A Consensus-based Group Decision Support System using a Multi-Agent MicroServices Approach

Demonstration

João Carneiro
GECAD, Polytechnic of Porto
Porto, Portugal
jrc@isep.ipp.pt

Rui Andrade
GECAD, Polytechnic of Porto
Porto, Portugal
rfaar@isep.ipp.pt

Patrícia Alves
GECAD, Polytechnic of Porto
Porto, Portugal
pat@isep.ipp.pt

Luís Conceição
GECAD, Polytechnic of Porto
Porto, Portugal
msc@isep.ipp.pt

Paulo Novais
Algoritmi, University of Minho
Braga, Portugal
pjon@di.uminho.pt

Goreti Marreiros
GECAD, Polytechnic of Porto
Porto, Portugal
mgt@isep.ipp.pt

ABSTRACT

In this demo, we present a Consensus-based Group Decision Support System that makes use of a Multi-Agent Microservices approach. The proposed system comprises a Client App, an API Gateway and a set of Microservices where different Artificial Intelligence methods are implemented. It allows to create and manage Multiple Criteria Decision Problems and to support dispersed decision-makers, allowing them to take advantage of the benefits associated to face-to-face Group Decision-Making processes.

KEYWORDS

Multi-Agent MicroServices, Group Decision-Making, Group Decision Support Systems

1 INTRODUCTION

Decision-makers all over the world demand for technologies to support them in Group Decision-Making (GDM) processes, where they can share ideas and contribute to a decision that evolves and changes over time. Today, the need for strategies to support disperse groups in the decision-making processes is much more than just to overcome the members impossibility to meet at the same space and at the same time. It is mandatory to have mechanisms capable of increasing the participants' knowledge, sense of others and sense of the process. In summary, the society asks for Group Decision Support Systems (GDSS) capable of creating intelligence, increasing the value and quality of decisions.

In this demo, we present a Consensus-based GDSS that intends to meet those demands whilst allowing dispersed decision-makers to take advantage of the benefits associated to face-to-face GDM processes. As the proposed system is complex, scalable and has

to integrate so many features and functionalities (implemented using different programming languages), we decided to use the Microservices Architecture Pattern since it has features capable of empowering a system like this, such as: a better faults isolation, continuous delivery, components spread across multiple servers, an easier understandability since they represent small pieces of functionality, an easier integration, automatic deployments, among others. In addition, microservices that include Multi-Agent Systems (MAS) follow the Multi-Agent Microservices (MAMS) [7] approach which allows to interact with agents through RESTful APIs and consequently to expose the agents' knowledge and capabilities directly as resources. The demonstration video can be found at: https://youtu.be/hv3Zy0PSqaY.
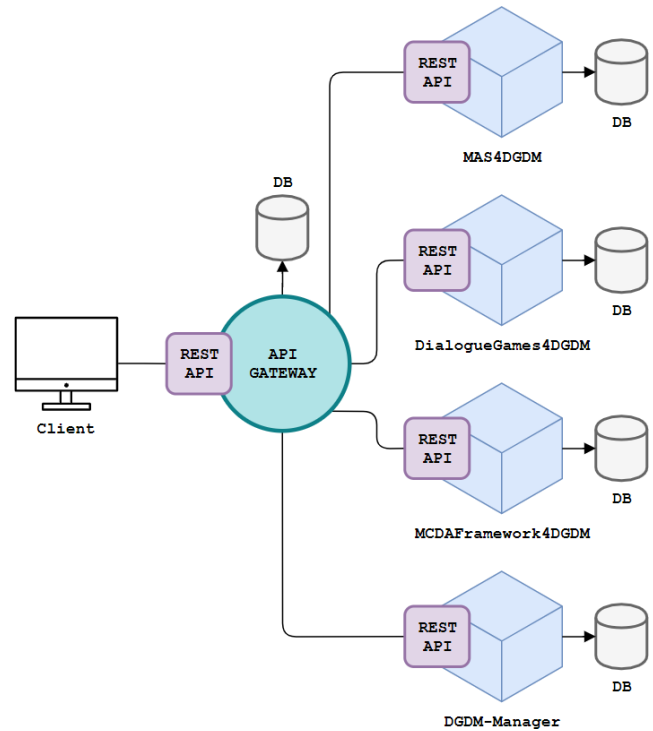
2 THE PROPOSED SYSTEM

The presented Consensus-based GDSS comprises a Client App, an API Gateway and a set of microservices (see Figure 1). The Client App was developed using the Angular framework and was designed using the Responsive Web Design approach. The API Gateway was developed using the ASP.NET Core framework and works as a single-entry point into the system, which allows the internal system architecture, comprised of microservices, to be encapsulated, providing an API tailored to each Client. The authentication/authorization functionalities are also of the API Gateway responsibility and were implemented using the ASP.NET Core Identity and JWT (JSON Web Tokens). All the requests in the GDSS are asynchronous. In terms of microservices, the system comprises 4:

- DialogueGames4DGDM: This microservice was developed using Java, and the MAS4GDM framework [2] was used to facilitate the implementation of dialogues between agents. These agents are referred to as "Private Agents" because they can only be accessed within the GDSS. In this microservice we implemented an argumentation-based dialogue model (according to the one proposed in [3]), which allows agents to exchange arguments and to anticipate the best solution according to the decision-makers preferences, interests and needs at that moment. These agents have a short life period, because they are solely created to perform the dialogue and they die when the process ends. In addition to the typical data related to the decision-makers' preferences (alternatives and

criteria), each agent is modeled with contextual aspects (the decision-maker's expertise level, style of behavior and what decision-makers he/she considers credible for that specific decision problem). As each agent can be modeled with a different style of behavior (Dominating, Obliging, Integrating, Avoiding and Compromising [5]), they behave differently in varied situations, such as: in the request evaluation, in the propensity to have a more "Concern for Self" or "Concern for Others" discourse and in their propensity to open new topics. As can happen with other microservices, these agents can themselves be clients of this GDSS, i.e., the agents are capable of using the other microservices endpoints (through the API Gateway) to search for new knowledge, allowing them to build new arguments and to use them in their dialogues.

- MAS4DGDM: Implemented using Java consists on a MAS developed using JADE [1]. The agents in this MAS represent real decision-makers (one agent per decision-maker), they are persistent and are referred as "Public Agents" because they can be externally accessed through a URI (https://<host:port>/<agent-unique-name>/<resource-path>). Since they are persistent, they include long-term data about the decision-maker they represent, such as his/her personality (we used the Five Factor Model of Personality to model the decision-makers' personality [6]) and the history of decision-makers he/she considered credible. These agents can communicate with each other in order to get information about other decision-makers and to learn from their experience. In order to facilitate the implementation of communication between agents, the MAS4GDM framework was used [2]. The information that flows through this microservice is analyzed and processed by these "Public Agents". Each agent is responsible for selecting the appropriate information that should be sent to the Client App and presented to the decision-maker it represents, considering aspects that were previously referred and contextual aspects, such as: expertise level, style of behavior [5], credibility and the predicted decision satisfaction [4]. This specific information is presented in one of the two screens of the Client App, which can be accessed by the decision-makers to view the results and other information about the decision process. As this screen presents decision-maker-related data selected by the "Public Agents", each decision-maker has his/her own personalized page.

- MCDAFramework4DGDM: This microservice was developed using the ASP.NET Core framework and includes a Multiple Criteria Decision Analysis framework that allows to perform a more objective (analytical) analysis of decision problems. It has an endpoint that returns the "Solved Problem", data that is presented in a screen of the Client App so the decision-makers can consult the results of each round (iteration). The data presented in this screen are the same to all decision-makers involved in the decision process. Some examples of the data returned are: the alternatives recommended to be accepted as solution, the alternatives recommended to be rejected in order to reduce the decision problem complexity, the predicted importance of each criterion and alternative in the perspective of each decision-maker, among others.

Figure 1: The proposed Consensus-based Group Decision Support System Architecture.

- DGDM-Manager: This microservice was developed using the ASP.NET Core framework and is intended to satisfy all the needs regarding the business model, such as: POST new decision problems, POST new rounds, GET the list of existing decision problems, POST decision-makers preferences, among dozens of other tasks.

3 CONCLUSIONS

According to what we experienced in the development of this prototype, the Microservice Architecture Pattern fits like a glove for the context of complex systems as is the case of GDSS. In addition, the MAMS approach demonstrated to be a good solution to overcome some of the limitations associated to the interaction with MAS. Moreover, the presented GDSS allows to benefit from the best of both worlds, i.e., if on the first hand microservices principles are focused in the definition of external attributes, which greatly facilitates the interaction with MAS, on the other hand, MAS have the main focus on internal attributes, which allows to benefit from their social abilities, pro-activity and knowledge.

REFERENCES

[1] Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. 2008. JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology* 50, 1-2 (2008), 10–21.

[2] João Carneiro, Patrícia Alves, Goreti Marreiros, and Paulo Novais. 2019. A multi-agent system framework for dialogue games in the group decision-making context. In *World Conference on Information Systems and Technologies*. Springer, 437–447.

[3] João Carneiro, Diogo Martinho, Goreti Marreiros, Paulo Novais, et al. 2019. Arguing with behavior influence: a model for web-based group decision support systems. *International Journal of Information Technology & Decision Making (IJITDM)* 18, 02 (2019), 517–553.

[4] João Carneiro, Pedro Saraiva, Luís Conceição, Ricardo Santos, Goreti Marreiros, and Paulo Novais. 2019. Predicting satisfaction: perceived decision quality by decision-makers in web-based group decision support systems. *Neurocomputing* 338 (2019), 399–417.

[5] João Carneiro, Pedro Saraiva, Diogo Martinho, Goreti Marreiros, and Paulo Novais. 2018. Representing decision-makers using styles of behavior: an approach designed for group decision support systems. *Cognitive Systems Research* 47 (2018), 109–132.

[6] Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality* 60, 2 (1992), 175–215.

[7] Rem W. Collier, Eoin O'Neill, David Lillis, and Gregory O'Hare. 2019. MAMS: Multi-Agent MicroServices. In *Companion Proceedings of The 2019 World Wide Web Conference*. 655–662.