Università di Genova | DIPARTIMENTO INFORMATICA, BIOINGEGNERIA, ROBOTICA E INGEGNERIA DEI SISTEMI

Computer Science Workshop
PhD program in Computer Science and Systems Engineering

Intentional dialogues in multi-agent systems based on ontologies and argumentation

Débora C. Engelmann, Alison R. Panisson, Viviana Mascardi, Rafael H. Bordini

Introduction

This research focuses on combining Argumentation Theory and Ontology techniques to support complex dialogues in natural language. In particular, we aim to create an approach to support the development of dialogue systems that take advantage of that combination of techniques within BDI (Belief-Desire-Intention) agents to assist humans in decision making. Our goal is to make this approach flexible enough that it can be applied across multiple domains. We are particularly interested in applying our approach to healthcare domain.

Main Objectives

General objective: *Investigate how argumentation theory and ontology techniques can be used together with reasoning about intentions to build complex natural language dialogues to support human decision making.*

Specific objectives:

1. Survey the literature to investigate how the scientific community is using argumentation techniques to achieve explainable artificial intelligence in dialogue systems.
2. Design and formalise an approach to argumentation-based dialogues and ontological reasoning.
3. Apply the proposed formal model in a real-world domain and problem, such as healthcare.
4. Validate the approach with domain experts.

A Multi-Agent Intentional Dialogue System

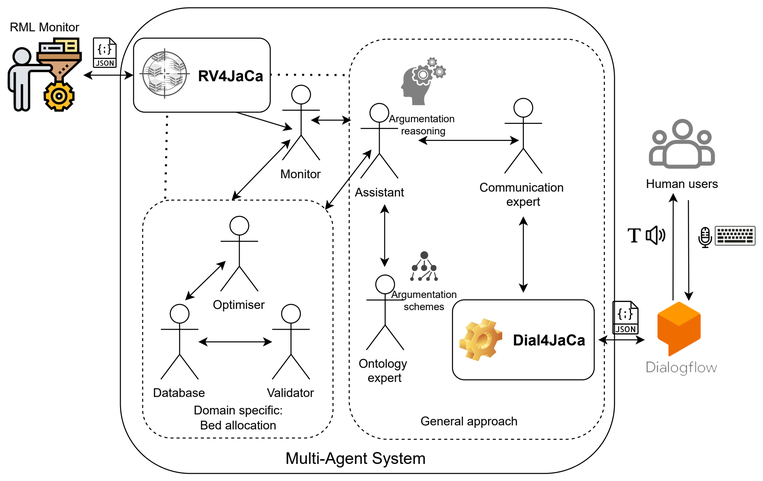

Figure 1: Framework overview

General approach:

- The `Human user` can interact with the chatbot through voice or text. This interaction is classified into intents by Dialogflow and sent to `Dial4JaCa` [1] (an interface to dialogue platforms such as Dialogflow), which makes the request available to the `Communication expert` agent assigned to that specific user.
- One or more `Communication expert` agents can be instantiated, each one responsible for representing a `Human user`. This agent also translates the responses of the `Assistant` (the result of the MAS[1] reasoning) into natural language messages, using templates as described in [4], to be sent to its corresponding `Human user`.
- The `Assistant` agent is responsible for communicating with other agents in search of information and for performing argument reasoning.
- Several `Ontology expert` agents perform ontological reasoning using the Pellet reasoner and its open-source continuation effort Openllet[2]. In addition, these agents can translate OWL[3] inference rules automatically [3] to defeasible rules (representing argumentation schemes) and use them during the reasoning process.
- `RV4JaCa` [2] is responsible for collecting information about all messages exchanged between agents and sending them through a REST request to the RML[4] monitor. After processing the received message, the monitor returns a result that identifies whether the message sent from one agent to the other violates any of the properties being checked by it. If a property is violated, RV4JaCa adds this information to the `Monitor` agent's belief base, which in turn warns the agents involved in the exchange of messages that there has been a violation. This makes it possible for our agents to take action to recover from the failure that the breach caused.

Domain specific (Bed allocation):

- The `Validator` agent validate bed allocation plans using a PDDL[5] plan validator;
- The `Optimiser` agent is responsible for making suggestions for optimised allocations using the GLPSol solver of GLPK (GNU Linear Programming Kit), which is a free open source software for solving linear programming problems; and
- The `Database` agent query and update the bed allocation system database.

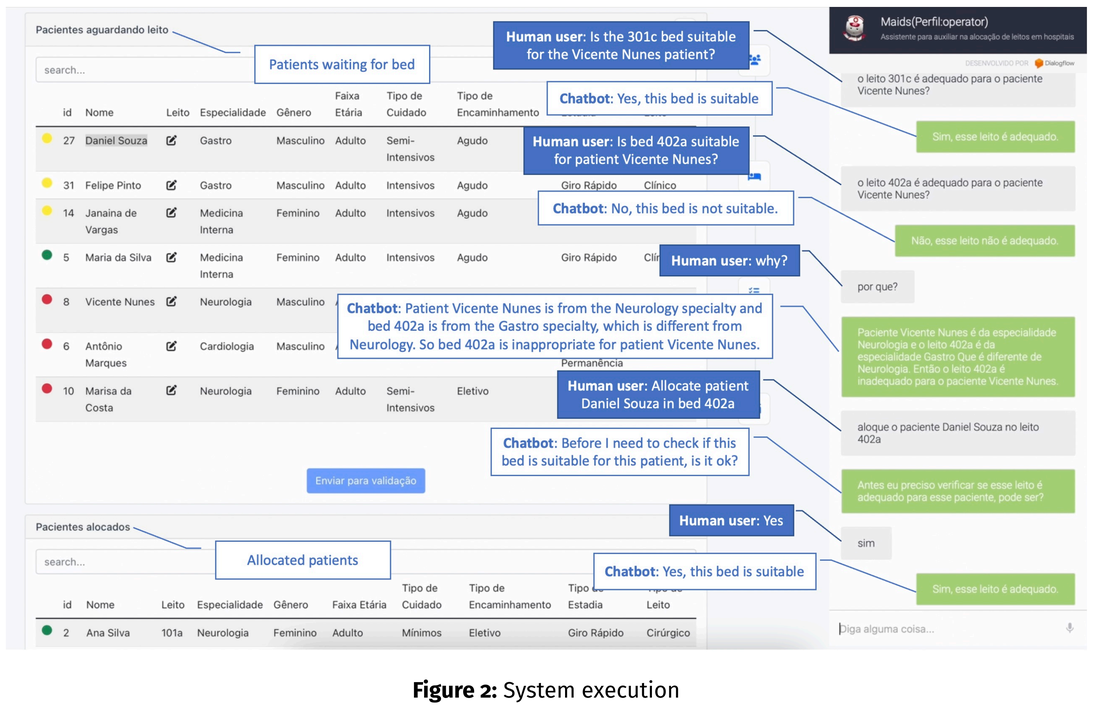

Figure 2: System execution

Conclusions and Forthcoming Research

- Our framework allows agents to engage in argumentation-based dialogues to support bed allocation, which makes them aware of other agents' reasons/justifications/opinions about a particular bed allocation, so interface agents can provide argumentation-based explanations to users, resulting from the collective construction of such arguments.
- Runtime verification's extra security layer provided by RV4JaCa is extremely useful in MAS. Since, in general, many communications between agents (and sometimes, humans) need to be performed, we need to consider that the developer does not have complete control over the interactions that human users can make during a dialogue. In this sense, RV allows us to avoid unexpected and probably inappropriate system behaviour.
- We are evaluating our approach with staff from Hospital São Lucas da PUCRS in Brazil.
- Among the intended future works are providing a pilot of the instance aimed at allocating beds of our framework to be used in the daily activities of Hospital São Lucas and applying our approach to other domains.

References

[1] Débora Engelmann, Juliana Damasio, Tabajara Krausburg, Olimar Borges, Mateus Colissi, Alison R Panisson, and Rafael H Bordini. Dial4jaca–a communication interface between multi-agent systems and chatbots. In *PAAMS*, pages 77–88. Springer, 2021.

[2] Débora C Engelmann, Angelo Ferrando, Davide Ancona, Alison R Panisson, Rafael H Bordini, and Viviana Mascardi. Rv4jaca - runtime verification for multi-agent systems. In *Proceedings of the Second Workshop on Agents and Robots for reliable Engineered Autonomy*. Open Publishing Association, 2022.

[3] Carlos Ferreira, Alison R Panisson, Débora C Engelmann, Renata Vieira, Viviana Mascardi, and Rafael H Bordini. Explaining semantic reasoning using argumentation. In *PAAMS*. Springer, 2022.

[4] Alison R Panisson, Débora C Engelmann, and Rafael H Bordini. Engineering explainable agents: an argumentation-based approach. In *International Workshop on Engineering Multi-Agent Systems*, pages 273–291. Springer, 2021.

Acknowledgements

This research was partially funded by CNPq and CAPES – Finance Code 001.

CONTACTS

Débora C. Engelmann
debora.engelmann@acad.pucrs.br

Viviana Mascardi
viviana.mascardi@unige.it

Alison R. Panisson
alison.panisson@ufsc.br

Rafael H. Bordini
rafael.bordini@pucrs.br

[1] Multi-agent system
[2] https://github.com/Galigator/openllet
[3] Web Ontology Language
[4] Runtime Monitoring Language
[5] Planning Domain Definition Language