

A Multi-Agent System Framework for Dialogue Games in the Group Decision-Making Context

João Carneiro¹[0000-0003-1430-5465], Patrícia Alves¹[0000-0003-3997-311X], Goreti Marreiros¹[0000-0003-4417-8401] and Paulo Novais²[0000-0002-3549-0754]

¹ GECAD – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering, Polytechnic of Porto, 4200-072 Porto, Portugal

{jrc,prjaa,mgt}@isep.ipp.pt

² ALGORITMI Centre, University of Minho, Guimarães 4800-058, Portugal
pjon@di.uminho.pt

Abstract. Dialogue games have been applied to various contexts in computer science and artificial intelligence, particularly to define interactions between autonomous software agents. However, in order to implement dialogue games, the developers need to deal with other important details besides what is presented in the model's definition. This is a complex work, mostly when it is expected that the agents' interactions correctly represent a human group behavior. In this work, we present a multi-agent system framework specifically designed to facilitate the implementation of dialogue games under the context of group decision-making in which agents interact as the humans do in face-to-face meetings. The proposed framework, named MAS4GDM, encapsulates the JADE framework and provides a layer that allows developers to easily implement their dialogue models without being concerned with some complex implementation details, such as: the communication model, the agents' life cycle, among others. We ran an experimental evaluation and verified that the proposed framework allows to implement dialogue models in an easier way and abstract the developers from important implementation details that can compromise the application's success.

Keywords: Multi-Agent Systems, Dialogue Games, Group Decision-Making, JADE.

1 Introduction

The ultimate goal of a group decision-making process is to select one or more alternatives as the solution for a certain problem [1]. However, due to the problems self-nature, and considering it involves several decision-makers with different preferences and beliefs, it is a complex process to deal with. In the last decades, different technological mechanisms have been proposed to help decision-makers in the group decision-making processes [2]. One well-known strategy consists in using autonomous software agents to represent decision-makers in terms of their preferences, beliefs and interactions [3, 4]. The use of a multi-agent system in the group decision-

making context facilitates the representation of the several entities involved in the process, as well as benefits from the agents' characteristics (autonomy, pro-activity, etc.) in important tasks, such as reasoning and communication [5]. In order to make agents capable of communicating and interacting with each other, the researchers have been developing formal models of several dialogue types [6, 7]. The deliberation dialogue type is one of the six primary dialogue types identified by Walton and Krabbe [8] and represents the scenario where participants collaborate to reach a decision [9]. Some of these models tend to represent the communication performed by the decision-makers in face-to-face scenarios where the decision evolves over time. However, in real face-to-face scenarios when a decision-maker speaks, all other decision-makers receive the message at the same time, which is not the case of the communication performed by the agents. When an agent sends a message to all other agents, the message is received in different time instants by each of them. This time difference, even extremely small (milliseconds), is enough to make agents act in the possession of different amount of knowledge, which can compromise the quality of the decision made. In addition, this technological limitation can deceive researchers in the dialogue models' evaluation phase.

In this work, we present the Multi-Agent System for Group Decision-Making (MAS4GDM) framework, which aims to facilitate the implementation of dialogue game models that somehow intend to simulate or virtualize the interaction of groups of decision-makers in face-to-face contexts. The MAS4GDM framework allows programmers to focus only on the implementation of dialogue models and to abstract themselves from other complex implementation aspects, such as: managing the agents lifecycle and their ids, and system failures due to incomplete or incorrect messages. The framework characteristics help programmers in the definition of the agent's behavior, as well as to access information. In addition, the communication flow that guarantees security, integrity and that agents always act in the possession of the same knowledge, is of the framework's responsibility.

The rest of the paper is organized in the following order: in the next section, the MAS4GDM framework is described and in Section 3 we can find the experimental evaluation. In Section 4 the discussion is presented. Finally, in Section 5 some conclusions are taken along with the work to be done hereafter.

2 MAS4GDM

MAS4GDM is a framework that allows the implementation of dialogue models that intend to simulate/represent/virtualize the type of interaction practiced by decision-makers in face-to-face group decision-making processes. The MAS4GDM framework encapsulates the JADE framework, implements a communication model that virtualizes a face-to-face context and allows the developers to focus only on the implementation of the agents' behavior. In addition to the communication model, the MAS4GDM framework creates the agents automatically, manages their ids, prevents the system from failing due to incomplete or incorrect messages, and assists the developers in defining the agents' behavior as well as to access information.

2.1 Architecture

In Fig. 1, the “*external Application*” represents an application using the MAS4GDM framework. The layer provided by the framework for the developers consists on the following classes: *AbstractMeeting*, *AbstractDecisionMaker* and *Message*.

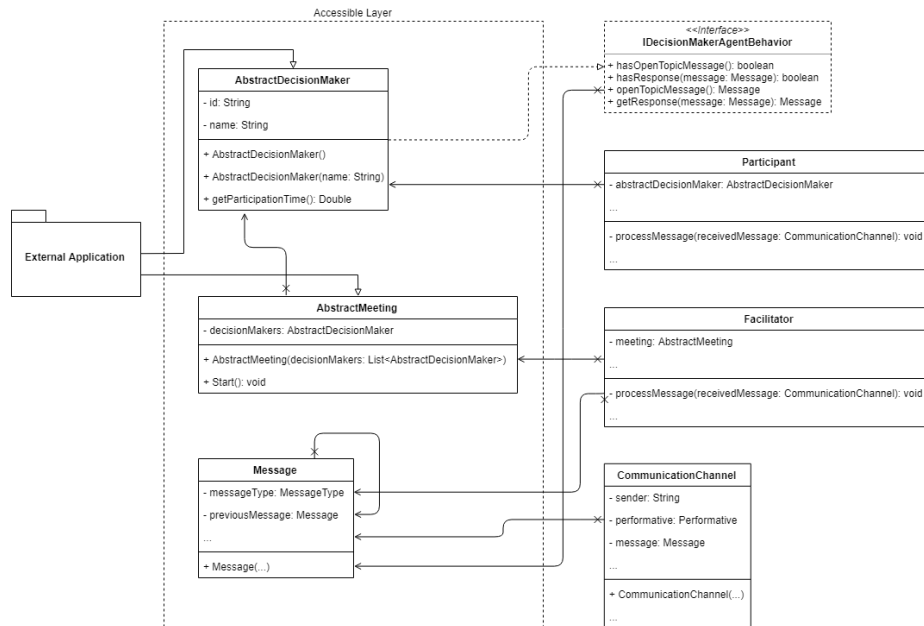


Fig. 1. Short version of the MAS4GDM framework’s class diagram.

These are the 3 classes that the developers can extend and that allow them to implement their business:

- *AbstractMeeting*: From the programmer’s point of view, this class is the heart of the framework, since it implements the “*start()*” method that allows to start the whole process. It would not be possible to start a group decision-making process without a “Meeting”. For a *Meeting* object to be created, it is necessary to send a list of *AbstractDecisionMaker* as an argument, since there are no group decision processes without participants;
- *AbstractDecisionMaker*: This class represents a decision-maker. The developers can create as many extensions of *AbstractDecisionMaker* as the types of decision-makers they wish to represent. The fact that it is possible to create several types of decision-makers, allows the agents representing each of those decision-makers to make use not only of different levels of knowledge but also to implement different behaviors. This class implements the *IDecisionMakerAgentBehavior* interface, which means that any *AbstractDecisionMaker* that the programmer defines will always know how to execute the 4 core behaviors of the framework:

- *boolean hasOpenTopicMessage()*: The agent checks if it is interested in starting a new topic;
- *Message getOpenTopicMessage()*: The agent returns a message to start a new topic;
- *boolean hasResponse()*: The agent checks if it is interested in responding to any message received in the current topic;
- *Message getResponse()*: The agent returns a message to respond to a previous message from the topic that is open.
- *Message*: This class represents the messages that are exchanged by the agents from the programmer's point of view.

As agents exchange messages of type *Message* and all messages are encapsulated in the *CommunicationChannel* of the framework, there are never any errors that affect the correct functioning of the framework derived from flaws in the construction of objects of type *Message*, such as: incorrect definition of the sender and receivers, or at the limit, even if the object is null.

2.2 Agents

The management of the agents' lifecycle is of the MAS4GDM framework complete responsibility. Thus, with the use of the MAS4GDM framework, developers have no need to manage/create agents.

The MAS4GDM framework considers the existence of two types of agents:

- **Participant Agent**: Each participant agent represents a decision-maker. It means that the number of agents involved will be equal to the number of decision-makers who are part of a decision-making process;
- **Facilitator Agent**: There is a facilitator agent for each decision process being executed. The facilitator agent is responsible for managing the communication process and creating the participant agents in the same *AgentContainer* in which it was created.

Whenever a decision process is initiated (*AbstractMeeting.start()*) a Facilitator Agent is created, receiving the *AbstractMeeting* and the *AgentContainer* in which it was created as arguments. In turn, the Facilitator Agent uses the *AbstractDecisionMaker* list that is contained in the *AbstractMeeting* to create the Participant Agents (in the *AgentContainer* that it received as argument). For each *AbstractDecisionMaker* in the list, a Participant Agent is created. Each Participant Agent receives as an argument the *AbstractDecisionMaker* that it will represent and where the behaviors, defined in the *IDecisionMakerAgentBehavior*, are implemented. When the process ends, the Facilitator Agent notifies the Participant Agents and after they respond with an acknowledgment, they terminate. When all the Participant Agents have responded, the Facilitator Agent also terminates.

2.3 Communication

One of the most important components of this framework is the virtualization by the communication model used by agents of the type of interaction between real decision-makers in face-to-face contexts. The type of communication implemented in this framework guarantees that all agents always act in possession of the same knowledge (when it is public). In addition, the framework is prepared to deal with agents that may have different levels of activity (different probability of starting dialogues or responding to a message), which means that it is possible to define agents that represent more or less “talking” styles. The framework does not oblige to implement any hard communication format, as is the case of “question-answer” models. On the contrary, it allows complete freedom in the implementation of the type of dialogue to be carried out by the agents. It is possible to implement dialogues that represent complex dialogues (via audio or writing) such as those performed on a social network like Facebook, where anyone can open a new topic and each message can successively originate several other messages, allowing at the same time to come up with successive answers on a certain subject and go back to resume a “reasoning”.

The MAS4GDM framework organizes dialogues by topics. Each topic is related to a specific subject; however, the subject’s detail level can be defined according to the model that is being implemented. Which means that in the limit, there can only be 1 topic. Each topic consists of an indefinite number of messages. The first message of a topic is the root message and has no previous messages. All the other messages in the topic are a response to any previous message, and as such, point to a previous message. The messages that constitute a topic are then represented in the form of a k-ary tree.

All messages exchanged by the agents are of the *CommunicationChannel* type. These messages may or may not carry messages of the type *Message*. Since all messages of type *Message* are encapsulated in the messages exchanged by the agents, the framework is immune to possible failures in the incorrect implementation of a dialogue model. All the messages exchanged by the agents have a defined performative:

- ENTER_DIALOGUE: ask about the interest in responding to some message of the topic being debated;
- ENTER_DIALOGUE_RESPONSE: Respond to ENTER_DIALOGUE with a participation time or null if not interested;
- OK: acknowledgment;
- OPEN_DIALOGUE: Ask about the interest in starting a new topic;
- OPEN_DIALOGUE_RESPONSE: reply to OPEN_DIALOGUE with a participation time or null if not interested;
- PROCESS_ENTER_DIALOGUE: notify the selected agent of the right to communicate a new message about the topic being debated;
- PROCESS_ENTER_DIALOGUE_RESPONSE: reply to PROCESS_ENTER_DIALOGUE with a Message or null;
- PROCESS_OPEN_DIALOGUE: notify the selected agent of the right to start a new topic;
- PROCESS_OPEN_DIALOGUE_RESPONSE: reply to PROCESS_OPEN_DIALOGUE with a Message or null;

- PROCESS_OVER: notify that the process has finished;
- READY: notify READY state.

Fig. 2 represents (in a non-formal format) the flow of communication between the Facilitator Agent and the Participant Agents. As noted above, the Facilitator Agent is responsible for creating the Participant Agents. Whenever a Participant Agent is created, it sends a message to the Facilitator Agent with the READY performative. When the Facilitator Agent receives READY from all Participant Agents, the process is ready to begin. To initiate the process, the Facilitator Agent sends a message to the Participant Agents with the OPEN_DIALOGUE performative. All Participant Agents that are interested in starting a new topic respond with a participation time (by default, they return a value between [0; 1], otherwise they will reply null). The message with the participation time is then sent to the Facilitator Agent and has the OPEN_DIALOGUE_RESPONSE performative.

When the Facilitator Agent receives the message with the participation time of all Participant Agents, it selects the Participant Agent that sent the highest participation time to start a new topic (if it received null from all Participant Agents, it means that the decision process has ended, since the agents had anything else to add). After selecting the Participant Agent that has earned the right to start a new topic, the Facilitator Agent sends a message with the PROCESS_OPEN_DIALOGUE performative, notifying all Participant Agents of the Participant Agent who has gained the right to start a new topic (including the selected Participant Agent). Then, the selected Participant Agent sends a message to the Facilitator Agent with the content with which it wants to start a new topic, with the performative PROCESS_OPEN_DIALOGUE_RESPONSE. If at the moment that the Participant Agent is informed that he has gained the right to speak, he no longer has interest in starting a new topic, it may respond with a null Message object. In this case, when the Facilitator Agent verifies that the Message is null, it re-sends a Message with the OPEN_DIALOGUE performative to all Participant Agents. After receiving the message from the Participant Agent with the performative PROCESS_OPEN_DIALOGUE_RESPONSE, the Facilitator Agent broadcasts the Message to all Participant Agents using the performative ENTER_DIALOGUE.

This Message, in addition to disseminate knowledge by all Participant Agents, also serves to question them about their interest in responding to that or any other message of the topic being debated. When the Participant Agents receive the message with the ENTER_DIALOGUE performative, they check to see if they are interested in responding to the topic being debated and, once again, respond with a message containing the participation time (or null) and the ENTER_DIALOGUE_RESPONSE performative. When the Facilitator Agent receives the message with the performative ENTER_DIALOGUE_RESPONSE from all Participant Agents, it again selects the Participant Agent with the highest participation time (if all Participant Agents have sent null, it means that there is nothing else to add about the topic being debated and the Facilitator Agent re-sends a message with the OPEN_DIALOGUE performative to the Participant Agents). After selecting the Participant Agent that has earned the right to

respond, the Facilitator Agent sends a message to all Participant Agents with the performative `PROCESS_ENTER_DIALOGUE` informing them of the selected agent.

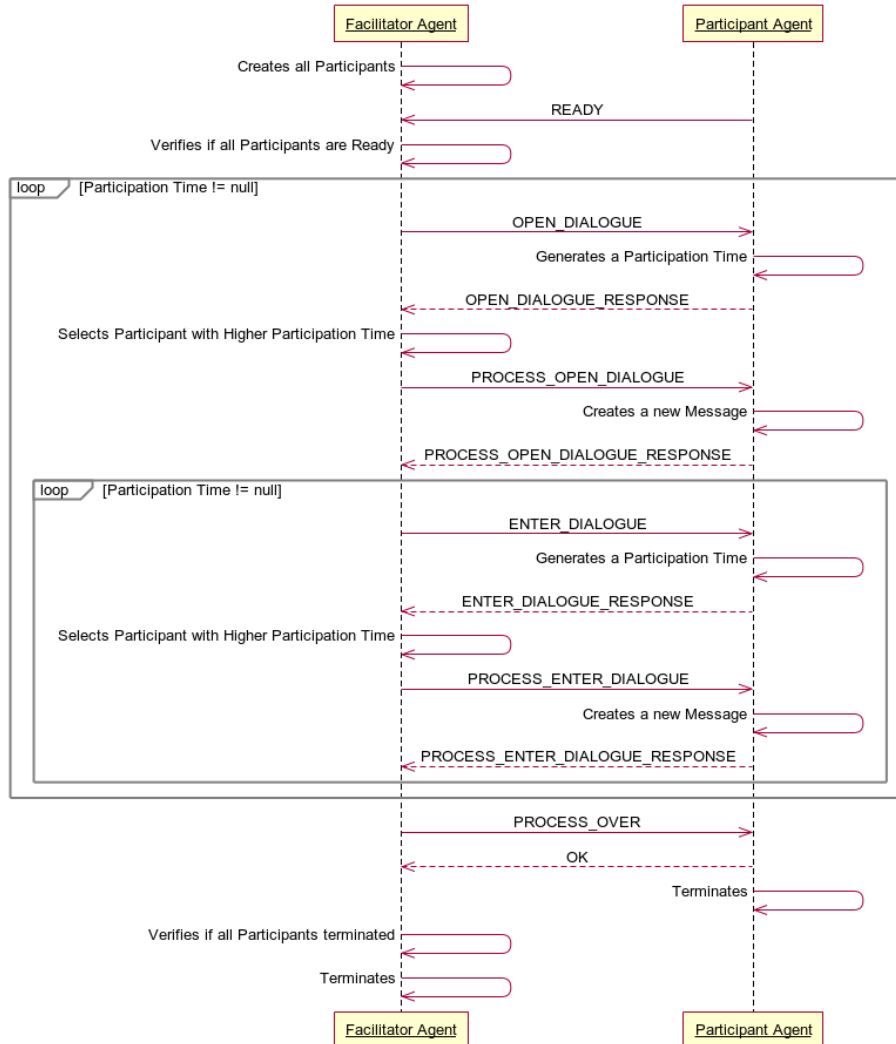


Fig. 2. Representation of the communication's flow between the Facilitator Agent and the Participant Agents.

Then, the selected Participant Agent sends a message to the Facilitator Agent with the `PROCESS_ENTER_DIALOGUE_RESPONSE` performative with the content with which it wants to respond (if at the moment the Participant Agent is informed that it has gained the right to speak but it is no longer interested in responding, it may respond with a null Message and the Facilitator Agent will resend a message to all agents with

the performative ENTER_DIALOGUE containing the last valid message it received). After replying to a message with the PROCESS_ENTER_DIALOGUE_RESPONSE performative, the Facilitator Agent re-sends a message with the performative ENTER_DIALOGUE to all Participant agents. Finally, when no Participant Agent wants to respond to the current topic or start a new topic, the process ends. The Facilitator Agent then sends a message with the PROCESS_OVER performative to all Participant Agents and they respond (before they finish) a message with the OK Performative. When the Facilitator Agent receives a message from all the Participant Agents with the OK performative, it also terminates.

3 Experimental Evaluation

In order to empirically evaluate the proposed framework, we implemented an argumentation-based dialogue model [10, 11] designed to the group decision-making context. In order to make the scenario more complex, the agents were defined with different social aspects: behavior styles, levels of expertise and credibility [12]. A decision scenario that consisted of the acquisition of 100 cars to renew the fleet of an organization was also defined (the considered multi-criterion problem can be found in Table 1). The existence of 5 decision-makers was considered, and preferences regarding the different criteria and alternatives were defined for each of them.

Table 1. Multi-criteria problem.

	Price	HP	Trunk	Consumption	Style	GPS	Color
Ford Fiesta	9900	70	292	3.5	2	True	Green
SEAT Ibiza	9900	75	355	3.8	2	False	Blue
Opel Corsa	8800	75	285	3.2	1	False	Red

The dialogue carried out by the agents was quite long and as such, only 4 of the more than 20 generated topics are presented below.

1. ag1: I agree with being in favor of Opel Corsa 1.4 90 Selective 5p S/S MTA because very good recent experience with their assistance service.
 - a. ag2: I agree.
 - b. ag3: I disagree.
 - c. ag4: I agree.
 - d. ag5: I disagree because, I think that argument can be applied to every brand.
 - (1) ag3: I agree.
 - (2) ag4: I agree.
 - (3) ag1: I agree.
 - (4) ag2: I disagree because, that is not true.
 - i. ag1: I do not have that information.
 - ii. ag3: I do not have that information.
 - iii. ag5: I do not have that information.

- iv. ag4: I disagree because, if you check car magazines you will see that the other brands also have good marks in terms of assistance service.
- (a) ag3: I agree.
 - (b) ag1: I disagree.
 - (c) ag5: I disagree because, that was verified on previous models. However, if you check the tests regarding the newest models, you will see that the difference is considerable.
 - A. ag2: I agree.
 - B. ag1: I agree.
 - C. ag4: I agree.
 - D. ag3: I agree.
 - (d) ag2: I disagree because, this particular brand has always the highest scores and more importantly we have a positive past experience with this brand.
 - A. ag5: I am not sure about that.
 - B. ag3: I disagree.
 - C. ag1: I do not have that information.
 - D. ag4: I agree.
2. ag5: Which criterion/a do you consider most important?
- a. ag3: For me the most important criterion/a is/are: consumption.
 - b. ag4: For me the most important criterion/a is/are: color.
 - c. ag2: For me the most important criterion/a is/are: price.
 - d. ag5: For me the most important criterion/a is/are: price.
 - e. ag1: For me the most important criterion/a is/are: consumption.
3. ag5: Hey ag1, do you accept this alternative as the solution? Opel Corsa 1.4 90 Selective 5p S/S MTA (Appeal to Self Interest)
- a. ag1: I accept.
4. ag5: Hey ag4, do you accept this alternative as the solution? Opel Corsa 1.4 90 Selective 5p S/S MTA (Appeal to Common Sense)
- a. ag4: I accept.

In this small excerpt, as in the full version of the dialogue, some agents were more active, which was due to the use of different styles of behavior (defined with different levels of activity). As we have seen previously, the MAS4GDM framework allows to define different probabilities of activity in the agents. In the first topic created by the agents it was verified that the agents “resumed a previous reasoning” when after having responded to the message iv - (c), an agent sent a message (d) to respond to the message iv. In each topic a different number of messages were exchanged according to the interest of the agents in sharing or not new knowledge, as well as in their interest in defending a certain idea.

4 Discussion

In the light of the current state of the art, the MAS4GDM framework provides the discussion of several important aspects. Next, we will see those we considered most

relevant to the context of this publication. It seems clear to us that although the context of the framework is that of group decision-making, it is possible to use it in other contexts. Any type of dialogue defined by Walton and Krabbe [8] can be implemented using the MAS4GDM framework. However, since this framework implements an acknowledgment system that guarantees that agents always behave in equality of knowledge, they tend to exchange more messages, which consequently affects performance, which is unnecessary (but possibly not relevant) in certain contexts.

Another relevant aspect is that, the MAS4GDM framework helps, even intuitively, in the definition of the dialogue game protocol as proposed by McBurney and Parsons [13]. The structure of communication flow that composes it, based on topics and responses not necessarily sequential (as verified in the experimental evaluation), facilitates the implementation of elements such as commencement rules, commitments termination rules.

5 Conclusions and Future Work

At this moment we can observe the MAS4GDM framework in terms of the perception the experimental evaluation transmitted to us. We concluded that the MAS4GDM framework greatly facilitates the implementation of dialogue models, since it abstracts the developers from a series of implementation details. In addition, it makes the code much more legible because it completely separates the intelligent layer (dialogue model's definition) from the communication layer and the agents' lifecycle management. The fact that the framework is responsible for the agents' lifecycle and communication flow facilitates the detection of implementation errors. As the MAS4GDM framework was designed to operate in the group decision-making's context, it allows to obtain richer outputs when compared to systems that aim to virtualize face-to-face scenarios and that do not include mechanisms that guarantee that all agents always act in equality of knowledge, and as a consequence, behaviors, interactions and outputs do not represent properly the conceptual definition of those models.

As future work, we want to transform the architecture of the MAS4GDM framework into a micro services architecture in order to make the framework completely independent. We also intend to conduct performance tests to study the behavior of the framework in complex scenarios that involve many agents exchanging a large volume of messages simultaneously.

Acknowledgments

This work was supported by the GrouPlanner Project (POCI-01-0145-FEDER-29178) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within the Projects UID/CEC/00319/2013 and UID/EEA/00760/2013.

References

1. Pérez, I.J., Cabrerizo, F.J., Alonso, S., Herrera-Viedma, E.: A new consensus model for group decision making problems with non-homogeneous experts. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, 494-498 (2014)
2. Alonso, S., Herrera-Viedma, E., Chiclana, F., Herrera, F.: A web based consensus support system for group decision making problems and incomplete preferences. *Information Sciences* 180, 4477-4495 (2010)
3. Groeneveld, J., Müller, B., Buchmann, C.M., Dressler, G., Guo, C., Hase, N., Hoffmann, F., John, F., Klassert, C., Lauf, T.: Theoretical foundations of human decision-making in agent-based land use models—A review. *Environmental modelling & software* 87, 39-48 (2017)
4. An, L.: Modeling human decisions in coupled human and natural systems: review of agent-based models. *Ecological Modelling* 229, 25-36 (2012)
5. Russell, S.J., Norvig, P.: *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited (2016)
6. Walton, D., Toniolo, A., Norman, T.J.: Towards a richer model of deliberation dialogue: Closure problem and change of circumstances. *Argument & Computation* 7, 155-173 (2016)
7. Thimm, M.: Strategic argumentation in multi-agent systems. *KI-Künstliche Intelligenz* 28, 159-168 (2014)
8. Walton, D., Krabbe, E.C.: *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press (1995)
9. McBurney, P., Hitchcock, D., Parsons, S.: The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems* 22, 95-132 (2007)
10. Carneiro, J., Martinho, D., Marreiros, G., Jimenez, A., Novais, P.: Dynamic argumentation in UbiGDSS. *Knowledge and Information Systems* 1-37 (2017)
11. Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: Arguing with Behavior Influence: A Model for Web-based Group Decision Support Systems. *International Journal of Information Technology & Decision Making* (2018)
12. Carneiro, J., Saraiva, P., Martinho, D., Marreiros, G., Novais, P.: Representing decision-makers using styles of behavior: An approach designed for group decision support systems. *Cognitive Systems Research* 47, 109-132 (2018)
13. McBurney, P., Parsons, S.: Dialogue games for agent argumentation. *Argumentation in artificial intelligence*, pp. 261-280. Springer (2009)