Defining an Architecture for a Ubiquitous Group Decision Support System

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

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

Abstract. Supporting group decision-making in ubiquitous contexts is fundamental while developing Group Decision Support Systems (GDSS). Here we propose an architecture that assures ubiquity and allows the development of a system which can be used anywhere at any time and through almost any sort of electronic device. Our approach can be used by developers that intend to build Ubiquitous Group Decision Support Systems (UbiGDSS). It uses three main components that are interconnected and that will allow to collect and preserve the amount and quality of intelligence generated in face-to-face meetings.

Keywords: Group Decision Support Systems, Ubiquitous Computing, Multi-Agent Systems, Intelligent Reports

1 Introduction

Over the last years, we have observed a growing necessity for Group Decision Support Systems (GDSS) to include ubiquitous computing in their development [1, 2]. It no longer makes sense to approach the group decision-making problematic the same way as ten or twenty years ago, and it is now fundamental to provide intelligent and efficient answers in busy environments such as the ones observed in large multi-national organizations [3].

This work explains a system architecture which can be used to develop GDSS while supporting ubiquitous computing and allows every participant to exchange knowledge regardless of time or location constraints [1]. This architecture uses three main components that are interconnected and that will allow to collect and preserve the amount and quality of intelligence generated in face-to-face meetings in a way very similar to the real group decision-making process.

The rest of the paper is organized as follows: In the next section, the system architecture is exposed and each component is detailed. In the following section, it is described how each component was implemented and some screenshots of those components are shown. Finally, some conclusions are taken in the last section, along with the work to be done hereafter.

2 System Architecture

The architecture proposed in this work is aimed at ubiquitous group decision support systems and therefore should support decision-makers anytime and anywhere. Fig. 1 shows the architecture considered for our system.

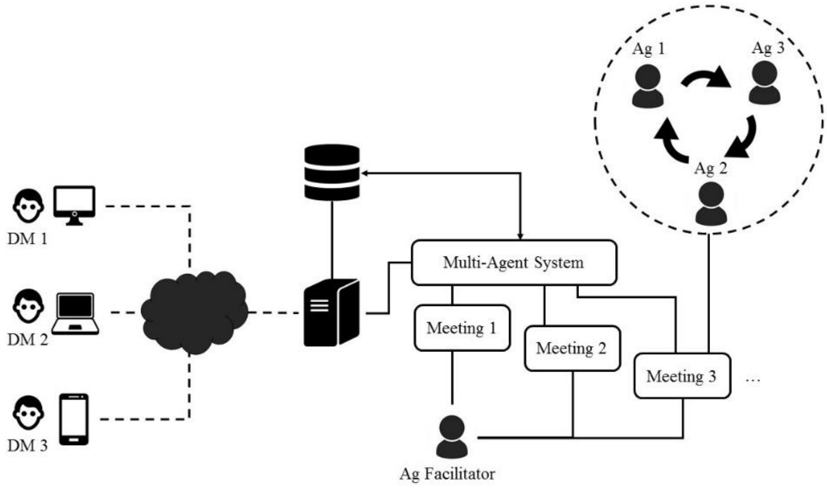

Fig. 1. Proposed System Architecture

The main components of the system are:

- A web application which will be the interface between the multi-agent system and the participant. This application allows problem configuration and reports the decision-making results to the participant. Our system is focused towards multi-criteria problems where participants can share their opinion about different alternatives based on existing criteria. Therefore, the interface first allows to define problem data (alternatives and criteria that will be discussed) and select which decision-makers can participate in the decision-making process. After that, each participant that was selected can make personal and problem configurations. After the decision-making process ends, the results will be reported back to the participant via an intelligent report adapted to each participant that is based on all these previous personal configurations. The web application is accessible by desktop and mobile browsers thus being available to almost any kind of electronic device;
- A database where all configurations related to each problem will be stored. These configurations include problem data, personal and problem configurations. The multi-agent system will load these configurations to simulate the group decision-making process;
- A multi-agent system which simulates different group decision-making processes based on the data available in the database. For each personal configuration, an agent

will be created and that will use the information provided in that configuration to represent the decision-maker and behave in the most desired way. It is two different types of agents will act in each meeting. There is always a facilitator agent that is responsible for coordinating and analysing the decision-making process, and there will be a group of participating agents that represent decision-makers and that will negotiate and persuade each other to choose an alternative as the solution for the problem.

2.1 Web application

Defining a configuration environment in which the decision-makers could model a multi-criteria problem is a complex process [4]. It is essential that the interface can support easy and fast configurations specially in the context of making decisions inside organizations. In this context, individuals with a very tight schedule (top managers and executives) should make decisions and it does not make sense to force them to fill extensive and lengthy configurations because they do not have the time nor will to do so [1]. The right idea is to assure some aspects related to interface development such as usability, simplicity, adaptability, and clarity. The interface should be complex enough whenever it is necessary. If the decision-maker does not have much interest in the problem being discussed, he/she will probably not want to look at very complex reports about the problem. On the other hand, if the decision-maker finds the problem being discussed to be very important he will likely prefer access to much more detailed information [5]. In our proposal, we have developed a web application which is the interface between the multi-agent system and the participant.

This application follows a general template that was proposed in [4] and that is divided in three main sections: Problem Data, Personal Configuration, and Problem Configuration. In the first section, it is specified problem specific information that includes criteria and alternatives information. It can also be specified other information indirectly related to the problem such as historical data, finances, statistics, etc. In the second section the decision-maker can model its own personal attributes and how he intends his/her representing agent to behave throughout the decision-making process. For this, the participant can select a conflict style and other participants who he/she considers to be credible and whose opinion should not be ignored. The third section is related to the configuration of problem-specific attributes. The third section is related to the problem configuration where each decision-maker can select: the preference towards each one of the available alternatives; the importance given for each criterion; preferred alternatives and criteria.

The web application was further extended in [5] to provide the results of the decision-making process via intelligent reports adapted to each decision-maker. These reports aim to clarify the decision-maker and show him/her what happened during the decision-making process and are built considering three key factors: expertise level, time, and interest. By combining (or not) these three factors the information provided to decision-makers could be more or less complex.

2.2 Database

The database component stores data related to each multi-criteria problem: problem data personal configurations (data inserted by decision-makers about personal attributes), problem configurations (alternatives and criteria preferences), user data (problems the user has been part of, user profile, etc.). It also collects and store information and results obtained from each decision-making process: alternative chosen, consensus percentages, requests exchanged between agents, average satisfaction level, etc.

2.3 Multi-agent system

Multi-agent systems have been frequently used as a tool to support group decision-making [6]. In this type of systems each decision-maker is represented by an agent that tries to negotiate and persuade other agents to accept his opinion. We have used a multi-agent system with two types of agents: a participant agent that represents a decision-maker according to the configurations that were provided and a facilitator that coordinates and analyses the entire decision-making process.

Facilitator Agent. The facilitator main actions include:

1. Load problem data – The facilitator loads problem data such as alternative, criteria and selected agents which is available to all participant agents;
2. Notify agents before the process begins – After each participant agent has been created it notifies those selected to be part of the decision-making process;
3. Manage participant agents' communications – The negotiation model used between agents follows a communication logic inspired in social networks [7] and there have been considered two types of communication: public and private. For both communications, the facilitator receives and forwards all sent messages to the respective recipients. Besides that, the facilitator analyzes the content of each message to assure that no duplicate messages will be sent towards the same recipient agent or to verify the necessary conditions to end the decision-making process. For communication, the facilitator chooses which participant agent can start a public conversation topic as well as when to close that topic and start another one.
4. Finish the process – The facilitator must end the decision-making process whenever an alternative is accepted by all participant agents or when agents have no more messages to exchange with other agents. As the process ends the facilitator should transmit its results such as the alternative that was chosen which will be stored in the database and be available to all decision-makers.

Participant Agent. Participant agents are the virtual representation of decision-makers. A participant agent will behave according to all configurations provided by the decision-maker and attempts to negotiate and persuade other agents to accept its opinion. It will receive information throughout the decision-making process regarding other agents' preferences and decide if it should make a certain request or receive requests and decide if it should accept or reject that request. shows the internal architecture of the participant agent. Fig. 2 details the three layers considered in the participant agent architecture.

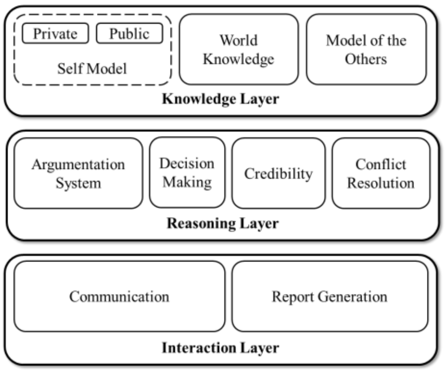

Fig. 2. Participant Agent Architecture

In the knowledge layer the participant agent has knowledge about the environment surrounding him. This includes information he detains about the problem, information about itself which includes its preferences and goals (public and private model) and information related to the public profile of other agents.

The reasoning layer allows the participant agent to reason about the received information, whether it receives a request or another kind of message. In this layer, it is considered four main components:

- Argumentation system – This system is responsible for identifying and evaluating arguments. The participant agent selects the most appropriate argument at a certain point throughout the decision-making process. It will analyze the current number of agents supporting each alternative as well as their criteria preferences to select the argument that should be sent in the request. The participant agent also uses the argumentation system to evaluate received requests and related arguments;
- Decision-Making – This component is used for the agent to measure its preferences and verify which alternatives are more likely to be accepted throughout the decision-making process;
- Credibility – This module is used with the information provided by the decision-maker regarding whose opinion he/she thinks to be credible. The participant agent will use this information when evaluating the opinion of other agents;
- Conflict Resolution – This module contains all the information regarding the conflict style which is used by the participant agent. This information includes values that were measured for all the dimensions identified for each conflict style and that will be essential to define the behavior of the agent. All conflict styles considered have been proposed in [8].

The interaction layer is responsible for the communication between agents as well as for the generation of reports for each agent after the decision-making process ends which will be provided to each decision-maker and whose content will vary according to the initial configurations provided by him/ her.

3 Implementation

In this section, it will be described how which component presented in the system architecture has been implemented.

3.1 Web application

The web application is a ubiquitous application that can be used in web browser in both desktop or mobile devices. The problem data definition, problem and personal configuration has been developed using JavaScript and Java, ASP .NET and C#. Any user can use the system from anywhere, at any time and from almost any kind of device with the only restriction being having access to the Internet. The meeting organizer should first define problem data and after that each decision-maker can make personal and problem configurations using the template proposed in [4]. The multi-agent system will perform the decision-making process with each agent representing each decision-maker and the meeting results will be sent to each decision-maker via an Intelligent report as seen in Fig. 3.

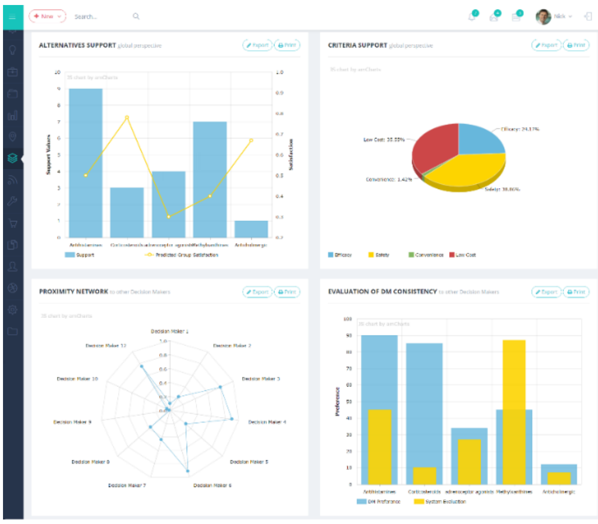

Fig. 3. Intelligent Report

The content of an Intelligent Report is defined according to the interests of the decision-maker and is divided in two sections. The first section presents global information about the level of consensus (preference given by each decision-maker) towards each alternative. The second section presents information about the self and other group elements. In this section, it is considered information with a higher level of complexity such as forecasts or simulations of different scenarios, identification of different groups (concerning their preferences), as well as explanations about decisions done by the system.

3.2 Multi-agent system

The multi-agent system is developed using Java Agent Development Framework (JADE) and offers a set of features which are essential in multi-agent systems development such as synchronous and asynchronous communication between agents, support concurrency, offer security mechanism via SSL and support agent mobility. Some screens of agents' communication can be seen in Fig. 4 and Fig. 5.

In Fig. 4 it is shown agents exchanging information regarding alternatives preference. An agent asks other agents what are they most preferred alternatives and they reply stating their most preferred alternative. In Fig. 5 it is shown an example of a private conversation where an agent performing a request to another agent.

Fig. 4. Agents' Communication via request

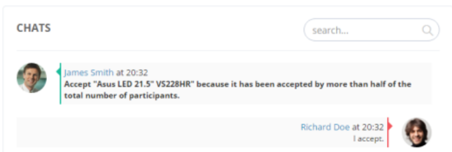

Fig. 5. Agents' Communication via statement

4 Conclusions and Future Work

The group decision-making process has evolved over the last decades and it only makes sense that GDSS should also evolve (using the latest technologies) to overcome emerging issues such as the difficulty to gather decision-makers at the same place and time. We now start seeing the development of UbiGDSS that can support users anywhere, at any time and from almost any sort of electronic device.

In this work, we propose an architecture which can be used to develop UbiGDSS and present three main components that will assure ubiquity to the system. Furthermore, we specify each component and show how they have been developed. As future work, we will keep improving the web application and perform usability tests focused on intelligent reports. We want decision-makers to be able to validate reports created by the system and the relevance of information presented in each report.

5 Acknowledgements

This work was supported by COMPETE Programme (operational programme for competitiveness) within Project POCI-01-0145-FEDER-007043, 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, UID/EEA/00760/2013, and the João Carneiro PhD Grant with the Reference SFRH/BD/89697/2012.

References

1. Carneiro, J., Santos, R., Marreiros, G., Novais, P.: Overcoming the lack of human-interaction in ubiquitous group decision support systems. Advanced Science and Technology Letters 49, 116-124 (2014)
2. 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)
3. Luthans, F.: Organizational Behavior 12 edition. McGraw-Hill Education (2010)
4. Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: A general template to configure multi-criteria problems in ubiquitous GDSS. International Journal of Software Engineering and its Applications 9, 193-206 (2015)
5. Conceição, L., Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: Generation of intelligent reports for ubiquitous group decision support systems. In: Global Information Infrastructure and Networking Symposium (GIIS), 2016, pp. 1-6. IEEE, (2016)
6. Ramos, C., Marreiros, G., Santos, R., Freitas, C.F.: Smart offices and intelligent decision rooms. Handbook of Ambient Intelligence and Smart Environments, pp. 851-880. Springer (2010)
7. Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: Intelligent negotiation model for ubiquitous group decision scenarios. Frontiers of Information Technology & Electronic Engineering 17, 296-308 (2016)
8. Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: Defining Agents' Behaviour for Negotiation Contexts. In: Portuguese Conference on Artificial Intelligence, pp. 3-14. Springer, (2015)