



# PACAs: Process-Aware Conversational Agents

Luis Fernando Lins<sup>1</sup>(✉), GlauCIA Melo<sup>2</sup>, Toacy Oliveira<sup>1</sup>, Paulo Alencar<sup>2</sup>,  
and Donald Cowan<sup>2</sup>

<sup>1</sup> Federal University of Rio de Janeiro, Rio de Janeiro, Brazil  
luisfernandolins@poli.ufrj.br, toacy@cos.ufrj.br

<sup>2</sup> Computer Systems Group, University of Waterloo, Waterloo, Canada  
{gmelo, palencar, dcowan}@uwaterloo.ca

**Abstract.** Processes are an essential concept in modern society, promoting the standardization, documentation, and control of complex interactions between businesses, governments, and individuals. Therefore, it is essential to provide approaches and tools that support the participants during the execution of these processes. However, current solutions for process guidance are usually proprietary, difficult to follow, and not effective in supporting these participants in their daily processes. In this paper, we present an approach for building a novel type of conversational agents – which we call PACAs – that can communicate with workflow management systems, using the BPMN standard, to support the execution of a process. We also describe a use case in which a PACA is implemented by using Rasa – a chatbot framework – and Camunda Engine – a workflow management system. We believe that promoting the integration of processes and conversational agents is key to simplifying process execution and, thus, making business processes more widespread and prevalent in the academy and industry.

**Keywords:** Business process · Conversational agent · Workflow · Chatbot · Camunda · Rasa

## 1 Introduction

Processes are pervasive in personal and organizational settings. They can support a wide variety of tasks, ranging from scheduling a trip [8] to hiring new employees or admitting students into a university program [4]. For this reason, organizations are increasingly relying on automated processes to improve efficiency and accuracy in the execution of their tasks.

In many cases, processes require people to perform tasks manually. These processes are known as human workflows [2]. Although mechanisms have been proposed to visualize, control, and manage processes, performing tasks in a process involving numerous human decisions can be more complicated than it seems. First, tasks may have either a strict or a loose order, and this might leave participants confused in complex scenarios. Second, dependencies between tasks can

be challenging to track, in which case, not being aware of the information that is required *a priori*, people may start executing tasks they are not able to finish properly. Third, processes that rely heavily on humans may differ based on their context or require access to ancillary documentation so that errors are avoided [3]. Owing to these challenges, process participants should have a solution to guide them through process instances.

In recent years, a new technology called “conversational agents” (CAs) [5] has been gaining increased attention. CAs are programs that assist in human-computer interactions (HCI) [6] through spoken and written language. Although CAs are flexible enough to support several types of HCI scenarios, they typically lack business process concepts, i.e., they do not consider the context of a process instance when interacting with users.

In this paper, we investigate an approach for integrating process concepts into conversational agents. This integration leads to agents of a novel type, which we call process-aware conversational agents (PACAs). The paper is structured as follows. Section 2 presents related work on process management in conversational agents. Section 3 provides an overview of our proposed solution and introduces a guide that can help with the development of PACAs. Section 4 describes how a PACA can be used in a real-life scenario involving trip planning. Finally, Sect. 5 presents concluding remarks and future work.

## 2 Related Work

A few conversational agents that assist with tasks have been developed by the academic community, and several process management tools are available in the industry. In 2017, Cranshaw et al. built and evaluated an app called Calendar.help [1] that uses the concept of process workflows for assisting with scheduling and rescheduling meetings. In 2018, Toxtli et al. created a tool called TaskBot [7] with the goal of helping teams to complete assignments. The authors intended to reduce the onus of context switching by using the chatbot to assign tasks to team members and remind them to start or finish their tasks. These solutions, although valuable, use an empiric process notation, lacking the compatibility that a universal modeling standard, such as BPMN, can provide. Furthermore, these related works do not provide an easy integration with any business process management (BPM) tools.

In recent years, there has been a rise in the number of BPM tools available in the industry. Tools such as Kissflow<sup>1</sup>, Process Street<sup>2</sup>, and Camunda<sup>3</sup> were developed to help businesses and their employees track the execution of essential and recurring processes. Nevertheless, when business processes become too large, employees can make mistakes during their execution [3], and these process management tools, owing to their complexity, may not adequately help

<sup>1</sup> <https://kissflow.com/>.

<sup>2</sup> <https://www.process.st/>.

<sup>3</sup> <https://www.camunda.com/>.

these workers. Therefore, the tool that we present in this paper aims to complement BPM tools – in this case, Camunda – to make them more user-friendly by leveraging natural language understanding via chatbots.

### 3 Proposed Solution

This section describes our approach and its implementation, offering an overview of how PACAs are configured and used, and presenting the outline of a guide for generating PACAs.

PACAs are developed by integrating chatbots with a Workflow Management System (WFMS) to control the execution of a BPMN process with natural language. Our proposed solution is based on Rasa, an open-source chatbot framework, and a WFMS called Camunda Engine. We have decided to use Camunda Engine because it is an open-source framework that is well documented and widely used both in academia and industry. In our proposed solution, Rasa will perform Natural Language Understanding (NLU) to determine what the user is saying. Then, Rasa Core, the framework's back end, will decide which action the chatbot should take next. The Core will call Camunda Engine's API to know which tasks are available, present them to the user in natural language, and, once the user tells the chatbot that a task is complete, the Core will complete that task in Camunda.

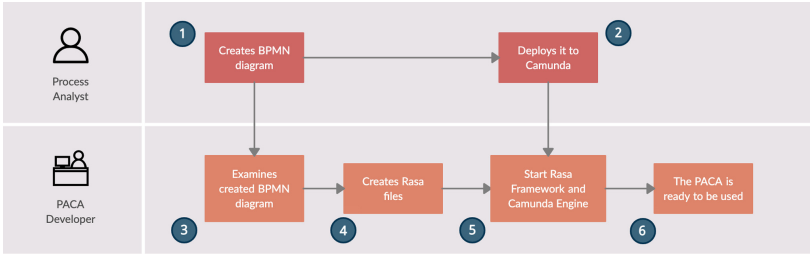
Once the chatbot is running, a user can communicate with it through the command line or API requests. It can also be run on Rasa X, a default GUI provided by the Rasa framework, or connected to Facebook Messenger, Slack, Telegram, and other messaging services.

We have split the life cycle of a PACA into two phases: configuration and usage. The former is shown in Fig. 1, while the latter is represented in Fig. 2. During the configuration phase, the process analyst creates a process model in Camunda Modeler<sup>4</sup>, makes necessary modifications to the model (Fig. 1.1), and deploys the process model to Camunda Engine (Fig. 1.2). Then, the developer will examine this BPMN model (Fig. 1.3) and convert process modeling elements from the BPMN diagram into Rasa files (Fig. 1.4). Finally, as soon as the Rasa Framework and Camunda Engine are running (Fig. 1.5), Rasa will be able to communicate with Camunda Engine, giving rise to a fully functioning Process-Aware Conversational Agent (Fig. 1.6).

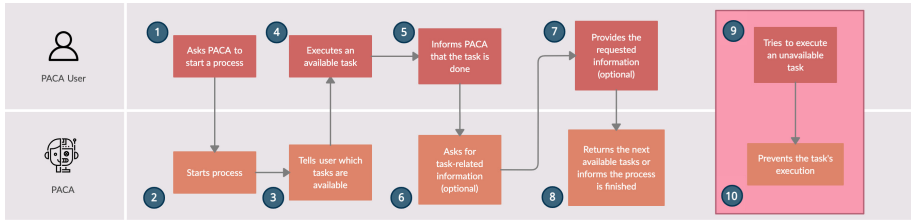
With an operational PACA, the user can then use it to navigate the process. Whenever a user says they want to start a process (Fig. 2.1), the PACA will start a process instance in Camunda Engine (Fig. 2.2) and then provide a list of all the available tasks for the user (Fig. 2.3). The user will then execute one of the available tasks (Fig. 2.4) and tell PACA when they have finished that task (Fig. 2.5). PACAs can also request task-related information if necessary. If that is the case, after the task is completed, the chatbot can request those pieces of information (Fig. 2.6), which will then be provided by the user (Fig. 2.7). Finally,

---

<sup>4</sup> <https://camunda.com/products/camunda-platform/modeler/>.



**Fig. 1.** Overview of the necessary steps to configure a PACA.



**Fig. 2.** Overview of the necessary steps to use a PACA.

the PACA will reply with the next available task(s) (Fig. 2.8) – in which case, the flow would go back to step 5 – or with a message indicating that the process is finished if the execution has reached the end event. In an exception flow, if the user tries to execute an unavailable task (Fig. 2.9), PACA will prevent that task’s execution (Fig. 2.10).

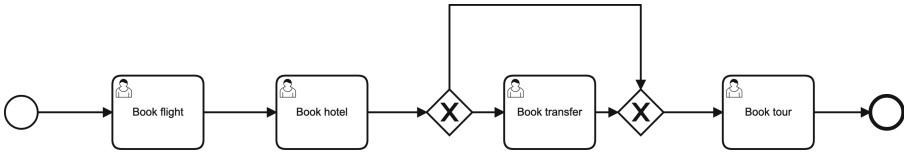
To assist in the development of PACAs, we have created a PACA Generation Guide, which should be used by the developers and process analysts who are working to create PACAs. The guide consists of multiple steps and substeps, but in this paper, we will only mention the main ones. They are: (i) Preparing the BPMN; (ii) Building the default Rasa chatbot; (iii) Adding process-specific pieces to the Rasa chatbot; and (iv) Running Camunda and the chatbot. The complete description of each of these steps is available at <http://www.repositorio.poli.ufrj.br/monografias/projpoli10033716.pdf>.

## 4 Use Case: The Trip Planning PACA

We will now explore a PACA that was built for a “Trip Planning” scenario. This scenario serves as an excellent example because the process is well known and has dependencies between tasks. The code for the “Trip Planning” PACA that was developed is available on GitHub<sup>5</sup>. Figure 3 presents the diagram for this process. It includes four user tasks (represented as rectangles), process gateways (represented as diamonds), and sequence flows (represented as arrows). The start

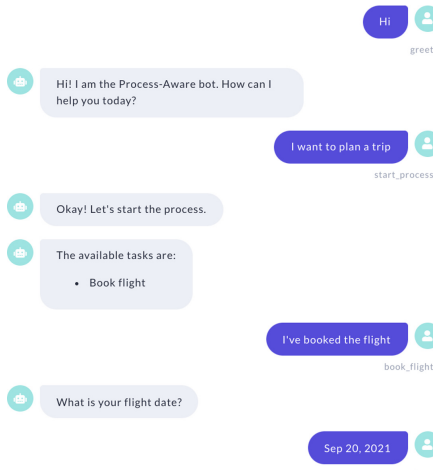
<sup>5</sup> <https://github.com/luis-f-lins/process-aware-conversational-agent>.

and end events are represented as circles at the beginning and end of the diagram, respectively.



**Fig. 3.** BPMN diagram for the “Trip Planning” process.

As shown in Fig. 4, after Camunda and Rasa are running, a user can then begin a conversation and ask to plan a trip, which will, in turn, start a process instance in Camunda Engine. To follow the process instance execution, we use a GUI provided by Camunda, called Camunda Cockpit. The “Trip Planning” process instance in Camunda Cockpit is shown in Fig. 5. After receiving the list of available tasks, including the “Book flight” task, the user can proceed to book the flight and inform the chatbot that the task is finished. In this case, before replying with the list of the next available tasks, the chatbot will ask the user for a required piece of information, which is the flight date. After the user provides the flight date, this piece of information will be stored in Camunda Engine, as shown on the list at the bottom of Fig. 5. After all the required information is provided and all tasks are executed, the PACA informs the user that the process is finished.



**Fig. 4.** Conversation with a PACA in a “Trip Planning” scenario.

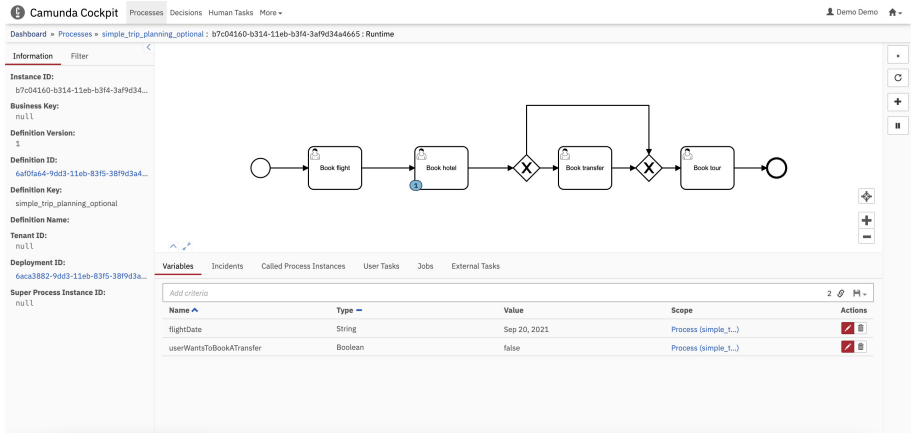


Fig. 5. Process instance shown in Camunda Cockpit after the “Book flight” task.

## 5 Conclusion and Future Work

In this paper, we have presented an approach and tool that embeds business process models into conversational agents. Besides having the potential to help users navigate their day-to-day processes, we believe this approach is a first step towards establishing a more conversational, chatbot-based strategy for supporting process execution. Future work might involve implementing more process examples as chatbots and a formal evaluation of the proposal, which might be based on user studies [1]. Other possible future endeavors include creating PACAs that support different processes and automatically recognize the process that the user wants to start, training RASA models automatically after parsing a BPMN process, and associating process mining methods with the PACAs.

## References

1. Cranshaw, J., et al.: Calendar.help: designing a workflow-based scheduling agent with humans in the loop. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 2382–2393 (2017)
2. Dang, J., Toklu, C., Hampel, K., Enke, U.: Human workflows via document-driven process choreography. In: 2008 International MCETECH Conference on e-Technologies (MCETECH 2008), pp. 25–33. IEEE (2008)
3. Dijkman, R., La Rosa, M., Reijers, H.: Managing large collections of business process models-current techniques and challenges. *Comput. Ind.* **63**(2), 91–97 (2012)
4. Dumas, M.: Fundamentals of Business Process Management. Springer, Berlin (2018). <https://doi.org/10.1007/978-3-662-56509-4>
5. Gnewuch, U., Morana, S., Maedche, A.: Towards designing cooperative and social conversational agents for customer service. In: ICIS (2017)
6. Helander, M.G.: Handbook of Human-Computer Interaction. Elsevier, Amsterdam (2014)

7. Toxtli, C., Monroy-Hernández, A., Cranshaw, J.: Understanding chatbot-mediated task management. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–6 (2018)
8. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Berlin, New York (2012). <https://doi.org/10.1007/978-3-642-28616-2>