and impatient.
- *trust(T)*: it denotes the level of trust between sender and receiver. *T* can take three values: low, medium and high.
- *authority(A)*: it indicates the relation of authority between sender and receiver. *A* can take three values: subordinated, peer and superior.

As regards the factors that influence the design of negotiation strategies defined by Rahwan et al. (2003), we can state that utility is related to the goals of the agent, urgency is related to the resources (time), and trust and authority, to the counterparts of the negotiation.

3.2 Preference Learning Process

To improve the effectiveness of the argument selection mechanism as the agent gains experience, we distinguish two goals of the preference learning process:

[1] Parameters whose names start with an uppercase character are variables.

- Preference level update: the process has to update the support and confidence values of the preferences. To do this, we take into account the correlation between the desired effect of the argument and the real effect that it produces in the negotiation.
- New preferences addition: the initial preferences are empty or lack specificity. So, it is necessary that the learning process adds new preferences, more specific, as the negotiations take place. In this sense, when the experience of the agent increases, the specificity of the preferences and the accuracy of the argument selection will increase too. At the same time, we want the information gathered by the process to be useful in unexpected negotiation situations.

The input of the preference learning process is the log of a negotiation (see Figure 1). This log contains the locutions uttered during the negotiation and the context in which they were uttered. An example of this log following the negotiation protocol defined by Sierra et al. (1998) is:

1. *a1 requests a2 to do action1.*
2. *a2 rejects to do action1.*
3. *a1 utters a reward saying "if a2 does action1, a1 will do action2".*
4. *a2 accepts to do action1.*

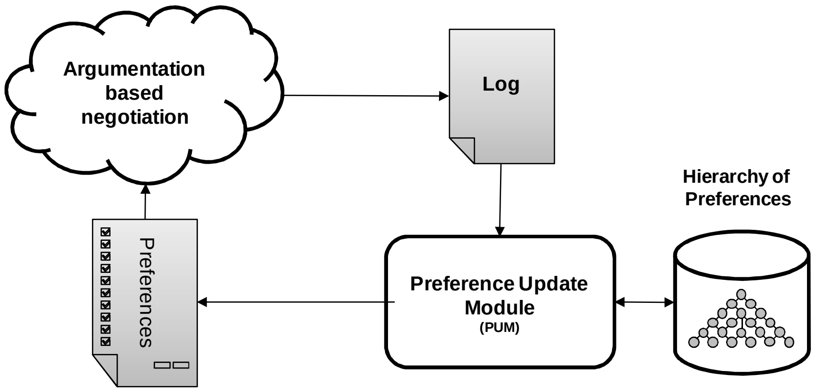

Figure 1. Preference learning process

Following this example, we suppose that the utility of the execution of the *action1* is low and its urgency is medium, and the agent *a2* is subordinated to agent *a1* and *a1* trusts completely in *a2*, so the context should be defined as *utility(low), urgency(medium), trust(high), authority(subordinated).* From this log, we can extract an argument: the reward represented by *"if a2 does action1, a1 will do action2"*. Formally, this reward can be expressed as *reward(a1; a2; do(a2, action1); [do(a1, action2)])* (see Sierra et al., 1998). As we can see, the argument was successful, because *a2* accepted to do *action1*.

```
1.  L = {Negotiation log}.
2.  H = {hierarchy of preferences}
3.  forall Argument a ∈ L {
4.    s = success_factor(a, L).
5.    if preference_exist(a, H) { // does the preference
      corresponding to the argument a exist in the hierarchy H?
6.        p = preference(a, H).
7.        forall preference pAnc ∈ ancestors(p, H)
8.                   preferences_update(pAnc, s).
9.    }
10.   else {
11.       p = new Preference(a, s).
12.       add_to_hierarchy(H, p, s). // It adds the preference to the
          hierarchy H and generates the ancestors to link to H. It
          uses the success factor to initialise the values of support
          and confidence.
13.   }
14. }
```

Figure 2. Algorithm for argument selection preferences update

In order to achieve the above goals, the Preference Update Module (PUM) processes the log of a negotiation using the algorithm described in Figure 2. The first task is to determine the success or failure of the arguments (step 4). We adopt a trivial vision for this task: if the negotiation finished with the expected agreement, the argument would be correct. The function *success_factor(a, L)* returns a success factor *s* for each argument recorded in *L*, where *s* is TRUE if the negotiation finished with the expected agreement, and FALSE if the argument was refused and the negotiation finished in conflict. For the previous reward, *s* is TRUE.

As we can see in the Figure 1, the PUM maintains a hierarchy of preferences. This hierarchy has in its top levels the most general preferences, for example p_1: *preference(argument(reward, a1, _, _, _), _, 0.4, 0.65)*[2]; and in its leaves, the most specific ones. Preference p_1 represents the fact that 40% of the arguments uttered by the agent *a1* were rewards, and that 65% of these rewards were successful, so the preference level to utter rewards is 0.26. The relation between preferences that originates the hierarchy is the inclusion of a child preference in a parent preference. In others words, a child preference gives more details to a parent preference in some of its parameters (sender, context, etc.). For instance, p_2: *preference(argument(reward, a1, a2, _, _), _, 0.24, 0.75)* is a child of p_1, since p_2 specify the receiver (*a2*). Furthermore, we can give more specificity with the context. For example, p_3: *preference(argument(reward, a1, a2, _, _), [utility(high)], 0.18, 0.66)* is a child of preference p_2, because it details the utility associated with the final agreement; and p_4: *preference(argument(reward, a1, a2, _, _), [utility(high), trust(high)], 0.12, 0.83)* is a child of p_3.

The generation a priori of this hierarchy will be really hard and inefficient. For this reason, the PUM is responsible for adding new preferences to the hierarchy. The arguments recorded in the log are instanced in the negotiation context and with the major information about the negotiation context that could be obtained. Then, after obtaining the success factor *s* for an argument *arg*, the PUM checks if the preference that exactly corresponds to *arg* exists in the

[2] Symbol "_" represents an unnamed variable, like in Prolog syntax.

hierarchy (step 5). If the preference exists, the module increases $count_{arg}$ and $count_{tot}$, and if *s* is TRUE, it increases $success_{arg}$. Then, the support *S* and the confidence *C* of the preference, and all its ancestors, is recalculated (step 6 to 8). Notice that the preferences situated over the hierarchy obtain more information since they are more general, as a consequence, the support will be higher. As the negotiations occur, the information will be propagated to the lower levels. In contrast, if the preference does not exist in the hierarchy, the module will create a new leaf with it (step 11) and generate the branch that links this leaf with the rest of the nodes taking into account the inclusion relation explained above (step 12). In this case, the module increases $count_{tot}$, initialises $count_{arg}$ with 1, and if *s* is TRUE, it initialises $success_{arg}$ with 1, otherwise $success_{arg}$ will be initialised with 0.

Let's see an example. Given a hierarchy composed of p_1, p_2, p_3 and p_4, the argument *reward(a1; a2; do(a2, action1); [do(a1, action2)])* and the context, we will update the preferences by following the proposed algorithm. First, as we have previously stated, the success factor *s* of the reward is TRUE, due to the fact that *a2* accepted to execute *action1* (step 4). Moreover, we suppose that $count_{tot}$ = 50; $count_{arg1}$= 20 and $success_{arg1}$ = 13[3]; $count_{arg2}$ = 12 and $success_{arg2}$ = 9; $count_{arg3}$ = 9 and $success_{arg3}$ = 6; and $count_{arg4}$ = 6 and $success_{arg4}$ = 5. Then, as there is no preference that exactly represents the reward, we have to create a new one (step 11), p_n: *preference(argument(reward, a1, a2, do(a2, action1), [do(a1, action2)]), [utility(low), urgency(medium), trust(high), authority(subordinated)], 0.02, 1)*. The support *S* of p_n is 0.02 and the confidence *C* is 1, because the new value of $count_{tot}$ is 51 and $count_{argn}$ and $success_{argn}$ are initialised with 1. Next, we build the branch of preferences that link p_n with some preference of the hierarchy (step 12). Some of the preferences of this branch could be: *preference(argument(reward, a1, a2, _, _), [utility(low), urgency(medium), trust(high), authority(subordinated)],0.02 , 1)* and *preference(argument(reward, a1, a2, _, _), [utility(low), urgency(medium), trust(high)], 0.02, 1)*[4]. In this case, the preference, where p_n links, is p_2. The preferences p_3 and p_4 do not link to p_n because their context include the fact *utility(low)* in contrast to *utility(high),* which is presents in p_n. Therefore, we have to update the preferences p_1: *preference(argument(reward, a1, _, _, _), _, 0.41, 0.66)* and p_2: *preference(argument(reward, a1, a2, _, _), _, 0.25, 0.77)*. Additionally, as the $count_{tot}$ has changed, the support of p_3 and p_4 changed too.

3.3 Argument Selection Using Preferences

Finally, after updating the hierarchy, we must define how the best argument is selected. The idea is simple, the argument selection mechanism averages the preference level (*S* times *C*) of all preferences that match with each candidate argument and then it selects the argument with the best mean. For instance, the argument *reward(a1; a3; do(a3, actionA); [do(a1, actionB)]),* in the context *[utility(medium), urgency(medium), trust(low)],* can be matched with the preference p_1, so the preference level will be 0.27. In contrast, the argument *reward(a1; a2; do(a2, actionA); [do(a1, actionB)]),* in the same context, can be matched with the preferences p_1 and p_2, so the preference level will be 0.235.

In summary, the use of the hierarchy of preferences allows the agent to capture experience from past argumentations with different degrees of accuracy. In the low levels, the preferences give more details about what argument the agent has to utter in a given situation, but with a

[3] $count_{argi}$ and $success_{argi}$ correspond to the preference p_i.
[4] We only show some preferences of the branch, due to the space limitations.

low support. These detailed preferences are appropriate when the negotiation context is similar to past negotiations. In contrast, top levels are composed of general preferences that can be specially applied in unknown contexts.

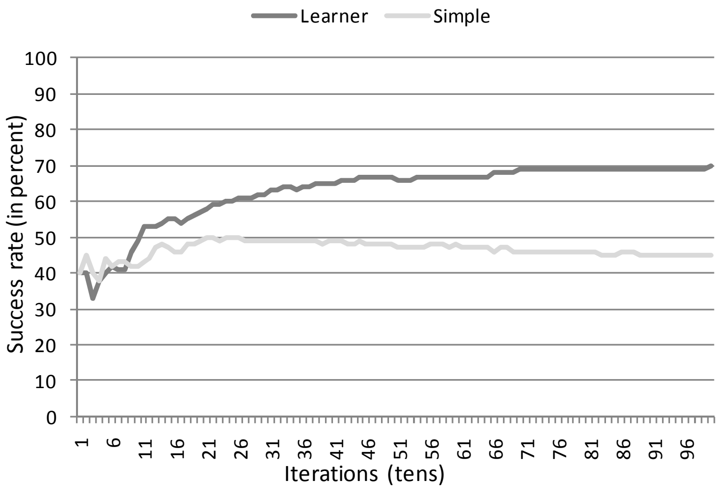

Figure 3. Comparison of success rate in static contexts

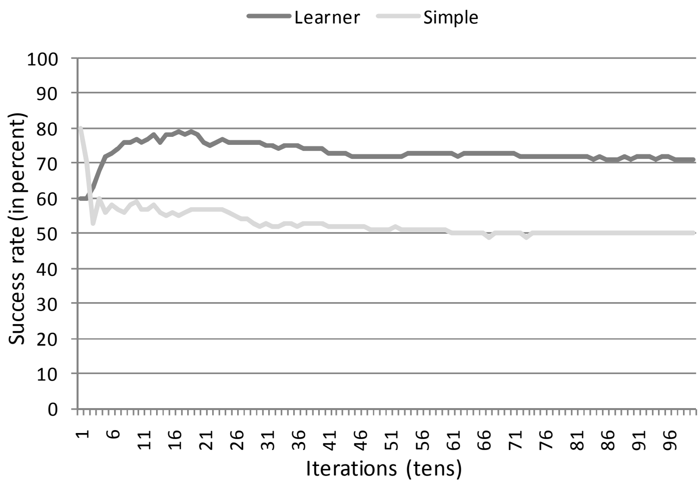

Figure 4. Comparison of success rate in dynamic contexts

4. EXPERIMENTAL RESULTS

To evaluate our proposal, we simulated a multiagent system in which the agents have to negotiate with other agents to reach agreements. To contrast our results, we compared the performance of an agent that learns argument selection preferences (*learner agent*) with an agent that does not learn them (*simple agent*). We did the comparison in a static and a dynamic context. In the static context, the multiagent system was composed of four heterogeneous agents[5]. One of them was the negotiator agent and the rest were opponents. Each opponent was set with the trust and the authority levels. We ran 1000 argumentative iterations. In each iteration, the agent had to solve a conflictive situation. A conflictive situation was specified by the agreement that the negotiator agent needed to reach (for instance, the execution of *action1*), and the opponent with which the agent had to negotiate (for example, agent *a2*). Also, the urgency and the utility levels were defined for each expected agreement. Following this specification, the agent generated a set of candidate arguments to persuade the opponent. Then, the agent selected an argument from this set by applying the argument selection mechanism, and sent it to the opponent, which accepts or rejects it by taking into account its individual preferences. During the negotiations, the argument exchange was registered in the negotiation log. Finally, the preference update was carried out every ten iterations, that is, the log processed by the PUM was composed of the arguments uttered in ten conflictive situations.

After each update, we calculated the success rate of the arguments uttered by the agent. Figure 3 shows a chart in which we compare the rate of success of the learner agent and the simple agents. We can appreciate that as the learner agent gains experience its success rate increases significantly. It starts in 40% and finishes in 70%. In contrast, the simple agent does not improve its success rate, but it keeps between 40% and 50%.

In the dynamic context, the opponents were changed every 100 iterations. That is, new opponents with different characteristics were introduced into the multiagent system to be part of the negotiations and the previous opponents were removed. In this context, the more specific preferences are not useful, but the general ones are. Figure 4 shows the chart where we compare the success rate of the learner agent and the simple agent. We can observe in the dynamic context that the rate of success of the learner agent remains higher than the rate of the simple agent.

If we compare the charts of Figure 3 and Figure 4, we can note that in the static context the success rate increases in a logarithmical way, whereas in the dynamic one the increment of the success rate occurs in the first 100 iterations and then the rate keeps constant. We think that this is caused by the continuous change of the opponents. In the static context, the preference update process improves the success rate in each iteration, due to the fact that the context does not change. In contrast, in the dynamic context, the learning is particularly effective when the preferences are empty. After reaching certain level of details, the preference update process keeps the success rate constant.

5. ASSISTING USERS THAT ARGUE IN CSCW

When a user participates during a discussion in a CSCW system (*Computer Supported Collaborative Work*), he/she must generate arguments to persuade his/her opponents. After generating these arguments, he/she must select the best one to be uttered during the argumentation by taking into account the negotiation context. For these reasons, we claim that

[5] The agents were implemented following the frameworks defined in [17,10,15].

a personal agent could automatically select the argument in a personalised way and suggest it to the user by using our approach to learn argument selection preferences.

We assume that we can access to the contextual information of the discussion in which the user is participating. For example, a personal agent (Maes, 1994), which observes the computational environment, can detect the intentions (Kautz, 1987; Charniak and Goldman, 1993) of the user or can observe both the proposal that he/she utters and the additional information of the discussion. Once the intentions or the proposals are detected and the information is gathered, a personal agent, which assists a user, can observe the negotiation context, the argument uttered and its effects, and update the preferences following the process defined in Section 3. After updating the preferences, the personal agent can assist the users by indicating which argument should be uttered according to the argumentative situation.

Also, this approach can be integrated with the user argumentative model presented in Monteserin y Amandi (2010). These models capture the argumentative style of the users by discovering the argument generation rules that the user applies to build his/her arguments. Thus, both approaches will comprise the generation and selection of incoming arguments.

6. CONCLUSIONS AND FUTURE WORKS

We have presented a novel approach to learn argument selection preferences in the context of argumentation- based negotiation. These preferences determine how preferable is to utter an argument in a given negotiation context. This context is composed of the agents that are part of the negotiation and a set of factors that influence the selection process (for example, utility expected, trust in the opponent, urgency, authority roles, etc.). These factors can be added dynamically as the agent updates the preferences taking into account past negotiations. We think that this point is especially relevant due to the fact that previous works define the factors, which have an effect in the argument selection, statically. To allow the use and the update of the preferences, they are structured in a hierarchy, in which top levels are the most general preferences, and the most particular ones, in the low levels. As we stated above, this difference of detail among the levels of the hierarchy allows the approach to be efficient in static contexts as well as in dynamic ones. Specific preferences give high accuracy in particular negotiation context. In contrast, general preferences are appropriated for unknown ones. According to the experimental results, we have found that our approach allows the agent to improve the success rate of the arguments.

We have tested our proposal in a simulated multiagent system, however, future works aim to evaluate it in real environments. Other future direction is oriented to explore other machine learning technique to update the argument selection preferences. Specially, we are going to explore the reinforcement learning technique.

REFERENCES

L. Amgoud, Y. Dimopoulos, and P. Moraitis, 2007. A unified and general framework for argumentation-based negotiation. *Proc. AAMAS '07*, pp. 1-8, New York, NY, USA, ACM.

L. Amgoud, S. Parsons, and N. Maudet, 2000. Arguments, dialogue, and negotiation. *Journal of Artificial Intelligence Research*, Vol. 23.

L. Amgoud and H. Prade, 2004. Generation and evaluation of different types of arguments in negotiation. *Proc. of the international workshop on non-monotonic reasoning*, pp. 10-15, Whistler BC, Canada.

L. Amgoud and H. Prade, 2005. Handling threats, rewards, and explanatory arguments in a unified setting: Research articles. International Journal Intelligent Systems, Vol. 20, No. 12, pp. 1195-1218.

R. Ashri, I. Rahwan, and M. Luck, 2003. Architectures for negotiating agents. *Proc. of Multi-Agent Systems and Applications III*, pp. 136-146, Prague, Czech Republic. Springer.

E. Charniak and R. P. Goldman, 1993. A bayesian model of plan recognition. *Artificial Intelligence*, vol.64(1), pp.53-79.

C. D. Emele, F. Guerin, T. J. Norman, and P. Edwards, 2006. A framework for learning argumentation strategies. *Proc. of the Third International Workshop on Argumentation in Multi-Agent Systems*.

S. S. Fatima, M. Wooldridge, and N. R. Jennings, 2004. An agenda-based framework for multi-issue negotiation. Artificial Intelligence, Vol. 152, No. 1, pp 1-45.

M. M. Geipel and G.Weiss, 2007. A generic framework for argumentation-based negotiation. *Proc. of the 11th international workshop on Cooperative Information Agents XI*, pp. 209-223, Berlin, Heidelberg. Springer-Verlag.

N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra, 1998. On argumentation-based negotiation. *Proc. of the International Workshop on Multi-Agent Systems*, Boston, MA.

H. Kautz, 1987. A formal theory of plan recognition, PhD thesis, Dept. of Computer Science, University of Rochester.

S. Kraus, K. Sycara, and A. Evenchik, 1998. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, Vol. 104, No. 1-2, pp. 1-69.

P. Maes, 1994. Agents That reduce work and information overload, *Communications of the ACM*, vol.37(7), pp.30-40.

A. Monteserin and A. Amandi, 2010. Building user argumentative models. *Applied Intelligence*, vol.32(1): pp.131-145.

S. Parsons, C. Sierra, and N. R. Jennings, 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, Vol. 8, No. 3, pp. 261-292.

I. Rahwan, P. McBurney, and L. Sonenberg, 2003. Towards a theory of negotiation strategy (a preliminary report). *Proc. of AAMAS Workshop on Game Theoretic and Decision Theoretic Agents*, Melbourne, Australia.

I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, and L. Sonenberg, 2004. Argumentation-based negotiation. *Knowledge Engineering Review*, Vol. 18, No. 4, pp. 343-375.

I. Rahwan, L. Sonenberg, and P. Mcburney, 2005. Bargaining and argument-based negotiation: Some preliminary comparisons. *Lecture Notes in Computer Science*, pp. 176-191.

S. D. Ramchurn, N. R. Jennings, and C. Sierra, 2003. Persuasive negotiation for autonomous agents: A rhetorical approach. *Proc. of IJCAI Workshop on Computational Models of Natural Argument*, pp. 9-17.

M. Schroeder, 1999. An efficient argumentation framework for negotiating autonomous agents. *Proc. of MAA- MAW99*. Springer-Verlag.

C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, 1998. A framework for argumentation-based negotiation. *Proc. of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pp. 177-192, London, UK.