A Business Process Model Driven Chatbot Architecture

Srđan Daniel Simić[1], Toni Starčić[1], Aldo Ferlatti[1], Darko Etinger[1], Nikola Tanković[1]

[1] Juraj Dobrila University of Pula, Faculty of Informatics, Pula
Croatia

ABSTRACT

A successful business operates on many interacting processes to meet the business goals; thus, organizing these processes in a structured business model is essential. Business process modeling notation (BPMN) is a widely used business modeling technique based on simple business and logic specifications understandable not only by domain experts but also to the broader audience. Although it provides support for the process execution, many BPMN models serve only for documentation purposes. This paper positions the lack of coordination between the process execution and the frontend interface as drivers of slow BPMN adoption. To close the gap between execution and interface, we present a system architecture that provides external task automation and interactive human task completion using a chatbot structure powered by natural language processing (NLP) and BPMN. We have evaluated the performance of the proposed architecture on the process of internship applications for the Faculty of Informatics in Pula, Croatia.

Keywords: business process, chatbot, BPMN

INTRODUCTION

Customers are one of the main drivers that sustain every business organization. Business processes inside the organization are the key factors that provide goods and services for customers to consume. They can be defined as functions in a specific sequence that delivers value for customers (Kirchmer, 2017). Some processes require customer interaction, but the majority are internal. To keep those processes organized, many decide to structure them into model diagrams. A standardized way for modeling business processes is using Business Process Modeling Notation (BPMN) (OMG, 2014). Even with well-structured models, sometimes it can be challenging to pair customer's wishes, and even employee's intentions, with a specific process that needs to be carried out to realize a business goal.

With the introduction of the BPMN 2.0 version, the standard enabled the execution of BPMN diagrams which provided an outstanding foundation to create and run complete application architecture for process-driven applications (Volker et al. 2014). These process-driven applications have succeeded even outside of business informatics and applied to domains like engineering (Schäffer et al. 2021).

This paper explores the possibility of user-oriented business environment automation by using a chatbot paired with a well-established BPMN standard. We propose an architecture that can help users navigate the vast sea of processes one business environment can offer by identifying their wishes and aligning them with specific processes. Our contributions are as follows: (1) Python BPMN engine system for parsing and executing BPMN models, compatible with the Camunda models, and (2) a Chatbot system capable of recognizing user's business intent through the conversation with the user.

Section II discusses related work. Section III gives an in-depth overview of our proposed architecture, divided into the BPMN and DMN engine modules and the chatbot modules. In Section IV, we evaluate the proposed architecture on the internship application process for the Faculty of Informatics in Pula. We conclude with a future work proposal.

RELATED WORK

There have been many proposals on how to include conversational interfaces with the business processes. Baez et al. (Baez et al. 2020) provided an overview of so-called chatbot integration into an existing system so that they principal highpoint differences in concepts, technology, and purpose across existing chatbots.

The paper identified 347 relevant chatbot systems, which are categorized broadly into eight categories or patterns. One of those is the Conversational business process, which is defined briefly as a chatbot that helps participants interact with the business process.

Kalia et al. (Kalia et al. 2017) presented a novel methodology called Quark that evaluates the BPMN process model and produces a Watson Conversation dialogue model to substitute a people-driven process with a chatbot. Chatbot then interacts with human actors with a goal that is necessary for business process completion. Interactions can be produced and translated into the IBM Watson Model. Telang et al. (Telang et al. 2018) continue the effort based on goals and commitments, and the work itself is a generic conceptual effort for a multi-domain chatbot engineering framework. Interactive automation in a chatbot is presented in a framework by Rizk et al. (Rizk et al. 2020a). The framework instructs on creating a multi-agent orchestration model and autonomous agents that automate not the entire business process but specific tasks within one. In contrast, the orchestration model manages and synchronizes those agents that are needed for task completion.

In the automotive domain, a software module prototype of a conversational workflow for a Chatbot has been constructed and modeled with Petri nets that can communicate and propose the most suitable tires for users (Colace et al. 2017). A multi-agent orchestration framework for task automation is presented by Rizk et al. (Rizk et al. 2020b). Capabilities of the framework are querying data with natural language, autonomous business process execution, sending notifications to users, and data visualization. Delicado et al. (Delicado et al. 2017) state that organizations keep parallel representations of business processes so that they can be understandable to everyone in the organization, e.g., a process can be described in textual representations or BPMN. The online platform they created, NLP4BPM, is a solution for the stated problem, and one of the main components of the solutions architecture is a web service that uses NLP for converting BPMN to text format. The platform also works analogously.

The methodology that transforms a business process model into a chatbot that guides a business process actor to a process completion through necessary steps is presented by Lopez et al. (López et al. 2019); in other words, users' questions about the process state can be answered by the system. The dialogue management module is script-based. Furthermore, finite state automata are created based on the BPMN structure and then expanded with additional states and transitions. Partial automation of decision-support chatbot development is proposed in a novel methodology (Estrada-Torres et al. 2021) in which DMN elements of the existing decision model are systematically transformed so that the low-level tasks a developer must do, get substituted or discarded.

In this paper, we present an architecture that is a combination of BPMN and NLP technology. BPMN is used as a standard for process modeling, parsing, and execution. One of the main components of the proposed system is the process engine, and similar concepts are mentioned here, where it is called the Workflow Manager (Colace et al. 2017) and here, where it is called the Orchestrator (Rizk et al. 2020). Although they are used as the chatbot state management modules, the same can be said for our process engine, which is, in a way, a chatbot state orchestrator or so-called manager. The chatbot, as mentioned above, is not task-oriented (Rizk et al. 2020a) but rather process-oriented, more generalized in a way. The Telegram Chatbot was developed to provide students with information about the schedule to achieve greater student efficiency (Priadko et al. 2019). Most congruent work includes both BPMN and chatbot (López et al. 2019), but the dialogue management is script-based, not like ours - state-based. Also, semantics is different. Their chatbot provides information about the process to the user, which ultimately helps with process completion. However, it is not the explicit interface to the tasks, or the process itself, like the one we propose in the architecture.

ARCHITECTURE PROPOSAL

This section presents the proposed business process model-driven chatbot architecture. A diagram of the proposed architecture can be observed in Fig. 1. It can be noticed that it is divided into two main modules, chatbot and BPMN and DMN module. The chatbot module consists of two submodules, intent recognition and model training, with the database of all interactions between users and the chatbot used by them. The second module consisting of BPMN/DMN modules uses the engine to parse and execute BPMN and DMN models from the organizations model repository.

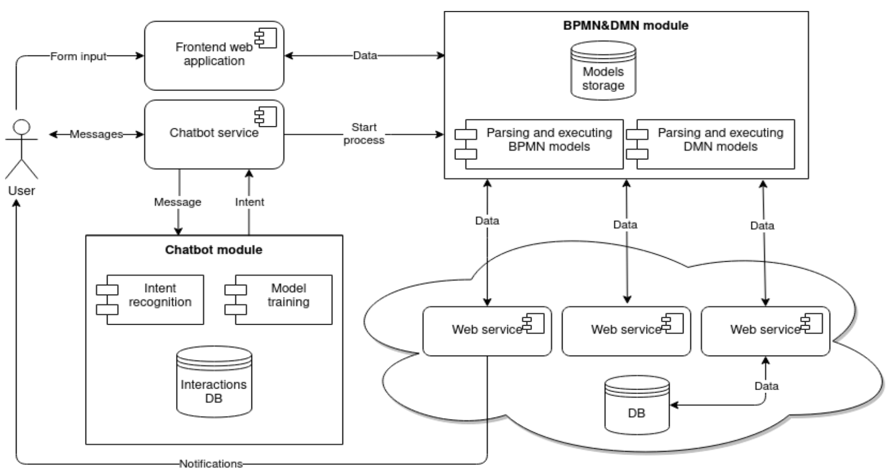

Figure 1. Proposed business process model-driven architecture

Besides those two main modules, the proposed architecture has two more essential components for user interaction with the system. Those two components are a frontend web application for User task completion in forms and a chatbot service responsible for natural language conversation with the user. In addition to those two components, the proposed architecture supports as many as needed additional web services that interact with the Python BPMN and DMN module to complete a specific business process. Those other web services may or may not interact with the user depending on their internal logic.

BPMN & DMN

We developed our own Python BPMN and DMN engine capable of parsing and executing BPMN and DMN models for the proposed architecture. Before parsing, every object in the BPMN model must have at least an *id* field specified inside the modeler. The engine can recognize and execute events, tasks, decision tables, sequence flow, and gateways at its current stage.

While the engine can recognize any task, at the moment, it contains particular logic for the execution of the following tasks:

User Task - *form fields* located inside the Forms tab in the Camunda modeler must be specified to allow user task execution. Each field must have an *id* and *type* with an optional parameter *label*. Validation and Property for each field can be added if there is a need for specific validation or some additional properties for a given field to ensure the quality of user-given input. After completing the user task, each form field will be saved into the engine process variables, where *id* would be the key, and the user input would be the value for the before mentioned key to be accessed throughout the process.

Service and Send Task - Connector implementation must be chosen in the Camunda general task tab to make them executable. Inside the Connector tab, id must be *HTTP-connector* and input parameters with variable names *URL* and *method* in which we specify the location of the web service and HTTP method we wish to use in that request. Additionally, variable *url_parameters* as type Map can be assigned for cases where it is needed. If we want to receive or send process variables from or into the Service or Send task, they must be specified inside the Input/Output tab. Input parameters are variables that will be sent with the request as JSON data. Expressions are supported to extract value from the process variable if the name is inside ${…}. Output parameters are used to store the resulting data from the services. Process variable names of the output parameters will filter response data and save just those values inside the process variables.

Call Activity Task - is used to start subprocess, which can be inside the running diagram or entirely new diagram. To make it executable, inside the General tab, *CallActivity Type* must be BPMN, and *Called Element* must be the process id of the process you wish to call as a subprocess.

Business Rule Task - can be implemented the same as the Service or Send task, but the primary intent is to implement it as the DMN model. Inside the *Decision Ref* id of the DMN model must be specified so that the engine knows which diagram to call. It is possible to save the output of the DMN model, which can be done by assigning the *Result variable* that will be dedicated to the process variables.

The main element in DMN is the decision table, and our engine is capable of parsing and executing it. A hit policy must be specified for each decision table together with input and output. The machine is capable of handling multiple decision tables in the diagram, given they are correctly set. Although DMN consists of few more elements that are currently irrelevant for executions and serve only for documentation purposes, they are not parsed by the engine.

CHATBOT

The chatbot service presents the first interaction with the system, which is dependent on the chatbot module. The chatbot's purpose is to give users structured answers and initiate processes that are handled by the BPMN and DMN modules. Currently, the chatbot module is composed of the *Interactions database* component, which stores records of historical conversations of the *Intent recognition model* and the *Model training* components that simultaneously exploit the database component. Accurate intent recognition is considered the first step for a sustainable discussion (Chai et al. 2006). In task-oriented systems, the importance of accurately recognizing the utterance intent becomes even more emphasized due to possible consequences caused by wrong initiated processes. Following are the detailed descriptions of individual components of the chatbot module.

Intent Recognition - following the above reflections, this component represents the main component of the module. For this paper, a dataset of conversational sentences was gathered, classified into four different intents. The intents categories correlate with the target process except for the out of domain intent, a con-trolled intent for non-supported intents. The model architecture comprises a BERT-based language-agnostic sentence encoder, LaBSE (Feng et al. 2020), and a single hidden classification layer with a heavy regularization dropout rate of 0.7. The model has a valuation accuracy of over 90%. The component gets the input message from the Chatbot service, which is then preprocessed through a pipeline of NLP techniques before being fed to the model.

Model training is used as a middle process for fine-tuning the intent recognition model on newly gathered data. The component monitors the database for increased records of wrongly classified intents and inputs classified as out of domain intent, which triggers the component to start the fine-tuning process of the intent recognition component on new data. As the model develops, the category out of domain will be classified less often because of the increased supported intents by the system. For future uses, this module will also be used for fine-tuning other models.

Interactions database is the core of the data structure for the chatbot module. It is composed of all the historical conversation sentences conducted through the chatbot service and predicted intentions from the Intent recognition component. This component aims to gather valuable unstructured data for future improvement and development of new models.

EVALUATION

In recent years, there have been many attempts to use chatbots in higher education to automate and improve various processes such as university admission (El Hefny et al. 2021) (Santoso et al. 2018) monitoring academic records (Heryandi, 2020), and scheduling (Kirchmer, 2017), to name a few. Encouraged by these insights, we have decided to evaluate our proposed systems on the Faculty of Informatics in Pula. Fig. 2. depicts the primary process model, and as it can be seen, it is divided into two parts. The first part is finding the user's intent through the chatbot's conversation with the user. The second part is mapping the user's intent to a specific subprocess which will then begin afterward.

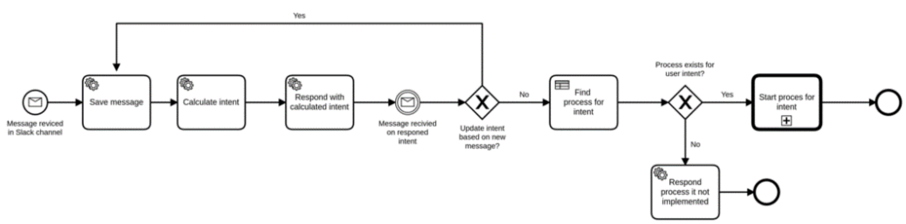

Figure 2. The primary BPMN process for model-driven chatbot

The primary process begins with the posted message in a Slack channel that mentions the chatbot. That marks the beginning of the conversation loop with the user. The loop consists of three service tasks. First, the message is saved to the conversations database, then the user's intent is calculated, followed by the chat-bot's response based on its prediction. The conversation loop ends with the user's affirmative response to the chatbot's predicted intent. With the correct user's in-tent process continues to the business rule task, implemented as a DMN table with first hit policy, in which all possible intents are mapped to the paths of BPMN models for them. If the subprocess for the user's intent is not yet implemented, the chatbot notifies the user and ends the process. Otherwise, the process flows to the call activity task that starts the subprocess, and with its completion, the primary process ends.

For the subprocess, we chose to model the process of internship applications for the Faculty of Informatics in Pula. BPMN model of the before mentioned process consists of three lanes and various tasks coupled with business logic needed to complete the process. Those three lanes are student, professor, and employer. Student and employer lanes consist of User tasks, where the employer's lane has one while the student's lane has three with the addition of start and end events. A web application has been developed to perform User tasks. The web application creates a form based on the form fields described inside the User task. Professor's lane is more complex, having five Service and Send tasks and one Manual and User task. Service tasks are divided into database web service, Airtable connector service, and Portable Document Format (PDF) web service, which has been developed for editing PDF files needed for process completion. Send tasks were used solely to notify participants of their upcoming tasks. Two methods for notification were implemented into two services. The first one being email service and the second via a chatbot in the Slack channel. Additional file storage web service was developed that doesn't directly tie with the BPMN model, meaning it wasn't specified in any Service or Send tasks. File storage service was used by other web services presented in the model to store files on it and, in return, get only the URL location of the file to be stored as the process variables. With that service, the engine uses less memory because only a URL string would be saved to process variables instead of a whole file. If files are needed for other tasks, the web service would just download the required file from the URL and continue the process.

CONCLUSION

In this paper, we have presented a system architecture consisting of a Python BPMN system for parsing and executing existing BPMN models together with a chatbot system with the capability for starting the correct process on the before mentioned engine by correctly identifying the user's business intent. We have evaluated our system architecture on the practical problem of internship applications on the Faculty of Informatics in Pula. The benefits of using the presented architecture are in business environments consisting of many business process models, which depend on the users that may or may not be part of the organization and their intent to start a specific process through human-like interaction with the chatbot. Hence, the main benefit is making the business environment more autonomous.

In future research, we plan to extend our Python BPMN engine so it can parse and execute every standard BPMN object and implement an access control list (ACL) for our engine. Furthermore, we will improve the chatbot system by implementing a dialog policy and adding more supported intents to provide a more natural interaction during the conversation with a user.

ACKNOWLEDGEMENTS

This work has been fully supported by Croatian Science Foundation under the project IP-2019-04-4216.

REFERENCES

A. Heryandi. "Developing Chatbot For Academic Record Monitoring in Higher Education Institution". In: IOP Conference Series: Materials Science and Engineering 879 (Aug. 2020), p. 012049. doi: 10.1088/1757-899x/879/1/012049. url: https://doi.org/10.1088/1757-899x/879/1/012049.

Andrii O. Priadko et al. "Development of a chatbot for informing students of the sched-ule". In: CEUR-WS, Nov. 2019, pp. 128–137. A. Heryandi. "Developing Chatbot For Academic Record Monitoring in Higher Education Institution". In: IOP Conference Series: Materials Science and Engineering 879 (Aug. 2020), p. 012049. doi: 10.1088/1757-899x/879/1/012049. url: https://doi.org/10.1088/1757-899x/879/1/012049.

A. K. Kalia et al. "Quark: A methodology to transform people-driven processes to chatbot services". In: 10601 (2017), pp. 53–61. doi: 10.1007/978-3-319-69035-3_4.

Bedilia Estrada-Torres, Adela Del-Río-Ortega, and Manuel Resinas. “Dialogbased Auto-mation of Decision Making in Processes”. In: (2021). arXiv: 2109.00822. url: http://arxiv.org/abs/2109.00822.

Colace et al. "BotWheels: A petri net based Chatbot for recommending tires". In: DATA 2017 - Proceedings of the 6th International Conference on Data Science, Technology and Applications. 2017, pp. 350–358. isbn: 9789897582554. doi: 10.5220/0006491903500358.

Eike Schäffer et al. "Process-Driven Approach within the Engineering Domain by Com-bining Business Process Model and Notation (BPMN) with Process Engines". In: Proce-dia CIRP 96 (2021). 8th CIRP Global Web Conference – Flexible Mass Customisation (CIRPe 2020), pp. 207–212. issn: 2212-8271. doi: https://doi.org/10.1016/j.procir.2021.01.076.url: https://www.sciencedirect.com/science/article/pii/S2212827121001037.

Fangxiaoyu Feng et al. "Language-agnostic BERT Sentence Embedding". In: (2020). url: http://arxiv.org/abs/2007.01852.

Heru Agus Santoso et al. "Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services". In: 2018 International Seminar on Application for Technology of Information and Communication. 2018, pp. 417–423. doi:10.1109/ISEMANTIC.2018.8549797.

Joyce Chai, Chen Zhang, and Tyler Baldwin. "Towards conversational QA: automatic identification of problematic situations and user intent". In: Proceedings of the COLING/ACL on Main conference poster sessions (2006).

Luis Delicado et al. "NLP4BPM - Natural language processing tools for business process management". In: CEUR Workshop Proceedings 1920 (2017). issn: 16130073.

Marcos Baez et al. "Chatbot integration in few patterns". In: IEEE Internet Computing 7801.c (2020). issn: 19410131. doi: 10.1109/MIC.2020. 3024605. arXiv: 2009.03101.

Mathias Kirchmer. High Performance Through Business Process Management. Springer International Publishing, 2017. doi: 10.1007/978-3-319-51259-4. url: https://doi.org/10.1007/978-3-319-51259-4.

OMG, Business Process Model and Notation (BPMN) 2.2 Specification. Version 2.2. 2014. url: https://www.omg.org/spec/BPMN.

P. R. Telang et al. "A Conceptual Framework for Engineering Chatbots". In: IEEE Internet Computing 22.6 (2018), pp. 54–59. doi: 10.1109/MIC.2018.2877827.

Volker Stiehl, Rachel Raw, and Paul Smith. Process-driven applications with BPMN. Springer, 2014.

Walid El Hefny et al. "Jooka: A Bilingual Chatbot for University Admission". In: Trends and Applications in Information Systems and Technologies. Ed. by Álvaro Rocha et al. Cham: Springer International Publishing, 2021, pp. 671–681. isbn: 978-3-030-72660-7

Yara Rizk et al. "A Conversational Digital Assistant for Intelligent Process Automation". In: Lecture Notes in Business Information Processing 393 LNBIP (2020), pp. 85–100. issn: 18651356. doi: 10.1007/978-3-030-58779-6_6.

Yara Rizk et al. "A Unified Conversational Assistant Framework for Business Process Automation". In: (2020). arXiv: 2001.03543. url: http://arxiv.org/abs/2001.03543.