Introducing Dynamic Argumentation to UbiGDSS

João Carneiro[1,2], Diogo Martinho[1], Goreti Marreiros[1], and Paulo Novais[2]

[1] GECAD, Institute of Engineering – Polytechnic of Porto, Porto, Portugal
{jomrc,1090557,mgt}@isep.ipp.pt
[2] ALGORITMI Centre, at University of Minho, Braga, Portugal
pjon@di.uminho.pt

Abstract. Supporting and representing the group decision-making process is a complex task that requires very specific characteristics. The current existing argumentation models cannot make good use of all the advantages inherent to group decision-making. There is no monitoring of the process or the possibility to provide dynamism to it. These issues can compromise the success of Group Decision Support Systems if those systems are not able to provide freedom and all necessary mechanisms to the decision-maker. We investigate the use of argumentation in a completely new perspective that will allow for a mutual understanding between agents and decision-makers. Besides this, our proposal allows to define an agent not only according to the preferences of the decision-maker but also according to his interests towards the decision-making process. We show that our definition respects the requirements that are essential for groups to interact without limitations and that can take advantage of those interactions to create valuable knowledge to support more and better.

Keywords: Argumentation, Automatic Negotiation, Multi-Agent Systems

1 Introduction

The UbiGDSS support the decision-making process by using the main characteristics of ubiquity ("anytime and "anywhere") [1, 2]. In order to support the decision-making process in an ubiquitous context it makes sense that UbiGDSS should: allow automatic negotiation, represent interests of decision-makers, allow the existence of a process, generate ideas, discuss points of view, etc [3, 4]. However, something is wrong with GDSS that we know today. Everyone has acknowledged the benefit of this type of systems; however, we do not actually see these systems being used in reality and it is not because the concept is still fresh. In fact, it is impossible to identify the reasons that lead to their absence and why they are not accepted in the industry sector nowadays. On the other hand, what we know is that most of artificial intelligence techniques proposed in the literature that could be used in UbiGDSS go against what are considered to be the benefits of group decision-making [5-7].

In this work, we propose a refreshing look into the concept of what and how should the artificial intelligence mechanism that composes an UbiGDSS be. For that, we in-

troduce a dynamic argumentation framework, which provides the system with the features that are necessary for decision-makers to enjoy the benefits of group decision-making. Our proposal intends to follow decision-makers throughout the entire process. Our approach allows a group of decision-makers, where each agent represents a decision-maker, to seek a possible solution to a problem (choosing between several alternatives) while taking into account all the preferences of the decision-makers. Besides this, and considering that the decision-makers can understand the conversation performed between agents, those agents will also be able understand the new arguments that are created and exchanged between decision-makers. These new arguments can be processed and used by agents not only to advice decision-makers, but also to find solutions throughout the decision-making process.

The rest of the paper is organized as follows: in the Section 2 our approach is presented and the definition of the argumentation model is described. In Section 3, we present some conclusions and the work to be done hereafter.

2 The Argumentation Model

In this paper, we consider the following structure of a decision problem: there are a set of possible alternatives A, a set of criteria C, and a set of agents Ag, such that an alternative $a \in A$ has a value for all the defined criteria C. The decision problem has a defined communication language $\mathcal{L}_c$ which allow agents Ag to communicate. In order to operate with the defined $\mathcal{L}_c$, there is a set of algorithms $\mathcal{L}_a$, which specify for each locution $\varphi \in \mathcal{L}_c$ its effect. The relations between alternatives, criteria, agents, communication language and algorithms jointly form a decision system, represented as follows:

Definition 1: A decision system $(C, A, Ag, \mathcal{L}_c, \mathcal{L}_a)$, consists of:

— a set of criteria $C = \{c_1, c_2, \dots, c_n\}, n > 0$;
— a set of alternatives $A = \{a_1, a_2, \dots, a_m\}, m > 0$;
— a set of agents $Ag = \{ag_1, ag_2, \dots, ag_k\}, k > 0$;
— a communication language $\mathcal{L}_c$, consisting of a set of all locutions;
— a set of algorithms working as regulation $\mathcal{L}_a$ for $\mathcal{L}_c$, specifying for each locution $\varphi \in \mathcal{L}_c$ its effects.

Rule 1: Each alternative is related with each criterion. There cannot be an existing alternative with values for criteria that is not considered in the problem.

Definition 2: A criterion $c_i = \{id_{c_i}, v_{c_i}, m_{c_i}\}$ consists of:

— $\forall\, c_i \in C, i \in \{1, 2, \dots, n\}$;
— id_{c_i} is the identification of a particular criterion;
— v_{c_i} is the value of a particular criterion (Numeric, Boolean or Classificatory);
— m_{c_i} is the greatness associated with the criterion (Maximization, Minimization, Positivity, Negativity and Without Value).

Definition 3: An alternative $a_i = \{id_{a_i}, [c_{1_{a_i}}, c_{2_{a_1}}, \ldots, c_{n_{a_i}}]\}$ consists of:

— $\forall a_i \in A, i \in \{1, 2, \ldots, n\}$;
— id_{a_i} is the identification of a particular alternative;
— $[c_{1_{a_i}}, c_{2_{a_i}}, \ldots, c_{n_{a_i}}]$ is the instantiation of each criterion.

The way each criterion is defined, allows an agent to know (in the previous example) that $c_{1_{a_1}} > c_{1_{a_2}} \wedge c_{1_{a_1}} > c_{1_{a_3}} \wedge c_{1_{a_2}} > c_{1_{a_3}}$, and $c_{2_{a_1}} \neq c_{2_{a_2}} \wedge c_{2_{a_1}} \neq c_{2_{a_3}} \wedge c_{2_{a_2}} = c_{2_{a_3}}$ and $c_{3_{a_2}} > c_{3_{a_1}} \wedge c_{3_{a_2}} > c_{3_{a_3}} \wedge c_{3_{a_1}} = c_{3_{a_3}}$.

An agent has a special structure that allows him to act according to the interests of the decision-maker he represents. Besides the agent's identification code, the decision-maker can also define the agent's style of behaviour for a certain time interval. The decision-maker may change that style of behaviour whenever he thinks to be appropriate. The proposed styles of behaviour (previously defined in [8]) allow the agent to act according to 4 dimensions: activity, resistance to change, concern for other and concern for self. Agents also include a protocol where it is defined a set of locutions available to ag_i in a time instant of t. An agent also holds the information about the evaluation done by the decision-maker about the preference for each alternative and the importance given to each considered criterion.

Definition 4: An agent $ag_i = \{id_{ag_i}, uid_{ag_i}, \beta_{ag_i}, Pr_{ag_i}, C_{ag_i}, A_{ag_i}, O_{ag_i}, K_{ag_i}\}$ consists of:

— $\forall ag_i \in Ag, i \in \{1, 2, \ldots, n\}$;
— id_{ag_i} is the identification of a particular agent;
— uid_{ag_i} is the identification of the decision-maker represented by the agent ag_i;
— β_{ag_i} is the agent's behaviour (Dominating, Compromising, Obliging, Integrating, Avoiding and No Style);
— Pr_{ag_i} is the agent's protocol for $\mathcal{L}_c$, specifying the 'legal' moves at each instant. A protocol on $\mathcal{L}_c$ is a set of locution available to ag_i, where $Pr_{ag_i} \subseteq \mathcal{L}_c$;
— C_{ag_i} is the agent's evaluation of each criterion, $C_{ag_i} = \{E_{c_1}, E_{c_2}, \ldots, E_{c_n}\}, Ev_{c_i} \in \{[0,1], \bot\}$;
— A_{ag_i} is the agent's evaluation of each alternative, $A_{ag_i} = \{E_{a_1}, E_{a_2}, \ldots, E_{a_n}\}, Ev_{a_i} \in \{[0,1], \bot\}$;
— O_{ag_i} is the set of agent's objectives, $O_{ag_i} \subseteq A \cup C$, *preference relation* $\geq$ *on the set* O_{ag_i};
— K_{ag_i} is the agent's knowledge, where he can access the list of all sent and received messages, as well as the preferences of other agents, according to the knowledge he possess in a certain time instant of t.

Definition 5: A behaviour $\beta_i = \{Rc_{\beta_i}, Al_{\beta_i}, Cs_{\beta_i}, Co_{\beta_i}\}$ consists of (according to [8]):

— Rc_{β_i} is the agent's resistance to change dimension value;
— Al_{β_i} is agent's activity level dimension value;

— Cs_{β_i} is the agent's concern for self dimension value;
— Co_{β_i} is the agent's concern for others dimension value.

Definition 6: A locution $\varphi_i = \{id_{\varphi_i}, Tp_{\varphi_i}, Tx_{\varphi_i}, Ct_{\varphi_i}, Vr_{\varphi_i}, Dr_{\varphi_i}, Dm_{\varphi_i}, Av_{\varphi_i}\}$ consists of:

— $i \in \{1, 2, ..., n\}$;
— id_{φ_i} is the locution's id (unique);
— Tp_{φ_i} is the locution's type (Question, Statement and Request);
— Tx_{φ_i} is the text associated to the locution;
— Ct_{φ_i} is the locution's context (Alternative, Criterion or Without Context);
— Vr_{φ_i} is the set of variables associated to the locution (Alternative or Criterion);
— Dr_{φ_i} is the direction associated to the locution (infavor, against, null);
— Dm_{φ_i} is the locution's domain (General or Specific);
— Av_{φ_i} is the locution's state (Available or Not Available).

Rule 2: Whenever a locution is added to Pr_{ag_i} in the time instant t its state will be $Av_{\varphi_i}{}^t, Av = Available$.

Rule 3: Whenever a locution is used at a time instant t its state will change to $Av_{\varphi_i}{}^t, Av = Not\ Available$.

Rule 4: Whenever a locution $Dm_{\varphi_i}, Dm = Specific$ is added to Pr_{ag_k} at the time instant t, then $\forall Dm_{\varphi_j} \in \mathcal{L}_{Dm} \subset Pr_{ag_k}, Dm = General$ and its state will be $Av_{\varphi_j}{}^t, Av = Available$.

Rule 5: For any locution $\varphi_j \in \mathcal{L}_{sp_{c_i}} \wedge c_i \subset Vr_{\varphi_j}$ there cannot be another locution φ_k where $c_i \subset Vr_{\varphi_k} \wedge \varphi_k \notin \mathcal{L}_{sp_{c_i}}$.

Rule 6: For any locution $\varphi_j \in \mathcal{L}_{sp_{a_i}} \wedge a_i \subset Vr_{\varphi_j}$ there cannot be another locution φ_k where $a_i \subset Vr_{\varphi_k} \wedge \varphi_k \notin \mathcal{L}_{sp_{a_i}}$.

Definition 7: A message $\psi_i = \{id_{\psi_i}, \varphi_{\psi_i}, tr_{\psi_i}, \alpha_{\psi_i}, en_{s_{\psi_i}}, En_{r_{\psi_i}}\}$ consists of:

— $i \in \{1, 2, ..., n\}$;
— id_{ψ_i} is the conversation code;
— tr_{ψ_i} is the target associated with the message (can be null or be another message);
— φ_{ψ_i} is the locution sent in the message;
— α_{ψ_i} is the justification associated to the locution (can be an argument or can be null);
— $en_{s_{\psi_i}}$ is the agent/user who sent the message;
— $En_{r_{\psi_i}}$ is the set of agents/users who will receive the message (can be 1 or *).

Rule 7: For any message ψ created by a decision-maker $Dr_{\varphi_\psi}, Dr \neq null$. This means that the message's locution can only be either infavor or against Vr_{φ_ψ}.

Definition 8: An argument $\alpha_i = \{id_{\alpha_i}, Tx_{\alpha_i}, Vr_{\alpha_i}\}$ consists of:

— $i \in \{1, 2, \dots, n\}$;
— id_{α_i} is the identification of a particular argument;
— Tx_{α_i} is the text associated to a particular argument;
— Vr_{α_i} is the set of variables associated to a particular argument (can contain alternatives and criteria).

Up until now we have presented the definition for the proposed argumentation model only considering the agents' point of view. From this point forward all the definitions presented will be directed towards to the interactions between decision-makers. However these definitions are an extent of what has been proposed so far which will allow both agents and humans to use the same model definition.

Decision-makers, compared to agents, can also create messages, and those messages will hold the new knowledge as well as a new set of dynamic arguments. Our approach allows decision-makers to create messages that include arguments that are infavor or against. Each created message may lead to $n - 1$ messages, where n is the number of decision-makers involved in the process. Decision-makers can argue against a message through the use of an attack or support and argue in favour of a message through the use of reinforcement. The way the model is defined allows the message content to be of any sort of format like text or voice, since that information is irrelevant to the agent. A decision-maker may evaluate messages sent by other decision-makers, and this will allow agents to understand human interactions and the impact of every conversation.

Example 1 (message pro): $Dr_{\varphi_\psi}, Dr = infavor$. That means something positive related to Vr_{φ_ψ}. Such message is said pro the Vr_{φ_ψ}. For our previous example a message ψ pro could have $Vr_{\varphi_\psi}, Vr = a_1 \wedge Tx_{\alpha_\psi}, Tx = $ "1st maintenance service is free".

Example 2 (message cons): $Dr_{\varphi_\psi}, Dr = agaist$. That means something negative related to Vr_{φ_ψ}. Such message is said cons the Vr_{φ_ψ}. For our previous example a message ψ cons could have $Vr_{\varphi_\psi}, Vr = c_3, c_{3_a} = no \wedge Tx_{\alpha_\psi}, Tx = $ "The high temperatures in our area will damage the product in a car without air conditioning".

Definition 9: Let $\Psi = \{\psi_1, \psi_2, \dots, \psi_n\}, n > 0$ denote a finite set of n messages that are exchanged during a human discussion.

Let us now define two functions that relate the messages in favour or against an $a_i \in A$, $c_j \in C$, $A_k \subseteq A$ or $C_l \subseteq C$ (Let us consider $X = a_i \vee c_j \vee A_k \vee C_l$):

— $F_{infavor} : X \to \forall \psi \in \Psi, Dr_{\varphi_\psi} = infavor$, is a function that returns the messages in favour of X. Such messages are said pro the X;

— $F_{against}: X \rightarrow \forall \psi \in \Psi, Dr_{\varphi_\psi} = against$, is a function that returns the messages against X. Such messages are said cons the X.

Rule 8: A message is either in favour of the X or against the X. It cannot be both, so: $x \in X, \forall \psi \in \Psi s.t. \neg(\psi \in F_{infavor}(x) \wedge \psi \in F_{against}(x))$.

Definition 10: A message evaluation $\xi = \{en_{\xi_i}, \psi_{\xi_i}, ev_{\xi_i}\}$ consists of:

— $i \in \{1, 2, ..., n\}$;
— en_{ξ_i} is the user who performed the evaluation;
— ψ_{ξ_i} is the message being evaluated;
— ev_{ξ_i} is the evaluation value [-1, 1].

Definition 11: Let $\Xi = \{\xi_1, \xi_2, ..., \xi_n\}$ denote a finite set of n evaluations that are made during a human discussion.

Let us now define two functions that return all the evaluations approving or disapproving a message ψ_i:

— $F_{approval}: \psi_i \rightarrow \forall \xi \in \Xi, \psi_\xi = \psi_i \wedge ev_{\xi_i} > 0$, is a function that returns the messages approving ψ_i;
— $F_{disapproval}: \psi_i \rightarrow \forall \xi \in \Xi, \psi_\xi = \psi_i \wedge ev_{\xi_i} < 0$, is a function that returns the messages disapproving ψ_i.

Let us now define two functions that relate a message ψ_1 to the messages reinforcing it and to the messages attacking it:

— $F_{reinforcement}: \psi_1 \rightarrow \forall \psi \in \Psi, tr_\psi = \psi_1 \cap F_{approval}(\psi_1)$, is a function that returns the messages reinforcing ψ_1. Such messages are said pro the message ψ_1;
— $F_{attack}: \psi_1 \rightarrow \forall \psi \in \Psi, tr_\psi = \psi_1 \cap F_{disapproval}(\psi_1)$, is a function that returns the messages attacking ψ_1. Such messages are said cons the message ψ_1.

Let us define a function which returns the messages sent by en_1:

— $F_{sentby}: en_1 \rightarrow \forall \psi \in \Psi, en_\psi = en_1$, is a function that returns the messages sent by en_1.

Let us define a function that returns all the evaluation done by en_1:

— $F_{evaluatedby}: en_1 \rightarrow \forall \xi \in \Xi, en_\xi = en_1$, is a function that returns the evaluations done by en_1.

Let us now define a function to identify the lasts messages sent during a human discussion:

— $F_{last}: \Psi \rightarrow \forall \psi \in \Psi, F_{reinforcement}(\psi) \cup F_{attack}(\psi) = \emptyset$, is a function that returns the messages that are not attacked or reinforced thus being the last messages exchanged in a human discussion.

Let us now define a function that returns all the messages done until message ψ_1 which can be more than one reinforcement or just only one attack:

$$— F_{reinforcementrow}: \psi_1 \rightarrow \forall \psi \in \Psi, \begin{cases} F_{reinforcementrow}(\psi), \psi \in F_{reinforcement}(\psi_1) \\ \psi_1, \psi \in F_{attack}(\psi_1) \\ \psi_1, \psi \notin F_{attack}(\psi_1) \vee F_{reinforcement}(\psi_1) \end{cases},$$

is a function that returns all the reinforcement or the attack done until ψ_1.

Let us now define a function that returns all the messages prior to each last message in a human discussion that are either reinforcements (in the limit it could be all reinforcements until the initial message) or an attack:

— $F_{lastdiscussions}: \Psi \rightarrow \forall \psi \in F_{last}, F_{reinforcementrow}(\psi)$, is a function that returns all the messages prior to each last message that are either reinforcements or an attack.

Let us define a function that returns all the last evaluations done by en_1:

— $F_{lastevaluatedby}: en_1 \rightarrow \forall \xi \in F_{evaluatedby}(en_1), \psi_\xi \in F_{lastdiscussions}$, is a function that returns all the last evaluations done by en_1.

In order to identify all the last evaluations done by en_1 in a certain dialogue, the id_ψ can be used to filter those evaluations, and therefore the function which returns all the last evaluations done by en_1 in a dialogue with id_{ψ_1} would be:

— $F_{lastevaluationsin}: id_{\psi_1} \rightarrow \forall \xi \in F_{lastevaluatedby}(en_1), id_{\psi_\xi} = id_{\psi_1}$, is a function that returns all the last evaluations done by en_1 for the human dialogue with the id id_{ψ_1}.

In [9] the researchers state that argumentation-based decision process can be decomposed into the following steps: (1) Constructing arguments in favour/against statements (beliefs or decisions), (2) Evaluating the strength of each argument, (3) Determining the different conflicts among arguments, (4) Evaluating the acceptability of arguments and (5) Comparing decisions on the basis of relevant "accepted" arguments. We consider that in our work, we deal with the first 3 points, with the advantage of integrating those points in a perspective that deals with both agents and humans using the same definition. Points 4 and 5 frame what we intend to do as future work and that is described in Section 4.

3   Conclusions and Future Work

In this work, we introduced the possibility to create dynamic arguments through a definition of an argumentation model. With our proposal it is possible (and motivates) the interaction between decision-makers, allowing a decision-maker to create new arguments (in favour or against) and also reinforce or attack other arguments created by other decision-makers. The way decision-makers can evaluate each argument will allow the agents to understand the impact of the interactions for the decision-maker they represent and not the content of the conversation (which could be text, sound, etc.).

Besides this, since agents share the same problem definition, it allows decision-makers to understand all the interactions between those agents and the reason why they suggest a certain solution for the problem.

As future work, we intend to design all the necessary algorithms so that our proposal can be implemented. One major goal is to work in an algorithm that allows the agent to understand each message decision tree that is associated with each interaction between decision-makers. Besides this, the agent should also take his defined behaviour into account when doing any analysis in order to properly support and defend the perspective of the decision-maker he represents. Finally we intend to work on the branch of how and what type of information should be presented to the decision-maker, through what will call as intelligent reports.

4   Acknowledgements

References

1. Kwon, O., Yoo, K., Suh, E.: UbiDSS: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. Expert systems with applications 28, 149-161 (2005)
2. Daume, S., Robertson, D.: An architecture for the deployment of mobile decision support systems. Expert Systems with Applications 19, 305-318 (2000)
3. Carneiro, J., Santos, R., Marreiros, G., Novais, P.: UbiGDSS: A Theoretical Model to Predict Decision-Makers' Satisfaction. International Journal of Multimedia and Ubiquitous Engineering 10, 191-200 (2015)
4. Carneiro, J., Santos, R., Marreiros, G., Novais, P.: Overcoming the Lack of Human-Interaction in Ubiquitous Group Decision Support Systems. (2014)
5. Dennis, A.R.: Information exchange and use in small group decision making. Small Group Research 27, 532-550 (1996)
6. Huber, G.P.: Issues in the design of group decision support sytems. MIS quarterly 195-204 (1984)
7. Hill, G.W.: Group versus individual performance: Are N+ 1 heads better than one? Psychological bulletin 91, 517 (1982)
8. Martinho, D., Carneiro, J., Marreiros, G., Novais, P.: Dealing with Agents' Behaviour in the Decision-Making Process. In: Workshop Proceedings of the 11th International Conference on Intelligent Environments, pp. 4. IOS Press, (2015)
9. Rahwan, I., Simari, G.R., van Benthem, J.: Argumentation in artificial intelligence. Springer (2009)