










Dial4JaCa – A Demonstration

Débora Engelmann^{1,3} , Juliana Damasio¹ , Tabajara Krausburg^{1,4} ,
Olimar Borges¹ , Lucca Dornelles Cezar¹ , Alison R. Panisson² ,
and Rafael H. Bordini¹ 

¹ School of Technology, Pontifical Catholic University of Rio Grande do Sul,
Porto Alegre, Brazil

{debor.a.engelmann,juliana.damasio,tabajara.rodrigues,olimar.borges,
lucca.cezar}@edu.pucrs.br, rafael.bordini@pucrs.br

² Department of Computer Engineering, Federal University of Santa Catarina,
Florianópolis, Brazil

alison.panisson@ufsc.br

³ DIBRIS, University of Genoa, Genoa, Italy

⁴ Department of Informatics, Technische Universität Clausthal (TUC),
Clausthal-Zellerfeld, Germany

Abstract. Multi-Agent Systems (MAS) have become a powerful approach to the development of complex AI systems. However, the community faces the challenge of developing communication interfaces between software agents and humans, for instance, using natural language. In this paper, we demonstrate an approach to the integration of MAS and chatbot technologies named Dial4JaCa. Dial4JaCa enables the implementation of MAS integrated with chatbots, which can be applied to a variety of domains. In this particular work, we demonstrate how it is possible to integrate a JaCaMo application (agent system) with Dialogflow (a chatbot platform) using Dial4JaCa.

Keywords: Multi-agent systems · Natural language · Chatbots

1 Introduction

A conversational agent (chatbot) is a computer program that interacts with users through natural language. With the advances of Natural Language Processing (NLP), conversational agents have started to play an important role in a variety of contexts. As a result, many platforms are available to provide mechanisms for NLP and dialog management. On the other hand, Multi-Agent Systems (MAS) are built upon core concepts such as distributed systems and reactivity, as well as individual rationality. Thus, agent technology is a promising way to provide strong reasoning capabilities to chatbot platforms.

In this paper, we demonstrate the use of Dial4JaCa, a communication interface between MAS and chatbot platforms. In particular, we demonstrate the implementation of a communication interface between a JaCaMo [3] application for supporting hospital bed allocation and Dialogflow¹. Our communication

¹ <https://dialogflow.com/>.

interface makes it possible to implement multi-agent systems in which agents and humans are able to have dialogues in natural language.

```

1 +request(ResponseId, IntentName, Params, Contexts)
2   ⋮ (IntentName == "Reply With Context")
3 <- contextBuilder(ResponseId, "test context", "1", Context);
4   replyWithContext("Hello, I am your Jason agent, and I am
5                     responding with context ", Context).

```

Listing 1: An example of a plan to reply to a request with a context.

2 Main Purpose

Dial4JaCa provides an integration² of the JaCaMo framework [3] and Dialogflow, and it allows developers to implement intelligent agents that are able to communicate with humans through natural-language interaction. We aim at making our approach as modular as possible so that it can be imported into any MAS project developed in JaCaMo. Figure 1 depicts the Dial4JaCa architecture.

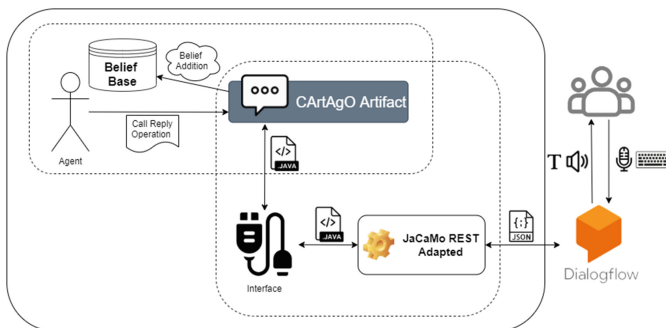


Fig. 1. The Dial4JaCa architecture.

Our approach is built upon a resource-oriented abstraction provided by the JaCaMo REST project³ [2]. We integrate it with *fulfillment* services available in Dialogflow. Our interface allows all agents that are observing a particular CArtAgO artefact to receive requests from Dialogflow. Then, it is up to the agents to decide whether they are going to react to such requests or not. We show in Listing 1 an example of a Jason plan to react to a request. In a request, Dial4JaCa makes an agent perceive **contexts** and **parameters** captured by Dialogflow. *Parameters* is a list containing the pattern **parameter**(key, value). *Contexts* is a list in which each element contains a context name, a lifespan, and a list of parameters.

² <https://git.io/Dial4JaCa>.

³ <https://git.io/jacamoRest>.

To reply to a request, the agents have at their disposal an operation in the Dial4JaCa artefact which may take in addition to the intended response a *context* or an *event*. The response is just plain text to be shown to the user. A reply with a context is important to provide to Dialogflow additional information besides the response to the user. To do so, an agent first creates a new context (line 3) in the same pattern as described above. Then it sends the information back through Dial4JaCa (lines 4–5). A reply with an event is useful in cases an agent is running out of time to reply to a request (e.g., if the chatbot platform has a predefined waiting timeout). In that case, Dial4JaCa automatically replies with an event for Dialogflow to reestablish the intention. Doing so, it gets more time for the agent to finish its reasoning. This mechanism can be used at most three times in a row.

Note that we can also use, given some small changes in Dial4JaCa, other natural language processing platforms such as Rasa, Watson, or Luis.

3 Demonstration

We use Dial4JaCa as a decision support system for hospital bed allocation⁴. This application extends the integration in [4], which was application specific.

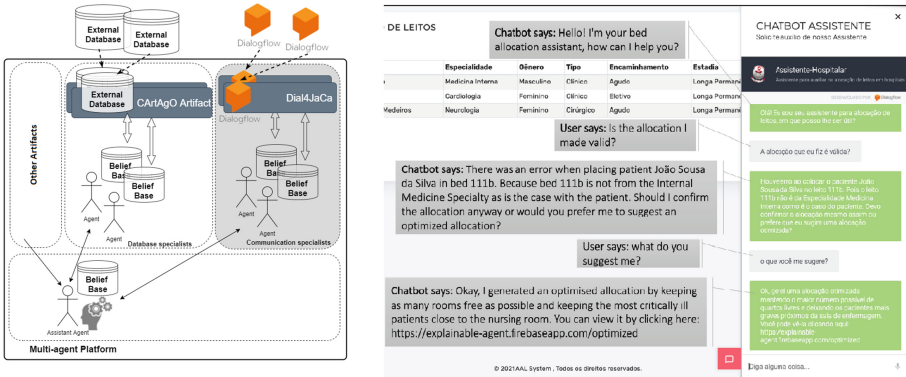


Fig. 2. Bed allocation system.

We show in Fig. 2 an overview of the application architecture (left) and part of the application interface⁵ (right). In it, the *assistant agent* is responsible for communicating with other agents to assist a user in a search for bed availability. The *database specialist* agents take care of any database query needed. These agents use a CArTAgO artefact to connect to hospital databases (in this application in particular).

⁴ <https://git.io/bedAlloSystem>.

⁵ <https://git.io/bedAllocSimul>.

Communication specialist agents use Dial4JaCa to provide a communication layer between an end user and the **assistant agent**. For instance, it is possible to instantiate an agent for each user of the system. This allows customised responses according to a user profile. This way an application can avoid giving too many explanatory answers to a user who has a specialist background, as well as avoid giving superficial answers to users with little background. The ability to instantiate multiple communication expert agents, one for each user of the system, also allows the **Assistant Agent** to engage in multiparty conversations, helping a team or a group of users to make joint decisions.

Moreover, there are two other agents populating the MAS application: the **optimiser agent** and the **validator agent**. The **optimiser agent** is responsible for communicating with a GLPK solver to generate optimal bed allocations. The **validator agent** uses a PDDL (Planning Domain Definition Language) validator⁶ and a CArtAgO artefact⁷ to validate bed allocation plans made by the user.

The current version of this MAS application has not been evaluated by professionals responsible for allocating beds in hospitals yet. However, we intend to do it in the near future. Dial4JaCa fulfilled its role in our preliminary tests, supporting communication and other functionalities as expected, and providing a complete integration between the MAS application and Dialogflow. In the next version of this multi-agent system application, we intend to use argumentation theory and ontology techniques, allowing agents to explain their suggestions for bed allocation. Explainability becomes an essential part of decision support systems, and Dial4JaCa can support this type of sophisticated interactions.

4 Conclusions

We made considerable progress towards natural language communication interfaces between humans and multi-agent systems. The scenario presented in this paper demonstrate the use of Dial4JaCa in practice, also showing promising preliminary results. We believe that several applications for various domains can be developed using Dial4JaCa, for example, ambient intelligence and law. Also, our approach provides support to the development of applications in the context of hybrid intelligence [1], in which human-agent collaboration is fundamental. In our future work, we intend to explore the integration with other chatbot development platforms. We also intend to apply Dial4JaCa in other domains.

References

1. Akata, Z., et al.: A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer* **53**(8), 18–28 (2020)

⁶ <https://git.io/PDDLPlanValid>.

⁷ <https://git.io/PDDLPlanValidArtif>.

2. Amaral, C.J., Hübner, J.F., Kampik, T.: Towards Jacamo-rest: a resource-oriented abstraction for managing multi-agent systems. arXiv preprint [arXiv:2006.05619](https://arxiv.org/abs/2006.05619) (2020)
3. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with Jacamo. *Sci. Comput. Programm.* **78**(6), 747–761 (2013)
4. Engelmann, D.C.: An interactive agent to support hospital bed allocation based on plan validation. dissertation, PUCRS (2019)