Rewards in Persuasive Group Decision Making

Ricardo Santos[1,2], Goreti Marreiros[1,3], Carlos Ramos[1,3], José Neves[4] and José Bulas-Cruz[5]

[1] GECAD – Knowledge Engineering and Decision Support Group
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt

[2] College of Management and Technology– Polytechnic of Porto
Felgueiras, Portugal
rjs@estgf.ipp.pt

[3] Institute of Engineering – Polytechnic of Porto
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt

[4] University of Minho
Braga, Portugal
jneves@di.uminho.pt

[5] University of Trás-os-Montes e Alto Douro
Vila Real, Portugal
jcruz@utad.pt

Abstract. Decision Making is one of the most important activities of the human being. Nowadays decision-making involves considering many different points of view, so decisions are commonly taken by formal or informal groups of persons. Groups exchange ideas, engage in a process of argumentation and counter-argumentation, negotiate, cooperate, collaborate and even discuss techniques and/or methodologies for problem solving. Group Decision Making is a social activity in which the discussion and results consider a combination of rational and emotional aspects. In this paper we propose a Multi-Agent model for the representation of the behaviour of individuals as members of a group involved in a decision making processes. It makes use of persuasive argumentation in the interactions between participants that makes use of a reward mechanism. The reward mechanism that we describe here consists in an algorithm to generate and evaluate problem independent rewards in order to allow participants to reach better agreements in a shorter time.

1 Introduction

Group decision making processes are omnipresent in several everyday activities. We can have more formal decision groups like committees or management teams who have to decide for instance if it is more advantageous to acquire a competitor enterprise and their technology or invest in research to develop the technology internally. Or we can have more informal groups like, for instance, a married couple who have to decide what movie to watch.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 99–110

Received 17/06/07
Accepted 31/08/07
Final version 14/09/07

During the past 20 years, several Group Decision Support Systems (GDSS) have been developed, some dedicated to be used exclusively in decision rooms and others with features to support ubiquitous group decision meetings. The main goal of GDSS is to support a group that is responsible for decision making.

A more recent approach to Group Decision support is the use of agent based systems to support or simulate group decision processes. Multi-agent systems are very suitable for simulating the behaviour of groups of people working together [1] and, in particular, for modelling group decision making, as they cater for individual modelling, flexibility and data distribution.

The group decision-making process begins with the identification of the problem. In this stage, relevant information concerning the problem is collected. During the design phase, the group develops and discusses several possible alternative solutions for the problem and establishes the decision criteria. There are quite a few ways for generating candidate solutions, some of them are: brainstorming, reviewing the literature, conducting research and issuing a request for proposals for alternative solutions. Finally, in the choice phase, group members choose a final solution from a set of possible alternatives.

The focus of our research is in the choice phase. Argumentation can be an excellent way to justify possible choices and to convince other elements of the group that an alternative is better or worse than another. Amgoud et al. [2] states that humans often use arguments to explain past choices or to evaluate potential alternatives.

In multi-agent systems literature, argumentation is referred to as a key form of interaction among autonomous agents [3][4], particularly in non-cooperative situations and when agents have to deal with incomplete information about each other and about the environment.

In the context of negotiation, argumentation is viewed as a mechanism whereby information may be exchange. An argument is viewed as a piece of information that may allow an agent to [5]:

- Justify its negotiation decision or option; and
- Influence other agents about the quality of its proposals.

In recent years there has been a significant number of proposals for argumentation-based negotiation that use the arguments referred to above. Among the earliest proposals was Katia Sycara's Persuader System, a framework for intelligent conflict resolution and mediation.

This work led to subsequent research by Sarit Kraus et al. [4], who proposed a logical model for reaching agreements through argumentation. In this model the argumentation style is based on the psychology of persuasion, using threats, rewards, appeals to precedents, appeals to self-interest and appeals to prevailing practices. Agents are characterized over several attributes: whether they are less, knowledgeable, cooperative, etc. The selection of the arguments is based on a pre-defined order.

Sierra et al. [6] proposed a generic framework for argumentation-based negotiation, in which agents exchange proposals and counter-proposals justified by arguments. The presented argumentation protocol is based on a finite state machine where exchanged arguments are persuasive: threats, reward and appeal. The evaluation of the arguments received by an agent is in part based on the Dung argumentation theory and its notion of argument acceptability. In this work it is not defined how agents select arguments to send to other agents, there being no pre-defined order as in the work of Sarit Kraus [4].

In the literature it is possible to see references to threats and promises as the most used arguments in human discussions.

This paper is organized as follows: Section 2 presents the multi-agent model of our system and is also presented the architecture of participant agents [7] describing its main components and interactions. Section 3 describes with more detail the argumentation system responsible for the arguments generation. Section 4 presents the reward mechanism. Section 5 presents some conclusions and directions for future work.

2 Multi-Agent Model and Participant Agent

Agent Based simulation is considered an important tool in a broad range of areas e.g. individual decision making (what if scenarios), e-commerce (to simulate the buyers and sellers behaviour), crisis situations (e.g. simulate fire combat), traffic simulation, military training, entertainment (e.g. movies).

According to the architecture that we are proposing we intend to give support to decision makers in both of the aspects identified by Zachary and Ryder [8], namely supporting them in a specific decision situation and giving them training facilities in order to acquire competencies and knowledge to be used in a real decision group meeting. We maintain that agent based simulation can be used with success in both tasks. As stated in the introduction multi-agent systems seem to be quite suitable for simulating the behaviour of groups of people working together [1][9].

Each participant of the group decision making process is associated with a set of agents for interaction with other participants. The community should be persistent because it is necessary to have information about previous group decision making processes, particularly the credibility, reputation and past behaviors of other participants [10].

There are three different types of agents in our model: Facilitator agent, Assistant agent and the Participant agent.

The Facilitator agent helps the responsible for the meeting in its organization (e.g. decision problem and alternatives definition). During the meeting, the Facilitator agent will coordinate all the processes and, at the end, will report the results of the meeting to the participant responsible for it.

The Assistant agent works as an assistant to the participant of the meeting presenting all the updated information of the meeting. This agent acts like a bridge between the participant (user) and the participant agent.

The participant agent has a very important role in the group decision support system, for that reason we will present the architecture and a detailed view of all the component parts.

The architecture is divided in three layers: the knowledge layer, the communication layer and the reasoning layer (Fig.1).

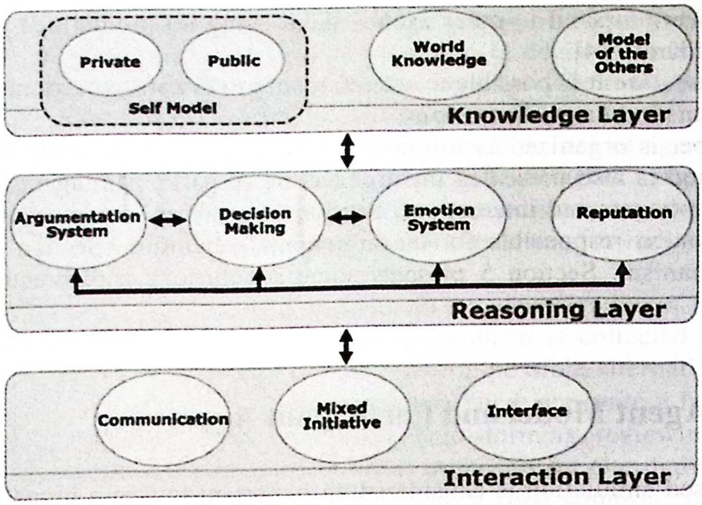

Fig. 1. Participant Agent Architecture

In the knowledge layer the agent has information about the environment in which it is situated, about the public profile of the other participant agents that compose the decision meeting group, and regarding its own preferences and goals (its own public and private profile). The information in the knowledge layer is dotted of uncertainty [8] and will be accurate for the duration of the interactions done by the agent.

The interaction layer is responsible for communication with other agents, the interface with the user of the group decision making simulator and the mixed initiative interaction between participant and agents.

The reasoning layer contains three major components:

- The argumentative system – that is responsible for the generation of arguments. This system will be described in more detailed in section 3.
- The decision making module – will support agents in the choice of the preferred alternative and will classify all the set of alternatives in three classes: preferred, indifferent and inadmissible;
- The emotional system [11] – will generate emotions and moods, affecting the choice of the arguments to send to the others participants, the evaluation of the received arguments and the final decision.
- The reputation module – this module tries to support the user in the definition of the level of trust given to the Participant agent in the delegation of actions.

3 Argumentation System

Arguments may be classified according to type. Here we assume that the following six types of argument have persuasive force in human negotiations [12][13]: threats; promise of a future reward and appeals; appeal to past reward; appeal to counter-example; appeal to prevailing practice; and appeal to self interest. These are the arguments that agents will use to persuade each other. This selection of arguments is compatible with the power relations identified in the political model: reward, coercive, referent, and legitimate [14].

This component will generate persuasive arguments based on the information that exists in the participant's agent knowledge base [15].

3.1 Argumentation Protocol

During a group decision meeting, Participant agents may exchange the following locutions: request, refuse, accept, request with argument.

Request $(AgP_i, AgP_j, \alpha, arg)$ - in this case agent AgP_i is asking agent AgP_j to perform action α, the parameter *arg* may be void and in that case it is a request without argument or may have one of the arguments specified in the end of this section.

Accept (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it accepts its request to perform α.

Refuse (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it cannot accept its request to perform α.

The purpose of the agent participant its to replace the user when he is not available. For example, in Fig. 2, it is possible to see the argumentation protocol for two agents. However, note that one of the participants is not available at the moment leaving all the actions to the participant agent (AgP_2). Note that his is the simplest scenario, because in reality, group decision making involves more than two agents and, at the same time AgP_1 is trying to persuade AgP_2 that agent may be involved in other persuasion dialogues with other group members.

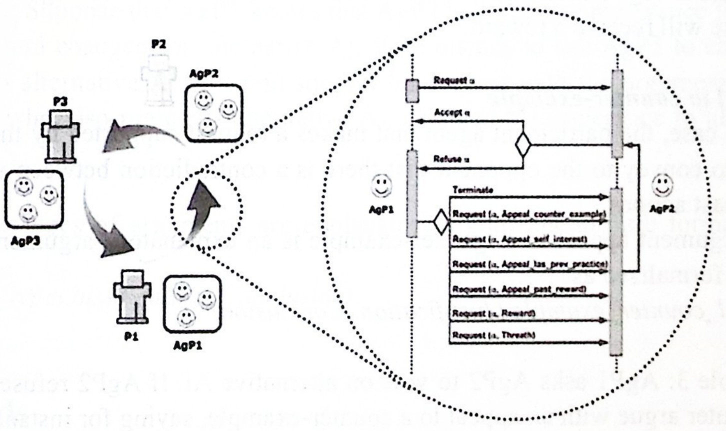

Fig. 2. Argumentation protocol

The autonomy of the participant agent is connected with the trust that the real participant has in the agent. As the agent exchanges locutions with other participants the user can approve or reject the locutions made by the participant agent there are no definitive locutions made by the agent. In that case trust level can be increased or decreased.

3.2 Generation of Arguments

In this section we describe and specify the persuasive arguments that our agents may change. For each argument a short example is given.

Threats

As previously stated, threats are very common in human negotiation, and they can assume two distinct forms: you should perform *action A* otherwise I will perform *action B*; and you should **not** perform *action A* otherwise I will perform *action B*.

In our model this type of arguments may be formalized as a triplet:

Threat(Justification, Conclusion, Threatened_goal)

Example 1: AgP1 asks AgP2 to vote on alternative Ai with the argument that if he refuses he will vote on alternative Aj that he believes is unacceptable for AgP2.

Promises

In this case, the participant that has a request refused can offer a reward for the acceptance of the request. The reward can be claimed by the opponent in the future. This mechanism of rewards will be described in more detail in section 4.

In our model this type of arguments may be formalized as a triplet:

Promise(Justification, Conclusion, Promised_goal)

Example 2: AgP1 asks AgP2 to vote on alternative Ai with the argument that if he accepts he will receive a reward.

Appeal to counter-example

In this case, the participant agent that makes a request supported by this argument, expects to convey to the opponent that there is a contradiction between what he says and his past actions.

The argument appeal to a counter-example is an explanatory argument and in our model is formalized as:

Appeal_counter_example (Justification, Conclusion)

Example 3: AgP1 asks AgP2 to vote on alternative Ai. If AgP2 refuses then AgP1 may counter argue with an appeal to a counter-example, saying for instance that in the past it preferred alternative Ak and changed to Al, so why not to do the same now?

Appeal to self-interest

In this case the participant agent that makes a request supported by this argument expects to convince his interlocutor that making action *A* is in his best interest.

The argument appeal to self-interest is an explanatory argument and in our model is

formalized as:
Appeal_self_interest (Justification, Conclusion)

Example 4: Suppose that AgP1 asks AgP2 not to vote for alternative Ai, supported by the argument that AgP3 voted for the Ai alternative and AgP2 does not like AgP3.

Appeal to past reward
In this case the participant agent that sends such an argument expects his interlocutor to perform an action based on a past promise.
In our model the appeals to past rewards are formalized as:
Appeal_past_reward (Promise(Justification, Conclusion, Promised_goal), Conclusion)

Example 5: If at some point in the past agent AgP1 sent a request to AgP2 to vote on a specific alternative with the promise of a reward that he will stay in a debt of gratitude, and if AgP2 accepted it, in a future decision AgP2 can send a request supported by this argument.

Appeal to prevailing practice
In this case, a participant agent believes that the opponent agent will refuse to perform a requested action since it contradicts one of its own goals. For that reason the participant agent sends a request with a counter-example from a third agent's actions or from past actions of the opponent.
In our model the appeal to prevailing practice is formalized as:
Appeal_prevailing_practice (Justification, Conclusion)

Example 6: Suppose that AgP1 knows that AgP2 had a strong preference for the alternative Ai and changed for alternative Aj. If he intends to ask AgP3 to change his preference to alternative Aj he could support his request with the argument that another agent, who also preferred alternative Ai, changed his preference to alternative Aj.

The last four types of arguments are explanatory arguments and are formalized as tuple:
Argument_type(Justification, Conclusion).

4 Rewards

Usually the participants of a meeting do not all share the same objectives, so it is usually difficult to reach a consensus. Rewards are an extremely important issue in reaching an agreement on the selection of an alternative in a group decision meeting. In the negotiation process participants send requests to others that can be accepted or rejected depending on their personal profile and objectives. In order to obtain better results via unanimous consensus we think it is necessary to have a method to define rewards independent of a specific problem that can be used to justify locutions in future meetings. Depending on the outgoing of the meeting, Participants can offer rewards, evaluate received proposals, and, in the future, in other requests or meetings claim the promised rewards.

Our approach to the rewards mechanism was developed independently of any problem domain and it is divided in two parts: the process of the reward offering; and the process of the reward evaluation. To explain our approach we will use a simple scenario involving two Participant agents: AgP_1 and AgP_2. AgP_1 is the agent participant that in the middle of a meeting more precisely in the negotiation process decides to send a request with a promise of a future reward to AgP_2. AgP_2 is the participant agent who should evaluate that request.

4.1 Process of Reward Offering

The offer of the reward is the responsibility of AgP_1, who sends a request in order to secure an acceptance of the alternative in question. If the Participant agent AgP_2 accepts the reward, in the future he can claim the reward related to this alternative, which may or may not be provided. In more detail, the next algorithm outlines the main steps of the Participant agent AgP_1.

1. Examine the request refused.
2. Calculate a value to formulate the reward.
3. Offer the reward.

Step 1: Examine the request refused

In this first step AgP_1 will examine the public profile of AgP_2, in order to find the reason for the rejection of the request. The refused request has to be analyzed in order to calculate the reward in the next step. From this analysis we can predict several possible results:

- All arguments are defined and in favor of the criteria of the alternative in the public knowledge; the request was refused by a direct order coming from the participant user;
- All arguments are defined in the public knowledge, some against and others in favor of the alternative; the request was refused because there are arguments against the alternative;
- Some arguments are defined as part of the public knowledge and others not. The other missing arguments could be in the private knowledge or could even be not defined;
- There are no arguments defined in the public knowledge. The arguments could be in the private knowledge or not defined.

The first possibility presented is an exception to the reward calculus because all of the criteria for the requested alternative are proved in the public knowledge of the participant. No reward can be defined because all arguments are in favor of the alternative and the request was refused intentionally by the participant user. This action brings us to a problem that is the possibility of the participant user to refuse a preferred request intentionally in order to collect a future reward. There are two distinct ways of solving this problem, one for the Participant agent and the other for the participant user:

- The participant agent may move to a threat argument of the kind "*If you do not accept this request I will refuse your request(s)*";
- The participant user may argue with the other participant.

The other possibilities are solved through the calculus of a maximum reward to be offered to the Participant agent AgP_2, explained in the next step.

Step 2: Calculate a value to formulate the reward

To determine the value of the reward we use a utility function to predict a fair value to offer to AgP_2. This function uses the self-knowledge of AgP_1 to determine the value of the reward, so what the reward will not be over evaluated. This issue has some importance by the fact that in the future AgP_1 will be in debt to AgP_2. The utility function its defined afterwards:

$$UReward(Argument_x) = \frac{WArgument_x}{\sum_{i=1..n} WArgument_i}$$

Where $Argument_x$ is an argument that proves a criterion of an Alternative and $WArgument_x$ is the importance that the participant user gives to the argument in question. For the following alterantive: *$Alternative_1(Price=10)$* a possible argument can be *(Price <15)* and the Wargument *(60 [0-100])*.

As explained previously this function must calculate a reward in several scenarios: when we know what arguments are contrary to the alternative criteria or when these are unknown. To illustrate the use of the reward utility function we will present an algorithm that estimates the maximum reward value. Consider the following assumptions:

- $Request(AgP_1, AgP_2, Alternative_x)$ as a request made by AgP_1 to AgP_2 to accept the alternative$_x$ that has been refused;
- $Alternative_x(Citeria_1, Criteria_2, \ldots, Criteria_n)$ as the alternative$_x$ with a set of criteria proved by a set of arguments $Argument_1, Argument_2, \ldots, Argument_n$ in the public knowledge of the participant.
- $PublicKnowledge_{AgP2}$ as the public knowledge of AgP_2;
- $UPrefAlternative_x$ is the utility of the preferred alternative$_x$ being in the range [0..1].

Begin

Foreach $Argument_i \in Alternative_x$ **do**

If $((Argument_i \notin PublicKnowledge_{AgP2}) \;||\; (\neg Argument_i \in PublicKnowledge_{AgP2}))$ **then**

$$Max\_Reward \leftarrow Max\_Reward + \frac{WArgument_i}{\sum_{j=1..n} WArgument_j}$$

End if

End for
$Max\_Reward \leftarrow Max\_Reward * UPrefAlternative_x$
End

Step 3: Offer the reward

The step of the offering reward consists in an iterative algorithm that sends promises of rewards containing the maximum reward value calculated in the previous step as a limit in order to make incremental reward offers to the opponent. The purpose of this algorithm its to offer the minimum possible reward to a selected opponent because when there is no information about some criteria of an alternative it is not possible to know which arguments are in favor, neutral or against. The algorithm of the reward offers is presented next. Consider:

- *NOfUnknownArgs* – number of unknown arguments for the alternative criteria;
- $Alternative_x$ – the $Alternative_x$ is the alternative for which acceptance is requested;
- *Justification* – arguments that justify the promise. This parameter can be empty;
- *Tolerance%* - percentage value of the reward that can be increased;
- *RequestAccepted* – can be true or false. If true the request sent was accepted if false the request was rejected.

Begin
$i \leftarrow 0$
Do
$$Request(AgP_1, AgP_2, Promisse(Justification, Alternative_x, \frac{Reward}{NOfUnkownArgs - i++}))$$
$while(i<NOfUnknownArgs)\ \&\&\ (\neg RequestAccepted))$
$if\ ((\neg RequestAccepted)\ then$
$$Request(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward_x * (1 + tolerance\%)))$$
end if
End

4.2 Process Of Reward Evaluation

The reward evaluation is the responsibility of AgP_2 who must analyze the requests received to see if the proposition is satisfactory. Bellow there is an example definition of a request with a reward proposition:

$$Request(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$$

After the receipt of this request, the following algorithm is executed to evaluate it, so that it may be accepted or rejected. In case of a rejection AgP_1 can propose another reward.

Begin
    Foreach $Argument_i \in Alternative_x$ ***do***
        If $((Argument_i \notin PublicKnowledge_{AgP2}) \; || \; (\neg Argument_i \in PublicKnowledge_{AgP2}))$ ***then***

$$Pretended\_Reward \leftarrow Pretended\_Reward + \frac{WArgument_i}{\sum_{j=1..n} WArgument_j}$$

        End if
    End for
    $Pretended\_Reward \leftarrow Pretended\_Reward * (1 - WPrefAlternative_x^{-1})$
    If $Reward \geq Pretended\_Reward$ ***then***
        $Accept(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$
    Else
        $Refuse(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$
    End if
End

Where *WPrefAlternative* is a value between [0..1] that represents the interest of the participant in that particular alternative.

5 Conclusion

In this work an agent-based architecture is proposed to support a group decision support system that considers persuasive argumentation between participants. In the persuasive argumentation it is possible to offer rewards to other participants of the meeting that can be claimed in the future. The reward mechanism defined in this paper is divided in two parts: the process of reward offering and the process of reward evaluation. In terms of the reward offering process we have created two algorithms, one to evaluate the maximum reward possible to offer to a specific opponent and another one to make the offer of the reward. In terms of the reward evaluation process we have also created and algorithm to evaluate the offered reward in order to verify if it is a fair and interesting offer according to the profile of the participant.

Future work includes the introduction of the reputation of the meeting participants in the rewards mechanism.

References

1. G. Marreiros, C. Ramos, J. Neves. "Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision", Lecture Notes in Computer Science, 3823 pp. 41-50, 2005.
2. Amgoud, L.; J. Bonnefon and H. Prade, (2005). An Argumentation-Based Approach to Multiple Criteria Decision. Lecture Notes in Computer Science, 3571, pp. 269 – 280.
3. Parsons, S. ; Sierra, C.; Jennings, N.R. – Agents that reason and negotiate by arguing. Journal of Logic and Computation, vol. 8 n. 3 pp 261-292, 1998.
4. Kraus, S.; K Sycara and A. Evenchick (1998). Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104(1-2) pp 1-69.
5. Jennings, N.; S. Parsons; P. Noriega and C. Sierra (1998). On Argumentation-Based Negotiation. Proceedings of the International Workshop on Multi-Agent Systems, USA.
6. Sierra, C.; Jennings, N.R.; Noriega,P.; Parsons, S. (1998). A framework for argumentation-based negotiation. Intelligent agents IV LNAI, Vol.1365 pp. 177-192.
7. G. Marreiros, R. Santos, C. Ramos, J. Neves, P. Novais, J. Machado, J. Bulas-Cruz, "Ambient Intelligence in Emotion Based Ubiquitous Decision Making", Proc. Artificial Intelligence Techniques for Ambient Intelligence, IJCAI'07 – Twentieth International Joint Conference on Artificial Intelligence. Hyderabad, India, 2007.
8. W. Zachary and J. Ryder, "Decision Support Systems: Integrating Decision Aiding and Decision Training". In: Handbook Of Human-Computer Interaction, pp. 1235-1258. The Netherlands: Elsevier Science, 1997.
9. P. Davidsson, "Multi agent based simulation: beyond social simulation", Proc. of the Second international Workshop on Multi-Agent Based Simulation. S. Moss and P. Davidsson, Eds. Springer-Verlag NJ, pp. 97-107, 2001.
10. F. Andrade, J. Neves, P. Novais, J. Machado, A. Abelha, "Legal Security and Credibility in Agent Based Virtual Enterprises, in Collaborative Networks and Their Breeding Environments", Springer-Verlag, pp 501-512, 2005.
11. R. Santos, G. Marreiros, C. Ramos, J. Neves, J. Bulas-Cruz. "Multi-agent Approach for Ubiquitous Group Decision Support Involving Emotions". Lectures Notes for Computer Science, 4159, pp. 1174 – 1185, 2006.
12. D. O'Keefe, "Persuasion: Theory and Research", SAGE Publications, 1990.
13. D. Pruitt.,"Negotiation Behavior", Academic Press, New York, N.Y., 1981.
14. G. Salancik, J. Pfeffer, "Who Gets Power – And how they hold on to it- A Strategic Contingency Model of Power. Organizational Dynamics, Vol. 5, pp. 3-21, 1977.
15. G. Marreiros, C. Ramos e J. Neves. "Multi-Agent Approach to Group Decision Making through Persuasive Argumentation". Proc. 6th International Conference on Argumentation (ISSA'06), Amesterdam, 2006.