

Evaluating and Comparing Generative-based Chatbots Based on Process Requirements

by

Luis Fernando Lins dos Santos

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

© Luis Fernando Lins dos Santos 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Business processes refer to the sequences of tasks and information flows needed to achieve a specific goal. Such processes are used in multiple sectors, such as healthcare, manufacturing, banking, among others. They can be represented using diverse notations such as Event-driven Process Chain (EPC) and Yet Another Workflow Language (YAWL). A popular standard notation for modelling business processes is the Business Process Model and Notation (BPMN), due to its breadth of features and intuitiveness. Despite organizations increasingly turning to automated processes to enhance efficiency and precision, business processes might still contain human-dependent tasks, which present several challenges, such as complex task orders, task dependencies, and the need for contextual adaptability. These challenges often make it difficult for process participants to understand process execution, and it is common for them to get lost in the process they are trying to execute.

Since the advent of process modelling languages, there have been multiple strategies to help users execute business processes. Most recently, chatbots, programs that allow users to interact with a machine using natural language, have been increasingly used for process execution support. A recent category of chatbots worth mentioning is generative-based chatbots, powered by Large Language Models (LLMs), which are trained on billions of parameters and support conversational intelligence.

However, several challenges remain unaddressed: (i) how to incorporate process knowledge into generative-based chatbots; (ii) how to evaluate if a generative-based chatbot is meeting the requirements of BPMN constructs for process execution support; (iii) how different generative-based chatbot models compare in terms of meeting the requirements of BPMN constructs. To address these challenges, this thesis presents an exploratory approach to evaluate and compare generative-based chatbots in terms of meeting the requirements of BPMN constructs. This research comprises three distinct phases. The first phase presents a literature review that examines how chatbots are used in conjunction with business processes. In the second phase, the focus shifts to retrieval-based chatbots and their potential for process execution support when maintaining direct communication with a process engine system, in an attempt to achieve a gold standard of process-aware chatbots. Finally, the third phase delves into the realm of using generative-based models as process-aware chatbots to enhance business process execution support, including an evaluation of how two distinct generative models (GPT and PaLM) perform when helping in the execution of two business processes (Trip Planning and Wedding Planning).

This thesis makes several contributions. First, it offers a comprehensive exploration of the challenges and gaps in the existing literature regarding the development of process-

aware chatbots. Second, it provides an exploratory study about the use of generative-based chatbots for process execution support that meets the requirements of BPMN constructs. Finally, through comparative qualitative and quantitative evaluations, it sheds light on the performance of prominent generative models, GPT and PaLM, in the context of process execution support, contributing valuable insights for future research and development in this evolving field.

Acknowledgements

I would like to thank first and foremost my mother, Maria Lúcia, who always sees the positive side of life and helps me navigate through all difficulties with her wise words. To my grandmother Lúcia, caring and tender, who was by my side at all times, assisting in my education and personal growth.

To my aunts Simone and Silvia, who have always cheered for me and believed in my potential, and my grandparents Margarida and Ozanir, who taught me from a young age the importance of education for my development.

To my supervisor Dr. Paulo Alencar, who saw the potential of this work and welcomed me to pursue my research with him at the University of Waterloo, and Dr. Toacy Oliveira, who was my supervisor during my undergraduate program and made it possible for me to be completing my Master's program today.

I also thank all the professors at the University of Waterloo who contributed to my education, introducing me to new fields within Computer Science and encouraging rich discussions about various topics.

To my friends, including Glaucia, Nathalia, Carol, Giulia, and so many more, I thank you for your affection, laughter, and the strength you provided me throughout this program.

Finally, I thank the Universe, which, in its beautiful and incomprehensible turmoil, granted me the opportunity and privilege to study at this University, have incredible experiences, and meet special people whom I will carry with me for the rest of my life.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	v
List of Figures	x
List of Tables	xi
List of Abbreviations	xii
1 Introduction	1
1.1 Motivation	1
1.2 Related Areas	2
1.2.1 Business Processes	2
1.2.2 Chatbots	5
1.3 Challenges	8
1.4 Research Questions	9
1.5 Research Design	10
1.5.1 Phase I: Literature Review	11
1.5.2 Phase II: Retrieval-based Process-Aware Chatbots	12

1.5.3	Phase III: Generative-based Process-Aware Chatbots	12
1.6	Thesis Structure	13
2	Literature Review	16
2.1	Research Method	16
2.1.1	Utilized sources and results	16
2.1.2	Research Questions	18
2.2	Paper Summaries	19
2.2.1	Process-aware chatbots with no standardized notation	20
2.2.2	Using chatbots to model business processes	25
2.2.3	Process-aware chatbots with standardized notation	27
2.2.4	Evidence-based studies about chatbot usage	29
2.3	Results and Critical Analysis	30
2.3.1	RQ1: Which chatbot frameworks are utilized in the papers?	30
2.3.2	RQ2: What design techniques were utilized for the development of the chatbots in the found papers?	31
3	Retrieval-based process-aware chatbots	34
3.1	Implementation Strategy	35
3.2	Use Case 1: Trip Planning	37
3.2.1	Process Model	37
3.2.2	Chatbot Implementation	38
3.3	Use Case 2: Wedding Planning	39
3.3.1	Process Model	39
3.3.2	Chatbot Implementation	41
3.4	Lessons Learned	42
4	Generative-based process-aware chatbots: Initial approach	45
4.1	Implementation strategy	45
4.2	Use Case: Trip Planning	47

5	Generative-based process-aware chatbots: Improved approach	52
5.1	Implementation Strategy	52
5.2	Comparison Methodology	54
5.2.1	Category 1: Start Event	55
5.2.2	Category 2: Forward Flow	56
5.2.3	Category 3: Previous History	57
5.2.4	Category 4: End Event	58
5.2.5	Category 5: Process Decision Points	59
5.2.6	Category 6: Unintended paths	59
5.3	GPT	60
5.3.1	Use Case 1: Trip Planning	60
5.3.2	Use Case 2: Wedding Planning	69
5.4	PaLM	76
5.4.1	Use Case 1: Trip Planning	76
5.4.2	Use Case 2: Wedding Planning	82
5.5	Use Case Comparison	86
5.5.1	Qualitative Comparison	87
5.5.2	Quantitative Comparison	89
5.5.3	Evaluation Conclusions and Limitations	89
6	Conclusion and Future Work	92
6.1	Conclusion	92
6.2	Future Work	93
6.2.1	Possible Integration with Camunda Engine	93
6.2.2	Evaluation Criteria	94
6.2.3	Process Modelling	94
6.2.4	Additional BPMN Constructs	95
6.2.5	Utilized Chatbot Models	95

References	96
APPENDICES	102
A BPMN XML documents	103
A.1 Trip Planning BPMN XML document	103
A.2 Wedding Planning BPMN XML document	106
B Literature Review Search Results	112

List of Figures

1.1	Visual description of the three-phase research design of the current study.	15
2.1	An example of a Delta Airlines support conversation conducted by Kecht et al.'s chatbot.	21
3.1	BPMN diagram for the “Trip Planning” process.	37
3.2	Process instance shown in Camunda Cockpit after the “Book flight” task.	39
3.3	Conversation with a process-aware chatbot in a “Trip Planning” scenario.	40
3.4	BPMN diagram for the “Wedding Planning” process.	42
3.5	A complete conversation with the “Wedding Planning” bot.	43

List of Tables

4.1	BPMN elements extracted with “bpmn-moddle”	47
5.1	Fulfillment of each evaluation question (EQ) in each of the four test cases .	88
5.2	Percentage of fulfillment of EQs by each chatbot (GPT and PaLM) for each test case (Trip Planning and Wedding Planning)	89
B.1	Studies retrieved when using our selected search query	112

List of Abbreviations

AI Artificial Intelligence 19, 29, 33, 35

BPMI Business Process Management Initiative 3

BPMN Business Process Model and Notation 2–5, 10–14, 16, 19, 20, 22–24, 26–28, 33–35, 37, 42, 45–60, 64–66, 69, 70, 73, 75, 76, 81, 82, 86, 87, 89, 90, 92–95

CMMN Case Management Model and Notation 3

DMN Decision Management Notation 3

EPC Event-driven Process Chain 3, 4, 20

GPT Generative Pre-Trained Transformer 7, 8, 12–14, 45, 49–52, 54, 60, 69, 76, 82, 83, 87, 89, 90, 92, 93, 95

GUI Graphical User Interface 38

IoT Internet of Things 5

IPA Intelligent Personal Assistants 5

LLM Large Language Model 7, 10, 16, 32, 52, 92

NLP Natural Language Processing 33

NLU Natural Language Understanding 6, 7, 22, 30, 31, 35, 36, 38, 42

OMG Object Management Group 3

OPT Open Pre-trained Transformers 8, 95

PACA Process-Aware Conversational Agent 27

PaLM Pathways Language Model 8, 13, 14, 45, 51, 52, 54, 76, 77, 81, 82, 86, 87, 89, 90, 92, 93, 95

S-BPM Subject-oriented Business Process Management 11, 19, 26

SDK Software Development Kit 36

UML Unified Modeling Language 25

VR Virtual Reality 28

WFMS Workflow Management Systems 4, 27, 34

XML Extensible Markup Language 13, 46, 47, 52–54, 60, 69, 70, 76, 82, 94

YAWL Yet Another Workflow Language 3, 4

Chapter 1

Introduction

1.1 Motivation

Processes are pervasive in personal and organizational settings. They can support a wide variety of tasks, ranging from scheduling a trip [49] to hiring new employees, admitting students into a university program to producing and delivering a product [17]. 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 [14]. 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 [16]. Owing to these challenges, process participants should have assistance to guide them through process instances.

One possible solution to help guide users through business processes are chatbots, also known as conversational agents. Chatbots are an emerging technology that has become present in various domains, from healthcare to customer service [31]. This technology's main selling point is that chatbots are able to talk to humans, understand their needs,

and respond in a coherent way, while also being available 24/7. Many businesses and academic projects have been using chatbots as a valuable tool, but there are still some challenges in the chatbot field that remain to be solved in an effective way. Some of these challenges include personalized responses based on user context, analysis of customer behaviour from interaction history, sentiment and emotion analysis, and negotiation without human intervention [10] [44].

Furthermore, chatbots can be very unpredictable, deciding conversation paths by itself that do not always correspond to the business's intention, especially when the chatbot is machine-learning-based [48]. Therefore, another challenge in the development of chatbots is creating one that closely follows the business requirements. For this reason, this work has the goal of analyzing approaches to building and using chatbots that can help users execute business processes while strictly adhering to specific business process requirements, based on constructs that are laid out by the Business Process Model and Notation (BPMN). Over the next few sections, we will introduce the related areas, challenges, research questions, research design and thesis structure that we used to achieve this goal.

1.2 Related Areas

Because our plan is to build chatbots that can help users execute business processes, we will introduce next some context about two of the major components of our solution, the first one being the business process domain and the second being the chatbot domain. For business processes, we will explain some of the common notations that exist in the literature and industry for representing business processes, and then introduce some specific concepts about BPMN, which was the chosen notation for the current work. Subsequently, for chatbots, we will cover the multiple classifications of chatbots that exist, and focus on two particular categories – retrieval-based and generative-based chatbots – that are the ones that will be further explored throughout this thesis.

1.2.1 Business Processes

Every business executes multiple activities daily that need to be organized. A good example that showcases the importance of process modelling can be seen in the chemical industry. In chemical production, whether it involves mixing chemicals to create a reactor or managing an existing reactor, it is crucial to stick to well-defined processes in order to avoid disasters [12].

In the case of technical projects, to accomplish the goal of modelling the processes related to a specific project, a process analyst may be required to create a workflow that will serve as a foundation throughout the project's life cycle. This workflow will contain strategic decisions, tasks, and technological aspects of the project. Thus, it is crucial that this workflow is built using an effective methodology. Previous works such as [49] and [17] establish a shared understanding of business process concepts and technologies. These resources cater to both business administrators and computer scientists, who often possess different scopes of knowledge.

For a while, businesses employed flowcharts to represent their processes graphically, which emerged from a loosely standardized notation system developed in the 1920s. However, in 2004, the Business Process Management Initiative (BPMI) – later merged with the Object Management Group (OMG) – introduced a more comprehensive notation called BPMN. The purpose behind BPMN was to create process representations that could be readily understood by various users, including business analysts, technical developers, and business users. Currently, BPMN is at version 2.0, and it serves a structured notation for developing graphic representations of business operations.

Since the release of BPMN, two new standards were released in 2014: Case Management Model and Notation (CMMN) and Decision Management Notation (DMN), both developed by OMG, the same organization that maintains the BPMN. With CMMN, processes that are more weakly structured can be represented, because instead of defining a rigorous process, it defines a Case, which is a high-level activity composed by actions that should be taken towards a known goal, but that does not specify the path to achieve this goal [50]. However, because of this lack of structure, CMMN is less suitable for presenting workflows that have a fixed order and has fewer functions than BPMN. For instance, it is not easy to perform task synchronization or control flow pattern with CMMN, except for the case when a workaround is used.

Meanwhile, DMN is a notation for decision handling and representation [35]. DMN can be used to model decisions and their requirements, and to perform decision automation. However, since DMN is only designed for decisions, it cannot be used to represent a workflow or model a process. It is expected that eventually, these three notations, given that they are all maintained by OMG, will be combined. However, for the time being, we decided to focus solely on BPMN because it allows for a complete modelling of a business process, and includes support for more features, such as messaging.

Some other alternatives to modelling business processes in addition to BPMN are Event-driven Process Chain (EPC) and Yet Another Workflow Language (YAWL). Both of these are also process modelling languages used in the field of business process management.

However, EPC has fewer features than BPMN and is more strongly focused on events and functions. Nowadays, it is mostly used only in German-speaking regions [45]. On the other hand, YAWL is more technical and less graphical than BPMN, and therefore, a considerable programming effort is required for it to be used. According to Neumann et al., compared to YAWL, BPMN proves itself to be far more intuitive to use [34]. Therefore, we decided to use BPMN for this thesis since it proved to be the most widespread and intuitive notation among all the available options.

To facilitate the execution of modelled business processes, Workflow Management Systems (WFMS) – also known as “process engines” – are a type of tool that has gained prominence over the years. These systems assist in managing and tracking the state of activities within a workflow, enabling the smooth flow of tasks, information, and events. Since most WFMS utilize the BPMN standard, they share numerous process concepts. In this work, we have selected Camunda¹ as our chosen WFMS for this work due to its open-source nature and widespread usage within our research group.

BPMN concepts

BPMN processes can have a wide variety of elements, but in this thesis, we will cover a few of the most common ones. The concepts we will cover here are: (1) events, (2) tasks, (3) sequence flows, and (4) gateways.

An *Event* is when something relevant to the process happens transiently, i.e. with no duration in time [17]. The *Start and End events* are two examples of those that are most frequently present and that always occur atomically. In Camunda, the *Start Event* is what triggers the execution of a new process instance, while the *End Event* is what indicates the end of a running process instance.

Throughout the process, there are *tasks* that are executed by the chatbot. They are pieces of work that take a certain amount of time and need to be completed for the process to continue. A few sub-types of *tasks* are *User Tasks*, *Manual Tasks*, and *Service Tasks*. In common BPMN processes, *Service Tasks* are executed automatically by a running back-end server, while *Manual Tasks* rely solely on physical human execution – without the use of software – such as filling out a paper document. *User Tasks* are similar to *Manual Tasks* as in both require an external execution; they are not executed automatically. However, *User Tasks* are still supposed to be completed online by a human being, while *Manual Tasks* are usually performed offline.

¹<https://camunda.com/>

Besides events and tasks, a typical process can also include *Gateways*, which are spots where the process flows can diverge or converge, i.e., where the process flow can go in more than one direction. *Gateways* can be of many different types, such as *Parallel*, *Exclusive*, *Inclusive Gateways*, and more. For this work, we will only be focusing on *Exclusive* and *Parallel Gateways*. This choice was made because, from our understanding, these two types of gateways cover most of the business process scenarios that we wanted to explore. However, the other types of gateways presented in the BPMN standard could also be used in the model and would work with our proposed solution. *Exclusive Gateways* are a type of gateway that uses a specified condition to determine which of the next tasks are available to be executed. When reaching the gateway, each condition is evaluated, and even when multiple paths are available, the execution can only follow one of them. On the other hand, *Parallel Gateways* can be used to create parallel flows where all the outgoing process flows are available.

Finally, we have *Sequence Flows*, which are represented as a solid line with an arrowhead. They are used to connect elements in the BPMN diagram – e.g., connecting two tasks, or connecting a task to a gateway – and they present the order in which the elements should be traversed. It may also contain additional information about the execution of each flow. For instance, when a process has an Exclusive Gateway, its outgoing sequence flows will contain the conditional expression that will be evaluated for each path by the process engine.

Previous work has shown that the usage of BPMN improves the relationship between business and IT fields inside a company [43]. Therefore, it seemed like a good idea to use this notation for the development of a process execution support chatbot, given that this kind of chatbot will exist at an intersection between the company’s IT and business teams – e.g. it is developed by IT employees but contains knowledge that comes from business personnel.

1.2.2 Chatbots

The popularity of chatbots, also known as conversational agents, has been on the rise since the introduction of Intelligent Personal Assistants (IPA) like Siri, Alexa, Google Assistant, and Cortana. This trend has further escalated with the increasing adoption of IPA devices such as Amazon Echo and Google Home [7]. These personal assistants possess the ability to fulfill various user requests, ranging from checking the weather and setting alarms to playing music and interacting with Internet of Things (IoT) devices like smart lamps and smart locks [11].

According to Gnewuch et al. [22], chatbots can be classified based on their primary mode of communication (voice-based, text-based, or embodied) [28], or their context, indicating whether they have a general or task-oriented goal. Evaluating a chatbot often involves considering its usability, encompassing efficiency, effectiveness, and user satisfaction [39]. Notably, chatbots that excel in these three quality attributes tend to maintain coherent and engaging conversations.

Another chatbot classification is presented by Ramesh et al., who define that chatbots can be classified as either retrieval-based or generative-based [40]. Retrieval-based chatbots utilize a set of pre-existing responses and employ a heuristic approach to select an appropriate response that matches the input and context. The selection of an appropriate response can involve multiple heuristics, ranging from simple rule-based expression matching to more sophisticated techniques involving combinations of Machine Learning classifiers. As it is possible to realize, these systems do not generate novel responses, but rather choose from a predefined pool of responses.

Retrieval-based chatbot models

Recent works, such as Iris, a chatbot developed by Fast et al. [18], have demonstrated the potential for conversational agents to handle even complex mathematical questions. Another example of the potential of chatbots is introduced by Devy, developed by Bradley and colleagues [9], which focuses on supporting software engineers with a voice-based conversational assistant that leverages context elements specific to software development workflows.

While some conversational agents, like Iris [18] and Devy [9], are built from scratch, other chatbots are built using chatbot frameworks, so that the developers can avoid the challenges of starting from scratch. The industry offers several conversational agent frameworks, including Google’s Dialogflow², IBM Watson³, Facebook’s Wit.ai⁴, as well as open-source options like Rasa⁵ and Botpress⁶. These frameworks provide pre-existing logic and interfaces, streamlining the development process.

Rasa is already used by thousands of developers worldwide [8]. This framework is composed of a pair of tools, the Natural Language Understanding (NLU) component and Rasa Core, which are open-source python libraries for building conversational software.

²<https://dialogflow.com/>

³<https://www.ibm.com/watson/>

⁴<https://wit.ai/>

⁵<https://rasa.com/>

⁶<https://botpress.com/>

While Rasa NLU is responsible for performing natural language understanding, to make sense of users' inputs, Rasa Core is responsible for dialogue management. That means that when a user sends a message, the NLU component will try to understand and categorize that message based on a series of sentence examples. Then, after it has been correctly identified, Rasa Core will come in to determine the next action the chatbot should take, by selecting the most appropriate response from a list of possible actions. In Rasa, the two components that perform language understanding and dialogue management are fully decoupled. This allows Rasa NLU and Core to be used independently of one another, and allows trained dialogue models to be reused across languages [8]. Because of its open-source status and its widespread usage, Rasa was the framework that we chose to build some of the chatbots in the current study.

Generative-based chatbot models

Although retrieval-based chatbots built the foundations of what we know as chatbots today, they present some challenges, as they are usually able to support only a limited context and a few predefined responses. Generative models overcome this lack of flexibility by generating new responses through the use of Large Language Models (LLMs). These models often leverage machine translation techniques, although instead of translating between languages, they translate an input into a response after acquiring knowledge from the training set. By utilizing this approach, generative models have the capability to create unique and contextually-relevant responses that are not limited to a predetermined set.

The Generative Pre-Trained Transformer (GPT) model, in particular, is a large language model developed by a research laboratory called OpenAI and it is capable of generating text that is virtually indistinguishable from natural human language [30]. In February 2019, OpenAI announced GPT-2, an unsupervised language model with 1.5B parameters trained on 40GB of text (about 10B tokens). This model was able to generate very reasonable output, but it also generated content that did not make much sense. In any case, it generated a lot of interest and media attention given the similarity to human language that it already showcased.

Then, in June 2020, OpenAI announced GPT-3, a new language model with 175B parameters (over 10 times larger than GPT-2) trained on a set of 499B tokens of web content, which made it the largest language model in existence up to its release. Just like GPT-2, the model itself was not made available, due to security concerns, however, access to it was provided via an API.

The GPT model has a few limitations, such as the fact that its outputs may lack

semantic coherence, resulting in text that is gibberish, or may correspond to assertions that are not true [13]. According to Dale, GPT-3’s credibility is compromised by the fact that it is disconnected from the truth, which makes it risky to use for question-answering or advice-giving systems, where it is important that the result be true.

However, to counteract this problem, there has been an increase in GPT-based applications where a human stays in the loop, which makes them much safer. Most of these applications are augmented writing tools that take a user’s textual input and provide an alternative version of that input, either longer or shorter depending on the application in question. Therefore, GPT is generating an alternative textual input but a human must still review it to make sure it is sound. This is a problem if GPT is, for instance, translating a document from or to a language that the reviewer does not understand, but can be quite useful in other scenarios where the reviewer fully comprehends both the input and output of the GPT model.

More recently, in March 2023, GPT-4 was introduced, and OpenAI has kept working on ways to improve its accuracy and functionality. However, in addition to GPT, there are other large language models being developed in the industry by big companies. Meta, for example, developed the Open Pre-trained Transformers (OPT) model, its large language model that ranges from 125M to 175B parameters. Their research team showed that OPT-175B is comparable to GPT-3, while requiring only 1/7th the carbon footprint to develop [52]. Meanwhile, Google has developed the Pathways Language Model (PaLM) and PaLM 2, the first one with 540B parameters [5] and the second one with 340B parameters [53]. Interestingly, PaLM 2 is quite smaller than the largest PaLM model but uses more training computation. According to Google’s research team, PaLM 2 outperforms PaLM on a variety of tasks, including natural language generation, translation, and reasoning, and achieves results competitive with GPT-4, which suggests that model scaling is not the only way to improve performance [3].

In this thesis, we have decided to explore GPT and PaLM for our use case of process execution support, due to the prominence and availability of these models. We will use PaLM 2 – the most recent version of PaLM – and GPT-3.5 – given that GPT-4 still has limited access.

1.3 Challenges

After a systematic literature review, detailed in Section 2, three challenges (C1, C2 and C3) have emerged. This review not only served as a foundation for our research, but

also highlighted the pressing need to address these challenges. Each challenge represents a different aspect of the intricate landscape of chatbot development and integration with business processes, capturing the unique issues and opportunities that researchers and developers encounter. By shedding light upon these challenges, we aim to contribute valuable insights that can drive advancements in the field of conversational AI.

C1: Incorporating process knowledge into generative-based chatbots

One of the primary challenges in this research domain is the need to bridge the gap between generative-based chatbots and process knowledge. There's a lack of methodologies for making chatbots process-aware, especially in the case that the chatbot is generative-based. The challenge lies in developing effective approaches that help the chatbot understand the process structure and requirements and can respond appropriately.

C2: Developing comprehensive evaluation criteria for process requirements alignment

Another significant challenge is the development of comprehensive evaluation criteria to assess the extent to which a chatbot adheres to the requirements of process constructs. Crafting appropriate evaluation questions that accurately evaluate the chatbot's compliance with process principles, while considering the intricacies of different process models, is a challenge. These metrics must be both objective and applicable to diverse process-aware scenarios.

C3: Comparing different generative models in terms of their alignment to process requirements

The challenge here is to conduct a meaningful comparative assessment of different generative models, such as OpenAI's GPT-3.5 and Google's PaLM 2. There's a need to compare and determine which model better aligns to process requirements for our specific use cases of business process execution support, and this comparison should follow a careful experimental setup to demonstrate the unique strengths and limitations of each model, ensuring a fair comparison.

1.4 Research Questions

To address the challenges from the previous section and the objectives of this study effectively, we have formulated a set of key research questions to guide our investigations. These research questions serve as the foundation upon which we build our analysis and

comparative evaluations in subsequent sections. They were thought to provide comprehensive insights into the development and performance of process-aware chatbots, determining the best approach to integrating process knowledge into chatbots and evaluating the capabilities of generative-based chatbot models like GPT-3.5 and PaLM 2. The proposed research questions are:

RQ1: What are some of the possible approaches to provide a generative-based chatbot with the process knowledge that is contained in a BPMN process model?

In this research question, our goal is to conduct experiments with various methods for introducing process knowledge into a generative-based chatbot, and investigate which approach best follows the restrictions inherent in BPMN process models. By performing this investigation, we aim to discover the unique challenges and benefits associated with each approach to integrating process knowledge into an LLM chatbot.

RQ2: How can we evaluate if a generative-based chatbot is meeting the requirements of BPMN constructs for process execution support?

For this research question, we will develop a set of general evaluation questions to assess the alignment of a generative-based chatbot with the requirements of BPMN constructs. These questions will enable us to determine the extent to which each chatbot adheres to the principles and guidelines laid out in BPMN process models.

RQ3: How does GPT-3.5 compare with other generative-based chatbot models, such as PaLM 2, in terms of meeting the requirements of BPMN constructs?

This research question explores the comparative strengths and weaknesses of OpenAI’s GPT-3.5 in relation to another prominent generative-based chatbot model, Google’s PaLM 2, by submitting the same prompts to both models and analyzing their prompt understanding and responses. Through the evaluation of these models based on the evaluation questions elaborated in RQ2, we aim to identify which model is better suited for our particular use cases, in the broad context of business process execution support.

1.5 Research Design

In this thesis, we use various empirical methods to answer the three research questions laid out in Section 1.4. More specifically, we follow a mixed-methods approach, starting with

a literature review phase, then going through the creation of a retrieval-based process-aware chatbot connected to a process engine, and finally, introducing process-awareness into generative-based chatbots without the need for a process engine. These three phases, together, constitute this study. The phases of this study are presented in Fig. 1.1.

1.5.1 Phase I: Literature Review

In the first phase of our research study, we performed an extensive literature review to explore the integration of chatbots with business processes. To ensure the comprehensiveness of our literature review, we employed a systematic approach, utilizing multiple reputable databases, including IEEE Xplore, ACM Digital Library, and Scopus.

Our designed search query, further detailed in Chapter 2, encompassed studies published between 2021 and the current year – 2023 –, and resulted in a substantial number of research articles and industry reports across the selected databases. As of the search date, July 26, 2023, our search resulted in 10 papers and 1 book from IEEE Xplore, 35 research articles from ACM Digital Library, and 33 papers from Scopus, including conference papers and articles. Six of the 33 papers from Scopus had already been identified through IEEE Xplore and were subsequently excluded. Therefore, our final count of unique papers sourced from Scopus amounted to 27, and the total number of papers from all three sources combined was 67. Then, employing specific inclusion and exclusion criteria outlined in Chapter 2, we assessed the relevance and quality of these identified papers. Through this screening process, we selected 18 papers for further examination in our literature review.

These papers were then categorized into four distinct categories to gain a comprehensive understanding of the existing landscape. The first category consisted of papers that established connections between chatbots and business processes without relying on standardized notations like BPMN. The second set of papers focused on utilizing chatbots to enhance the modeling of business processes. In the third category, we encountered papers that integrated chatbots with business processes using standardized notations such as Subject-oriented Business Process Management (S-BPM) and BPMN. Lastly, we identified papers that conducted surveys and provided evaluation data about the usage and impact of AI and chatbots in academic and industrial contexts. Through this literature review, we identified a notable gap – the lack of approaches to evaluate if generative-based chatbots can meet the requirements of BPMN constructs and to compare different generative-based chatbot models in terms of meeting these requirements. This gap highlighted the importance of our research direction and the need to address this missing aspect.

1.5.2 Phase II: Retrieval-based Process-Aware Chatbots

In Phase II of our research, before diving into the exploration of generative-based chatbots, we embark on an exploration of the use of retrieval-based chatbots for business process execution support. This phase drew upon insights from our literature review and aimed to understand how retrieval-based chatbots could help us get to the final goal of using and evaluating the more complex generative-based process-aware chatbots for process execution support, the main gap that was found in our literature review. Phase II consists of two studies, S2 and S3, that embody the development of chatbots with the Rasa framework for two different scenarios: a Trip Planning process, that contains rigorous dependencies between tasks, and a Wedding Planning process, that has looser dependencies.

These retrieval-based chatbots are integrated with Camunda Engine, which makes them the gold standard for process-aware chatbots since they will stay synchronized with the evolving process execution status. This was a crucial initial step to provide a benchmark for the behaviour expected from future generative-based chatbots, which will not have the ability to get real-time process status updates. These retrieval-based chatbots lay the foundation for understanding how generative-based process-aware chatbots should operate, leading to the formulation of potential strategies for their development. By analyzing the solid structure of retrieval-based process-aware chatbots, this phase guides us towards the development of approaches to make generative-based chatbots process-aware, which addresses our first research question, RQ1: “What are some of the possible approaches to provide a generative-based chatbot with the process knowledge that is contained in a BPMN process model?”

1.5.3 Phase III: Generative-based Process-Aware Chatbots

In Phase III of our study, we delve deeper into the world of process-aware chatbots, focusing on the use of generative-based models to help in business process execution. This phase, which we divide into two sub-subsections, consists of two different approaches to making generative-based chatbots operate in the realm of process support.

III-a: Initial Approach

In III-a, we present our initial attempt at translating process knowledge into generative-based chatbots. This initial approach consisted of explaining the intricacies of specific BPMN process models using natural language. To achieve this, we built an algorithm that

was able to translate each element from a BPMN process model into English sentences and send the final output text to a generative-based chatbot – in this case, GPT. However, this initial experiment revealed challenges, as several BPMN constructs were not able to be adequately represented in natural language sentences and were not correctly interpreted by GPT, despite multiple attempts at modifying the algorithm to explain the same process in different ways. These challenges led us to the desire to explore a different approach, and also establish concrete evaluation criteria for assessing how well future generative-based chatbots can adhere to BPMN constructs. This investigation directly addresses our second research question, RQ2: “How can we evaluate if a generative-based chatbot is meeting the requirements of BPMN constructs for process execution support?”

III-b: Improved Approach

We devise an improved approach to share process context with generative-based chatbots, based on the sending of entire BPMN Extensible Markup Language (XML) documents to a chatbot, and include four distinct studies (S5, S6, S7, and S8), each representing a computational experiment with different use cases and different chatbot models. The first two experiments, S5 and S6, utilize GPT to tackle the Trip Planning and Wedding Planning processes, respectively, the same scenarios that had previously been implemented with Rasa and Camunda Engine in Phase II. This is achieved by taking the BPMN XML documents for each of these two processes after they were modelled on Camunda Modeler, and sending these documents to GPT. The subsequent experiments, S7 and S8, employ Google’s PaLM for the same processes. Each experiment is rigorously evaluated using the criteria that were established for RQ2 and presented in Section 5.2.

These experiments provide a comprehensive comparative analysis of GPT and PaLM. By examining their ability to adhere to BPMN constructs across multiple use cases, we aim to address our third research question, RQ3: “How does GPT-3.5 compare with other generative-based chatbot models, such as PaLM 2, in terms of meeting the requirements of BPMN constructs?” This phase is instrumental in discerning the strengths and weaknesses of different generative-based models in meeting the requirements of BPMN constructs, ultimately contributing valuable insights to the field of generative-based chatbot development.

1.6 Thesis Structure

As seen in the previous section, our research design was carefully crafted to explore the integration of chatbots with business processes. Through a mixed-methods approach, going

through a literature review and empirical analysis, we have taken a three-phase journey that deepens our understanding of chatbots' role in the business landscape. The literature review highlighted various dimensions of this integration and revealed an intriguing gap concerning generative-based chatbots within this context. Subsequent phases analyzed retrieval-based chatbots and their enhancement through process integration and, finally, we inspected the use of generative-based models for our process-aware chatbot solution. These insights contribute to a better comprehension of the capabilities and limitations of chatbots within the business process domain.

The structure of this thesis aims to reflect the multiple phases of our research design, telling a coherent story about the research methodology and implementations that were executed throughout the course of this study. The following chapters are laid out as follows. Chapter 2 contains the literature review, which encompasses a total of 18 papers across 4 different categories and sheds some light on the challenges of building process-aware chatbots and the gaps that exist in the literature related to this field. Building on those insights, Chapter 3 demonstrates an approach to building retrieval-based process-aware chatbots that are connected to a process engine, in an attempt to understand better how process-aware chatbots should behave and guide us towards the development of generative-based process-aware chatbots. Then, Chapters 4 and 5 cover two approaches to building these generative-based process-aware chatbots following the standards that were seen in the previous chapter. Chapter 4 illustrates an initial approach to using generative-based chatbots for process execution support, while Chapter 5 builds upon that approach and uses certain evaluation criteria to measure how well generative-based chatbots stick to BPMN constructs. In Chapter 5, two use cases (Trip Planning and Wedding Planning) are evaluated on both GPT and PaLM, in an attempt to compare both of them and see how they stack up against each other when it comes to aiding in process execution. Finally, Chapter 6 presents the conclusion of this thesis and introduces the future work that is expected to be undertaken after the current work is finished.

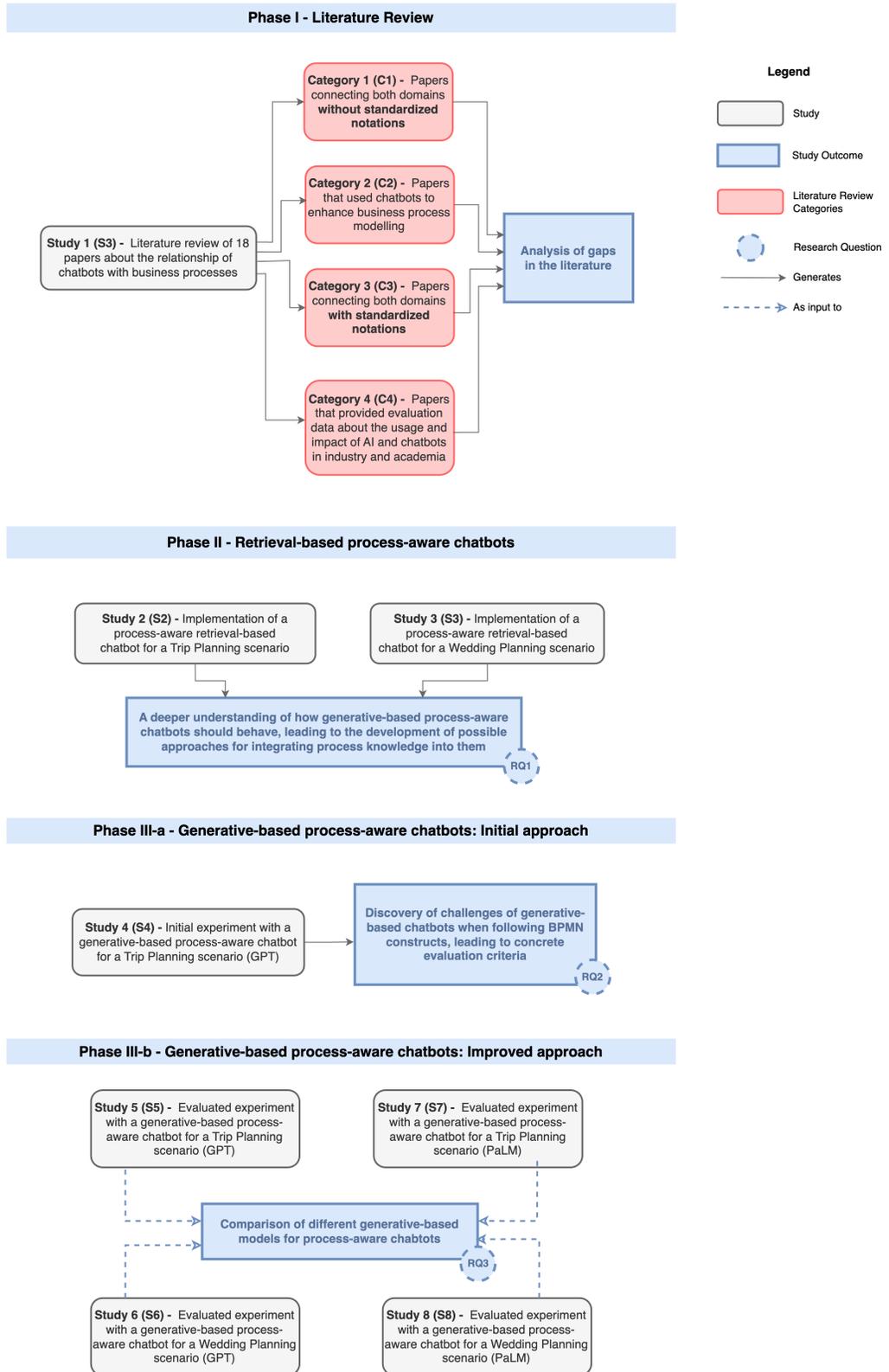


Figure 1.1: Visual description of the three-phase research design of the current study.

Chapter 2

Literature Review

In this chapter, we examine the existing body of literature that is related to the utilization of chatbots for navigating business processes. As organizations increasingly seek efficient ways to streamline their operations and enhance customer experiences, chatbots have emerged as a promising technology that can optimize how businesses execute their own processes. By utilizing natural language processing and complex algorithms, chatbots have the potential to automate routine tasks, provide instant support, and improve the overall efficiency of various business processes.

2.1 Research Method

To explore the landscape of research related to chatbots and their role in business process navigation, a comprehensive search query was formulated. The search query employed a combination of keywords and logical operators. The query used the terms (BPMN OR “business processes”) AND (ChatGPT OR “generative AI” OR LLM OR chatbots). The goal was to retrieve relevant academic papers, conference proceedings, industry reports, and case studies that focus on the application of chatbots in the context of business process optimization.

2.1.1 Utilized sources and results

To ensure a comprehensive and rigorous literature review, multiple reputable databases, including IEEE Xplore, ACM Digital Library, and Scopus, were utilized. The search

process encompassed studies published from 2021 to the present (2023), focusing on peer-reviewed articles written in English, to retrieve the most state-of-the-art papers.

The initial search query yielded a substantial number of potential research articles and industry reports. On July 26, 2023, when the initial search was performed, 10 papers and 1 book were found through IEEE Xplore, 35 research articles through ACM Digital Library, and 33 papers among conference papers and articles were retrieved from Scopus. It is important to note that 6 out of the 33 papers from Scopus had already been found on IEEE Xplore, so were removed. Thus, our final count of unique papers coming from Scopus was 27 in total, and the total number of papers from all three sources combined was 67. After applying the screening criteria detailed in the next section, 18 papers were selected for further examination and inclusion in this review.

To ensure the relevance and quality of the identified papers, a two-stage screening process was conducted. In the first stage, the titles and abstracts of the retrieved papers were reviewed to eliminate irrelevant studies. Papers that focused on unrelated domains or did not specifically address the use of chatbots for business process navigation were excluded. In the second stage, the full texts of the remaining papers were carefully examined to determine their suitability for inclusion based on their methodology, findings, and alignment with the research objectives.

The studies were selected according to with several criteria relevant to the research questions that were devised for our literature review and described in Section 2.1.2. The selection criteria (inclusion and exclusion) were constructed based on the suggested guidelines by Kuhrmann et al. [26]. The following inclusion criteria (IC) were established:

- IC1: The study introduces a foundation of a type of integration between chatbots and business processes;
- IC2: The study presents one or more use cases of chatbots integrated with business processes;
- IC3: The study presents data about the usage of chatbots in academia or in the industry;

Studies that met at least one of the following exclusion criteria (EC) were excluded:

- EC1: The study is not in English;
- EC2: The study is not a research article or conference paper;

- EC3: The study does not reference chatbots in depth;
- EC4: The study does not reference business processes in depth;
- EC5: The study is about automating business processes in graphical user interfaces (GUI);
- EC6: The full text of the study is not available;

The main reason for excluding studies that automate processes in GUIs is because, although having a similar goal to our research, which is to facilitate the execution of business processes, our focus is on the development of written instructions for the users who are executing a business process. The layer of automating actions in interfaces, such as clicking on buttons and filling out forms automatically, is beyond the scope of this research.

The list of all search results, with details on which ones were included in this research and which ones were excluded, and for what reasons, can be seen in Appendix B.

2.1.2 Research Questions

This literature review aims to address the following research questions, which guide the exploration of the existing work in the field of chatbots and business processes:

RQ1: Which chatbot frameworks are utilized in the papers?

The choice of chatbot frameworks significantly influences the capabilities, efficiency, and adaptability of conversational agents. By exploring the chatbot frameworks utilized in the papers, we aim to identify the major trends and preferences within the academia and industry. This analysis will provide insights into the technical foundations upon which chatbots are built and how they evolve over time.

RQ2: What design techniques were utilized for the development of the chatbots in the found papers?

This research question seeks to investigate the types of design techniques that have been used for the development the chatbots, such as retrieval-based or generative-based. By identifying the types of chatbots utilized and the existing gaps, we aim to contribute to a better understanding of that kinds of chatbots have been historically more utilized.

Both of these research questions play a key role in shaping the following discussions within this chapter. The analysis of these questions will enable us to assess how chatbots can be connected to business processes and highlight both the established knowledge and the potential areas for further research.

2.2 Paper Summaries

In this section, we provide a summary of the selected papers, organized by the key categories and topics that were found. During the course of our literature research, we systematically categorized the papers into four distinct categories, each shedding light on various intersections between chatbots and business processes. The four categories are as follows:

1. Process-aware chatbots with no standardized notation
2. Using chatbots to model business processes
3. Process-aware chatbots with standardized notation
4. Evidence-based studies about chatbot usage

The first category includes papers that established connections between chatbots and business processes **without relying on standardized notations such as BPMN**. We also identified a second set of papers that utilized **chatbots for the specific task of modelling business processes**, enhancing their structure and organization. The third category encompassed papers that, in contrast to the first category, integrated **chatbots with business processes, but now employing standardized notations such as S-BPM and BPMN to ensure formalized representations**. Finally, for our fourth category, we encountered **papers that conducted surveys and shared valuable evaluation data**, providing insights into the usage and impact of Artificial Intelligence (AI) and chatbots in both academic and industrial contexts.

For categories 1 and 3, in particular – i.e. the papers that are about the connection between chatbots and business processes (both with and without the use of a standardized notation) –, we found two main subtopics. The first subtopic centered around the foundational aspects of linking chatbots and business processes, covering potential methodologies and approaches for integrating these two domains. These papers explored theoretical frameworks without necessarily presenting concrete implementations. On the other hand, the second subtopic within these categories revolved around practical use cases of chatbots that have either been connected to existing business processes or proposed for such integration. These papers introduced real-world scenarios, demonstrating how chatbots can enhance various aspects of business processes.

By presenting these summaries, we aim to provide an overview of the existing literature and highlight the contributions made by each study in the field of using chatbots for navigating business processes.

2.2.1 Process-aware chatbots with no standardized notation

In this subsection, we will report papers that have been found to lay some of the foundations for the connection of chatbots with business processes, but only those that do not include the use of any standardized notation (like EPC or BPMN). The papers included here discuss some of the possible approaches to connecting both domains, as well as the challenges related to these approaches. We will then go over some works that develop actual implementations of chatbots that are used to help with business process execution, but still with no standardized notation.

Kecht et al. [24] performed an interesting study on quantifying a chatbot’s ability to learn and adhere to organizations’ business processes. To do this, they trained three chatbots with a corpus of 500,000 tweets in total of customer support conversations from human support agents, and then quantified the ability of these chatbots to learn the process from the dataset, and also adhere to the discovered process. Furthermore, the authors provide a way to compare the chatbot execution to a “normative process model”, which would be the actual business rules that the process should follow, and as suggested by them, could be the enterprise’s process model modelled using a common notation such as BPMN. Despite the suggested comparison to a normative process model, they leave this step as optional, and offer an alternative that is to mine these business processes from the dataset conversations using process mining algorithms. Although this process mining strategy is useful, in their paper, it remains the only strategy that was used, due to the lack of public access to internal business process information from the companies that they used the conversation datasets from. Therefore, in the current work, we aim to create a different approach from theirs, training a chatbot exclusively with BPMN models, instead of utilizing process mining techniques.

Another issue with this paper by Kecht et al. is that, because they train on a dataset that is specific to customer support conversations, their solution tends to misunderstand user input and produce responses that do not make any sense. That is due to the fact that they use a dataset of only 500,000 tweets, which at first may seem like a large dataset, however, compared with large language models, for example, GPT-3.5 that is trained on 300 billion words [38], it can be considered lacking. An example of a conversation carried by a chatbot created by Kecht et al. showing its languages and comprehension mistakes can be seen in Fig. 2.1.

Because of these mistakes, oftentimes, users can be skeptical of using chatbots. However, they can still be useful for helping users navigate processes and understand errors during process runtime. von Wolff et al. [47] conducted a research study to survey users’ acceptance of a chatbot application for business processes. This would allow them to better

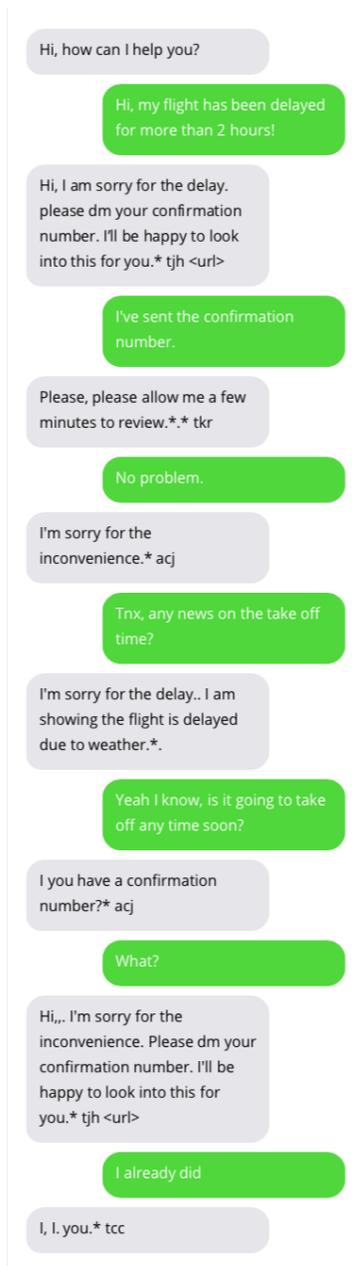


Figure 2.1: An example of a Delta Airlines support conversation conducted by Kecht et al.'s chatbot.

understand users' perspective in terms of design and acceptance of a process-based chatbot and the value that businesses and organizations could visualize for this kind of chatbot. After conducting their survey with 69 participants, their results showed that users can execute business processes in a similar or faster time when using process-based chatbots compared to existing solutions, and that this kind of chatbot also reduces the risk of users making critical errors or experiencing interruptions when using process-based chatbots to execute a business process. However, their study only analyzed a specific ad-hoc process, the case of a business travel request from a company's employee, that was only described in words and not in a systematic notation. Therefore, it is yet to be proved if their results would also apply to processes modelled in BPMN.

One issue that can arise when using chatbots to help in the execution of processes without a standard notation is the cold start, i.e., it can be challenging to make the chatbot follow a specific process if it does not have enough data about it. However, there have been approaches in the literature that aim to solve this problem. Zeltyn et al. [51] introduced a synthetic dataset that serves as a training resource for chatbots aligned with a specific business process, particularly a Human Resources Management Incentive Program (HR MIP). Their HR MIP dataset, provided in csv format, captures the dynamics of conversation turns between users and chatbots, and is based on a chatbot implementation that used IBM Watson Orchestrator. The dataset mirrors real-world scenarios encountered during the MIP process, encompassing recognized user intents, fallbacks, disambiguations, and other chatbot conversational elements. To determine the optimal actions for chatbot responses, the authors explore two distinctive approaches. The first, the non-process-aware (npa) approach, examines only the activity that was executed in the last conversation turn. In contrast, the process-aware (pa) approach augments these attributes with process path statistics, which reflect the frequency of activity occurrences within the current session until the present conversation turn, not only the last one executed. However, specific details regarding the exact process path statistics employed for process modeling remain unspecified within the study. Despite not using a standard notation, the authors' synthetic dataset, which incorporates business process modelling concepts in the set of user-chatbot conversations itself, represents a valuable tool for researchers and developers aiming to enhance chatbots' performance within business processes.

Another approach that attempts to allow chatbots to help in the execution of business processes without using a specific notation was presented by Rodríguez-García et al. [41]. The authors create their own framework for building process-aware chatbots, allowing process modellers to choose what NLU model they want to use, and model the process within their application's dashboard. Their presented application, IVRMaker, is an interactive telephone chatbot service designed for business process integration. Because IVRMaker

operates in a telephone context, it uses a voice-to-text conversion component that processes user voice prompts and transforms the chatbot's responses back into audio. After a user has modelled the process in the dashboard, the application then employs an information control component to manage conversational user flows. This information control component guides dialogues between customers and the chatbot, redirecting conversations to achieve customers' objectives. The evaluation of IVRMaker is conducted in two distinct domains: health and vehicle safety. In the health domain, the chatbot assists with customer service and patient appointment management, enhancing patient interaction efficiency. In the vehicle domain, the chatbot proactively schedules test appointments for certification renewal. Throughout the paper, it is possible to see that IVRMaker, thanks to its information control component, allows for a successful business process execution, even though it does not utilize a standard notation such as BPMN to model the processes.

In addition to the foundational papers that we have seen so far, there are other works that showcase some of the implementations that were developed in academia of chatbots being connected to business processes to help with the process execution.

Palacios et al. [37] proposed a mix of serverless functions, webhooks, and REST APIs to improve farm management systems, providing an integrated solution to manage tasks, users, and the field. It streamlines real-time communication between people across the farm operation, including crop advisors, irrigators and administrative staff. Their solution does not use a standard notation such as BPMN nor a proper workflow management system like Camunda, but instead, the authors created their own serverless functions to allow for the creation of tasks, assigning users to each task, and managing the completion of each farm-related task, such as nutrient, irrigation and pest management jobs. They created a chatbot using Twilio Serverless Functions that mirrors all communications happening on the field, and allows users to verify all history of issues and comments related a particular task. Because their focus is on creating the functions and APIs for managing the tasks, and the chatbot is only one component of it but not the main one, the authors do not provide any kind of evaluation or comparison between using the chabtot and the traditional REST API.

Ferreira et al. [19] built a chatbot using the DialogFlow framework that can help execute daily business operations and also generate insights to aid in the decision-making process. Their chatbot was able to log into the company's Intranet API, and also integrate with other applications used internally, such as event scheduling services, billing platforms, and project management platforms. It would also be able to generate data analysis and control access to them according to the user level in question. Their results showed that, for operations such as getting upcoming scheduled events, getting issued invoices, and getting a record of daily activity, the average time spent on each task was at least 50%

and at most 73.3% faster (60% on average) when using chatbots in comparison to using the previous methods, which was using the traditional interface. The authors also suggest that, because the chatbot is integrated with external services, it allows the users to get data from multiple services in real time, serving as a data aggregator, and thus, helping to make data-driven decisions.

Košecká et al. [25], on the other hand, did not build a fully working solution, but instead, proposed a chatbot solution for business processes at universities. Their paper showcased the evolution of the global chatbot market over the years, and the difference between different types of chatbots, such as the different methods of input processing and response generation, different possible goals, among other general characteristics of chatbot development. In their specific use case, they propose chatbots to be used for business processes related to student requests at universities. According to the authors, most student requests could be easily handled online by the student themselves, through the university’s website. However, the authors state that students usually prefer calling or emailing the office, which incurs human labour. Based on their data regarding the total processing time to answer student calls and respond to student emails and the hourly pay of the employees that dedicate their time to these tasks, the maximum Return On Investment (ROI) of adopting a chatbot solution for their problem would be 85.5%, and that implementing chatbots at universities would both eliminate staff costs and also help students faster.

Menshchikova et al. [32] conducted a study with the goal of integrating a chatbot built within ICQ New with the “1C: Enterprise” application. This application is used to facilitate accounting, as well as automate sales, document management, and employee-related processes. The authors focused on using 1C for business processes related to salary and employee management. In their research, the authors implemented a specific notification scenario within their system. Their chatbot was able to notify stakeholders about the completion of different document stages, such as the various steps of the salary payment process. Additionally, the chatbot could send reminders for essential tasks like report completion or impending deadlines, along with alerts for upcoming holidays. Despite not using a standard notation such as BPMN, this integration demonstrated how chatbot notifications and reminders can enhance the the execution of business processes, through the use of business automation software applications, in this case, 1C: Enterprise.

Another approach, different from the previous ones that connect a chatbot to a middleware server or an automation software, attempts to connect a chatbot directly to a database to execute process-related transactions. Gassen et al. [21] introduce “CAT (Conversational Agents for Transactional databases)”, a framework that employs the Rasa chatbot framework to directly link conversational agents with databases for efficient trans-

action execution. For any given task that requires a specific piece of information, such as booking a movie ticket – that requires a customer ID and screening ID –, it automatically identifies the required fields through the database, so that the developer does not need to specify them manually. In addition to reducing the effort needed to create a chatbot from scratch, this approach also supports various domains and databases. However, it does not cover scenarios in which multiple tasks need to be executed in a specific order. Nonetheless, CAT’s efficacy is demonstrated in a movie theatre database scenario, showcasing its ability to facilitate tasks like screening reservations and cancellations through natural language interactions, ultimately enhancing user experience and transaction execution efficiency.

Chatbots can also help in the execution of business processes that are machine-based, with no human involvement. Nazarkevych et al. [33] demonstrate a practical application of a Telegram-based chatbot that performs information retrieval from a website based on user-provided keywords. The chatbot works by first receiving a user request containing a particular website that they want to search and a keyword of interest. After retrieving the site’s page, it proceeds to analyze the content, comparing it to the user-provided keyword and generating relevant results. Once the analysis is complete, the bot sends the outcomes back to the user. Despite the successful use of a chatbot to complete a business process related to search requests, because the bulk of the paper is around the search algorithm that was used, the development of the chatbot itself appears to be based on the basic Telegram API, implementing simple commands without employing a dedicated framework. Furthermore, no evaluation of the solution’s effectiveness is provided.

2.2.2 Using chatbots to model business processes

In this subsection, we will see how chatbots can be and have been used to model business processes.

Almonte et al. [1] present Socio, a chatbot designed to facilitate the modelling of various types of diagrams. It enables active involvement from domain experts without technical backgrounds, utilizing natural language as the interface for model creation. For instance, in the case of a Unified Modeling Language (UML) class diagram creation, when given the message “School contains teachers and students,” Socio deduces the need for three classes — School, Teacher, and Student. In addition to model creation, Socio incorporates commands for model management, validation, and undoing/redoing modelling actions. Although it is not explicit which framework the authors used to build this chatbot, or whether they even used any framework at all, the authors mention that their chatbot can operate within social networks such as Telegram or Twitter. Furthermore, while the

presented use case concentrates on class diagram modelling, the authors state that Socio’s capabilities can also extend to other modelling technologies, and thus, one could assume that Socio could also be able to aid in the modelling of business processes, using a notation such as BPMN.

In addition to helping model business processes, chatbots can also be used to tweak existing processes, aiming to improve the execution time, money spent, and other metrics. Barón-Espitia et al. [6] introduce Coral, a chatbot developed using the Rasa framework for generating What-If scenarios within a business-related context. Coral takes as input a process simulation model, either built manually by using process simulation tools or automatically by using data-driven simulation tools. Then, operating through a text interface, Coral engages users to try out simulation intentions and build corresponding What-If scenarios. The chatbot then triggers the simulation based on the user-proposed changes, and provides users with anticipated process performance metrics. Coral additionally performs scenario comparisons and summarizes differences between various simulation scenarios. This application enables users to specify a range of simulation intentions, such as altering demand, resources, task execution times, task automation, and even comparing generated models. By offering tools for business users to define What-If scenarios and seeing their outcomes without the need for specific simulation tool expertise, Coral has the potential to accelerate decision-making and improve business process execution. However, it is important to note that this paper written by Barón-Espitia et al. lacks an evaluation section, which limits the demonstration of Coral’s effectiveness in practical applications.

When using chatbots to model processes, it is advantageous to use a standard notation. Although the focus of our current study is on the use of BPMN, there are other notations that have been used in the industry. For instance, Kurtz et al. [27] have used the S-BPM notation for creating chatbots connected to business processes.

In BPMN, the model entities are defined top-down, focusing on the sequence of activities and events in a process. On the other hand, S-BPM defines its model entities bottom-up and resembles how nature operates. In S-BPM, subjects represent real-world entities like machines, humans, and systems, that have specific abilities and can perform tasks and communicate with each other. They act like abstract state machines, moving from one state to another while performing tasks. When their states are complete, they may respond or pass information to other subjects. This contrasts with BPMN, which primarily focuses on the flow of activities and events in a process [20].

The focus of Kurtz et al.’s paper [27] is on the use of chatbots to aid in the creation of S-BPM process models. Because chatbots are capable of interpreting textual input from users, they can translate this input into graphical elements, even identifying negations,

tasks, subjects, business objects, and conditions. The authors develop a specialized chatbot named Albert to automate the extraction of business process knowledge from stakeholders and subsequently generate subject-oriented process models. Albert, developed as a minimum viable product (MVP) using the MS Bot Framework, demonstrates the feasibility of constructing process models exclusively through written input.

2.2.3 Process-aware chatbots with standardized notation

Similarly to what we presented in Section 2.2.1, we will begin this subsection by reporting papers that lay some foundations for the connection of chatbots with business processes, but now we will cover those that, in fact, utilize a standardized notation, in this case, BPMN. We will also go over some works that introduce implementations of chatbots for business process execution support and that use BPMN to help guide the chatbot implementation and/or process execution.

Our first foundational paper was written by our own research group. Lins et al. [29] introduce a novel approach to chatbots named Process-Aware Conversational Agents (PACAs). This approach leverages the BPMN standard to assist in real-time process execution. PACAs merge concepts from process and conversational agent domains, transforming chatbots into tools that guide and control conventional processes. PACAs are developed by integrating chatbots with a WFMS to control BPMN process execution using natural language. In their proposed solution, Rasa performs natural language understanding to comprehend user inputs, while the Rasa Core back-end communicates with the Camunda Engine's API to identify available tasks, present them in user-friendly language, and execute tasks within Camunda based on user instructions. However, the authors acknowledge limitations stemming from the rule-based nature of the utilized chatbot framework. The approach requires manual adaptation of all process model elements to the chatbot framework, which poses configuration challenges and demands considerable effort when the process model changes. While the inherent limitations are acknowledged, this approach represents a foundational step toward future approaches and demonstrates the potential of using BPMN-driven conversational agents to enhance process execution and guidance within various domains.

Another paper, written by Rooein et al. [42], investigates the use of chatbots to teach business models written in BPMN to different actors in a company. However, as the authors state, their chatbot is only used for teaching, and not during process runtime. They even mention that, as a future work, a chatbot that acts during process runtime could be useful, for example, in the case where an exceptional condition occurs, where the chatbot could

then be used to explain at which stage the problem is and possible countermeasures to be applied. Our solution aims to fulfill this gap, creating a chatbot that is based on a real BPMN model and executed during process runtime, to help users understand possible errors and fixes.

Process-aware chatbots can also be used at universities, to aid in the students' learning process. Baena-Perez et al. [4] constructed a framework to facilitate the creation of mobile learning apps featuring Chatbots within Virtual Reality (VR) environments. The authors focus on simplifying VR environment authoring by using BPMN diagrams to define app behavior and DialogFlow-developed chatbots for user interaction. The BPMN diagram is used to orchestrate the app's progression in a VR panoramic video, and during the experience, a chatbot component asks questions to the user and reacts based on their response. The authors conducted a student satisfaction analysis within a German language course, revealing that 18 out of 24 students perceived the app's use as easy or very easy, with none finding it difficult. Also, 23 out of 24 students believed the app improved their language learning, particularly in comprehension, pronunciation, and oral expression. This paper shows how the integration of chatbots and BPMN models can optimize software development by simplifying the creation of immersive VR learning environments. Furthermore, the framework's ability to enhance user satisfaction in educational contexts showcases its potential for innovative learning solutions.

Another example that was seen in the literature of the use of chatbots for business process execution support is in the healthcare field. Amantea et al. [2] present a framework that leverages both BPMN modelling and chatbot technology to enhance healthcare processes. Initially, the authors model the hospital service process using BPMN, with the goal of improving resource allocation and patient care. Then, after identifying the challenge of diet management for patients with Amyotrophic Lateral Sclerosis (ALS), the authors propose a chatbot to monitor patients' diets continuously, allowing them to register their meals through a chatbot. Their approach allows for continuous patient monitoring, and the collected data from the chatbot has the potential to be used to automatically detect deviations from the care process that was modelled in BPMN. In their evaluation, the chatbot was utilized by 22 patients over recent months, however, the authors do not clearly mention how the chatbot usage was connected to the process that had been modelled in BPMN. Instead, it seems that the chatbot was only used for a single step in the entire process, which was the diet management phase. Although a concrete connection between process model and chatbot seems to have been more theoretical than technically implemented in this study, the idea of an integration between BPMN modelling and chatbot technology shows how it can be applied to enhance healthcare processes and patient outcomes.

2.2.4 Evidence-based studies about chatbot usage

In this subsection, we aim at reviewing a few studies that showcase some data around how chatbots are in fact used in the real world, how they can improve companies' financial performance and how users have been accepting the use of chatbots.

Singh [46] analyzes a recent transition in the Indian economy towards the integration of AI. By analyzing data from the top 500 Indian companies, the study demonstrates that this transition towards AI adoption is related to an increase in company performance. Through content analysis of these annual reports, it is revealed that nearly 60% of Indian companies are proactively embracing and investing in intelligent technologies. This includes leveraging robotics and chatbots to remain competitive on the global stage. Furthermore, the research establishes a positive correlation between the use of AI in business and enhanced company performance, with statistically significant results (t-value 2.74, p-value 0.006 ; 0.01). The study also highlights the substantial utilization of AI and chatbots within multiple sectors of the economy, especially in medicine, health and banking, and mentions that the adoption of AI in the banking sector specifically has showed improved financial outcomes by reducing transaction costs.

In another study, Danylevych et al. [15] conducted a survey focused on the utilization of bots in the HR department of institutions of higher education and companies in the creative industry. Through their survey, they found that among AI technologies, chatbots are the second most commonly employed, adopted by 22% of the companies that utilize AI. Chatbots are recognized as versatile tools within HR management, assisting at various stages from talent acquisition to career planning. Robot recruiters demonstrate the ability to significantly expedite recruitment processes, reducing time and costs by an impressive margin. Out of all the respondents who said they use chatbots for HR, 40% mentioned they used this technology for enhancing work time management. The emergence of chatbots in recruitment processes also stood out as a noteworthy trend, being used by 37% of the employees who used chatbots for HR. Other uses for chatbots in HR included staff training, talent management, motivation, and leadership development. Interestingly, the respondents expressed reservations regarding AI's involvement in decision-making for employment, with 91% indicating a preference for human participation in this domain. This result serves as a basis for this current thesis, confirming our hypothesis that chatbots should be employed to help human users execute business processes, rather than aiming for full automation. In summary, Danylevych et al.'s research provides valuable insights into the current landscape of chatbots adoption in the HR department of various organizations. Their findings emphasize chatbots' utility in various business processes while also reinforcing the importance of human involvement in them.

2.3 Results and Critical Analysis

Following the paper summaries, we gather the results that were found for the first and second research questions, and critically analyze the strengths and weaknesses of the reviewed papers. We identify the gaps in the literature, and discuss the implications of the existing research for our own work and research questions. Through this analysis, we aim to gain insights into the current state of the field and identify opportunities for further research.

2.3.1 RQ1: Which chatbot frameworks are utilized in the papers?

In response to the first question, “Which chatbot frameworks are utilized in the papers?”, we specifically analyzed the papers that implement an actual chatbot. Therefore, papers from the “Evidence-based studies about chatbot usage” subsection are not included, and neither is [25], which only presents a conceptual chatbot for universities but does not in fact implement one. Therefore, we are left with 15 papers to analyze which frameworks they use.

Seven of these studies (almost 47%) use frameworks that are available in the industry, including their NLU components and the software piece that determines the chatbot’s actions. Three of them ([21], [29] and [6]) use Rasa¹, an open-source framework. Then, two of them ([19] and [4]) use Google’s DialogFlow², one ([51]) uses IBM’s Watson Orchestrate³ and the last one uses the Microsoft Bot Framework⁴.

Four of the selected studies did not use any specific framework apart from the basic messenger APIs. When a chatbot is developed solely utilizing messaging platforms’ APIs without the integration of comprehensive chatbot frameworks, like Rasa or DialogFlow, the resulting bot typically remains limited in its capabilities. In this case, the chatbot’s interactions with the user are restricted to responding to explicit commands initiated by users. This approach lacks the sophistication and flexibility offered by dedicated chatbot frameworks, which can leverage natural language processing, machine learning, and contextual understanding to engage in more dynamic and contextually relevant conversations. However, these four studies still decided to not utilize any specific framework, to simplify their implementation. Two of them ([33] and [1]) utilized Telegram’s API, one ([32]) used

¹<https://rasa.com/>

²<https://cloud.google.com/dialogflow>

³<https://www.ibm.com/products/watson-orchestrate>

⁴<https://dev.botframework.com/>

ICQ’s API, and the last one [37] used Twilio Serverless Functions to communicate with users via SMS, therefore, not relying on any specific messaging platform.

One of the studies ([2]) did not clarify what particular framework was used or if any framework was used at all. Due to the simplicity of the implemented application that was shown in the screenshots, which apparently only communicated with the user via buttons and a simple text input to answer a specific question, it was assumed that only a basic messenger API was used. However, this assumption was not able to be confirmed by reading the paper’s content.

Finally, three of the papers ([42], [41] and [24]) built their own custom framework, which means that they either create their own NLU component or allow the user to select a custom NLU component, and also develop the logic for deciding which actions the bot should take after specific user intents. This approach incurs a much higher complexity for the implementation, meaning that more time and effort must have been deployed for developing these solutions. However, the advantage is that the final result is much more tailored to the author’s particular use case, being more appropriate for complex scenarios where a traditional chatbot framework would probably lack in performance.

2.3.2 RQ2: What design techniques were utilized for the development of the chatbots in the found papers?

Moving on to the second question, “What design techniques were utilized for the development of the chatbots in the found papers?”, we decided to categorize each study into one of three categories that were laid out by Hussain et al. [23] in their survey on chatbot classification and design techniques for developing chatbots. In their study, Hussain et al. mention that a chatbot’s design can fall into one of three main categories:

1. Rule-Based Approaches
2. Retrieval-Based Approaches
3. Generative-Based Approaches

Rule-based chatbot models operate on predefined patterns or commands, generating responses based on explicit rules programmed into the system. When a user inputs a query or issues a command, the chatbot searches for that specific command in its database, or relevant keywords contained in the user’s request, to determine the appropriate response.

These patterns are predetermined by the developers and limit the chatbot’s ability to provide nuanced or contextually relevant answers.

Retrieval-based models, on the other hand, use classification algorithms to categorize user input into predefined intents or classes. These algorithms, trained on labeled training data, help the model understand the user’s intent and extract relevant entities, which are pieces of information that are crucial and specific to that request. The chatbot then retrieves a suitable response from a predefined repository of responses, considering both the context and the inferred intent. While these models offer more fluent responses and are able to better understand the users, they still rely on predefined responses.

In contrast, generative chatbot models take a different approach. These models have the capability to generate responses that may not have been part of the training data. Instead of relying solely on predefined patterns or response repositories, generative models generate responses based on the user input and previous conversation context and based on an LLM that was used to train the model, that can contain up to hundreds of billions or even trillions of parameters. They compute response probabilities across the entire vocabulary of the model and generate responses word by word. This enables them to provide more diverse and contextually relevant answers.

In our research, we found that five studies ([32], [33], [1], [37], and [2]) utilized the rule-based approach, while all the 13 other papers utilized the retrieval-based approach. None of them developed a generative-based chatbot.

The five studies that followed a rule-based approach were mainly the ones that used solely the messaging platforms’ APIs, since they only support basic commands, usually starting with a backslash (\) such as (\add, \send, etc). Since the only way to make these chatbots “retrieval-based” would be if they were able to analyze user intent by using classification algorithms, and these APIs do not support this, these studies are considered rule-based. Additionally, the paper that used the unclear framework ([2]) is also classified here as rule-based due to its nature of communication with its users, which is solely based on buttons and simple text input as an answer to a question. Therefore, this one also does not analyze user intents.

All the other papers, including the ones that used their own custom frameworks and the ones that used industry-standard chatbot development frameworks, were considered to be retrieval-based, since they all covered the aspect of understanding the user’s input using Machine Learning or specific classification techniques.

Overall, the literature review points to an absence of generative-based chatbot models among the 18 analyzed papers. More specifically, we found a significant research gap referring to the lack of presence of generative-based chatbots in the literature, and consequently,

the lack of approaches to evaluate if generative-based chatbots can meet the requirements of BPMN constructs and to compare different generative-based chatbot models in terms of meeting these requirements.

In our current study, we recognize the importance of addressing this gap by focusing on the usage and evaluation of generative-based chatbot models for business process execution support. By venturing into this unexplored territory, we aim to contribute to the broader understanding of how such models can enhance the naturalness, adaptability, and overall conversational quality of chatbot interactions, especially when aiding in business process execution. This approach aligns with the evolving trends in AI and Natural Language Processing (NLP), which emphasize the potential of generative models to offer more creative and contextually-relevant responses.

In conclusion, this chapter serves as a review of the existing research and literature related to the utilization of chatbots for navigating business processes. By examining a range of studies and critically analyzing their findings, we aim to establish a solid foundation for our own research and contribute to the advancement of this emerging field.

Chapter 3

Retrieval-based process-aware chatbots

During our literature review, we realized that there is a research gap related to the lack of approaches to evaluate and compare generative-based chatbots in terms of meeting the requirements of BPMN constructs. However, before we dive into the exploration of generative-based chatbots, we describe an approach to creating retrieval-based chatbots that are connected to a process engine, to help users execute business processes while following all the requirements set by the BPMN standard. The integration between these retrieval-based chatbots with Camunda, a process engine, makes them the gold standard for process-aware chatbots, since these chatbots will be able to stay continuously synchronized with the evolving process execution status, and therefore, meet all the requirements of the BPMN constructs.

To achieve that, we identified the need to use a chatbot framework that is both open-source and flexible, in the sense that it should be able to connect to external APIs. That is because Camunda Engine provides an API that allows users to track the execution of a process instance, and that resource is what we aimed to use for process adherence. For this work, we have decided to use Camunda Engine because it is an open-source WFMS that is well-documented and widely used both in academia and industry. With this connection in place, we can make the chatbot communicate with Camunda Engine to understand what tasks are available at each moment.

Because of this connection with Camunda Engine, the chatbot will be able to follow the process execution as determined by an actual process engine, and will always know the most up-to-date state of the process instance, therefore, being the best reference point for

how the process should be executed. For the same reason, we have decided not to carry out any evaluation of how well this retrieval-based chatbot follows the BPMN constructs, since we take as a fact that it will be following all of the BPMN constructs used in the original process model, in accordance with Camunda Engine.

In the next section, we will cover some chatbot concepts that are specific to Rasa, the framework that we chose for building this chatbot, to shed some light on how this framework works, and how it can be connected to Camunda Engine. We will then present two use cases of process-aware chatbots that are related to two different scenarios, the first one being a Trip Planning process and the second one being a Wedding Planning process.

3.1 Implementation Strategy

The Rasa framework is an open-source tool for building conversational AI applications, and it is composed of an NLU component, which is responsible for understanding users' messages, and Rasa Core, which determines the next actions the bot will take. On the NLU side, the most significant concepts are *intents* and *entities* since they influence the bot's understanding of user input. *Intents* represent the underlying purpose or goal behind a user's message, encapsulating the user's intention within the conversation. On the other hand, *entities* are specific pieces of information extracted from the user's input, such as dates, names, or numbers, which are essential for carrying out the desired action or providing an accurate response. Together, *intents* and *entities* enable chatbots to understand and respond to user messages with a high degree of accuracy, facilitating interactions between humans and AI-driven systems.

On the Core side, *rules* determine chatbot behaviors and enable personalized responses. *Rules* are predefined instructions that dictate how the chatbot should respond when specific conditions are met, typically after particular intents are identified in user input. For instance, when a user's input is identified as an "Order Pizza" intent, a rule can be set to trigger a series of actions, such as asking for pizza preferences, confirming the order, and providing an estimated delivery time. Similarly, after recognizing a "Check Weather" intent, there should be a rule that will trigger an action for getting weather data for the requested location and convey it to the user.

These *actions* that are triggered by rules can encompass a variety of functionalities. They can be categorized into two main types: simple *utter actions* and more complex *custom actions*. On the one hand, *utter actions* involve the chatbot responding with predefined text messages. These actions are typically used to provide immediate feedback

or an answer to the user. For instance, if a user greets the chatbot, a simple utter action might involve the chatbot responding with a message like “Hi, how are you?” These actions are straightforward to implement and are useful for delivering quick responses to common user queries but are always fixed. On the other hand, *custom actions*, usually implemented in Python, allow developers to execute custom logic or functions in response to user inputs. These actions can perform a wide range of tasks, such as accessing external APIs, retrieving data from databases, performing calculations, or interacting with other systems.

In our case, these custom actions hold a crucial role when it comes to integrating Rasa chatbots with process engines like Camunda Engine. They serve as the bridge that enables seamless communication between the chatbot and the Camunda Engine API. Through custom actions, we will connect our chatbot to the Camunda Engine, retrieve real-time information about process instances, monitor task availability, and track the state of ongoing processes. This integration empowers the chatbot to provide users with accurate and up-to-date information regarding their ongoing process instances, and allow them to interact with and manage them effectively.

These concepts that have been explained throughout this section are spread out across multiple files in a Rasa chatbot, and it is useful to understand what these files are and what they do. The following list describes each of these relevant files:

- `nlu.md` – contains the sentence examples that represent each intent, so that the NLU component can adequately identify the intent corresponding to each user input;
- `rules.md` – contains fixed instructions regarding which actions should be triggered by each intent;
- `actions.py` – contains the code that should be run by each action. As previously mentioned, actions are usually written in Python due to Rasa’s default Python Software Development Kit (SDK), but they can be written in any other server-side language;
- `domain.yml` – contains a summary of all the information that is relevant to the bot, including intents, entities, actions, etc., so that the chatbot can know what to expect from the other files and identify if any information is missing;
- `config.yml` – determines basic configuration settings for the chatbot.

3.2 Use Case 1: Trip Planning

After understanding some of the chatbot concepts that are used by Rasa, we will now go through a concrete use case of how a retrieval-based process-aware chatbot could be built to help in the execution of a business process modelled in BPMN.

3.2.1 Process Model

The chatbot we will create in this section will be built to follow a Trip Planning process. We utilize a BPMN model for this process that was published in a previous paper [29], and that has strict dependencies between tasks. Fig. 3.1 presents the diagram for this process. It includes four user tasks (represented as rectangles), exclusive gateways (represented as diamonds with an X in the centre), and sequence flows (represented as arrows). The start and end events are represented as circles at the beginning and end of the diagram, respectively.

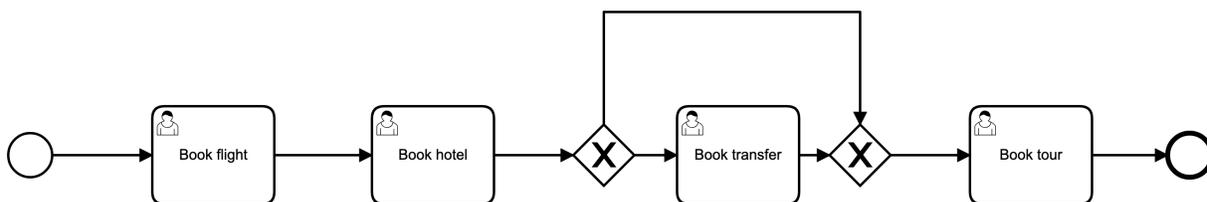


Figure 3.1: BPMN diagram for the “Trip Planning” process.

In this process model, after the process instance begins, the only available task is “Book flight” and it is required. As soon as the user marks it as complete, the chatbot will ask for a process variable “flightDate”, which is the date of the booked flight. After the “Book flight” task is completed, the user can execute the “Book hotel” task, also required. When completed, another process variable will be requested: “userWantsToBookATransfer”, which will store the user’s decision to book a transfer or not. Then, an exclusive gateway will be responsible for analyzing the value of this variable and determining the next available task. If the user does want to book a transfer, then the next available task will be “Book transfer”, and that will be a required task. If not, then the last available task will be “Book tour”, after which, the process will end. If the user has decided to book a transfer, they will still have to execute the “Book tour” task before ending the process, as this task is also required. In summary, there are two possible flows in this process, depending on whether the user decides to book a transfer:

1. Book flight
2. Book hotel
3. Book transfer
4. Book tour

or

1. Book flight
2. Book hotel
3. Book tour

3.2.2 Chatbot Implementation

In our proposed solution, Rasa will utilize its NLU component 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.

In our Trip Planning scenario, after Camunda and Rasa are running, a user can begin a conversation with the chatbot 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 Graphical User Interface (GUI) provided by Camunda, called Camunda Cockpit. The “Trip Planning” process instance in Camunda Cockpit is shown in Fig. 3.2.

After receiving the list of available tasks, which includes 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 on Camunda Engine, as shown on the list at the bottom of Fig. 3.2. The next task, “Book hotel”, will also request a piece of information to fill the “userWantsToBookATransfer” process variable in Camunda, and

determine the next available task, whether it will be the “Book transfer” or the “Book tour” task. The user then proceeds to complete the next tasks, and then, after all tasks are executed, the chatbot informs the user that the process is finished. An excerpt of a conversation with the Trip Planning chatbot, up to the completion of the “Book flight” task, can be seen in Fig. 3.3.

The screenshot displays the Camunda Cockpit interface for a process instance. On the left, the 'Information' panel shows details for the process instance, including Instance ID, Business Key, Definition Version, Definition ID, Definition Key, Definition Name, Tenant ID, Deployment ID, and Super Process Instance ID. The main area shows a BPMN flowchart with tasks: Book flight, Book hotel, Book transfer, and Book tour. The 'Book flight' task is completed, and the process is currently at the 'Book hotel' task. Below the flowchart, a table lists variables:

Name	Type	Value	Scope	Actions
flightDate	String	Sep 20, 2021	Process (simple_t...)	✓ 🗑️
userWantsToBookATransfer	Boolean	false	Process (simple_t...)	✓ 🗑️

Figure 3.2: Process instance shown in Camunda Cockpit after the “Book flight” task.

3.3 Use Case 2: Wedding Planning

3.3.1 Process Model

After experimenting with a Trip Planning process that had strict dependencies between tasks, for this second use case, we aimed to explore a more flexible scenario. Our goal was to assess whether the connection between the business process and chatbot concepts could still hold, even in the absence of rigid task interdependencies. To do so, we chose to examine a “Wedding Planning” process as an example. Within this context, we identified five primary tasks:

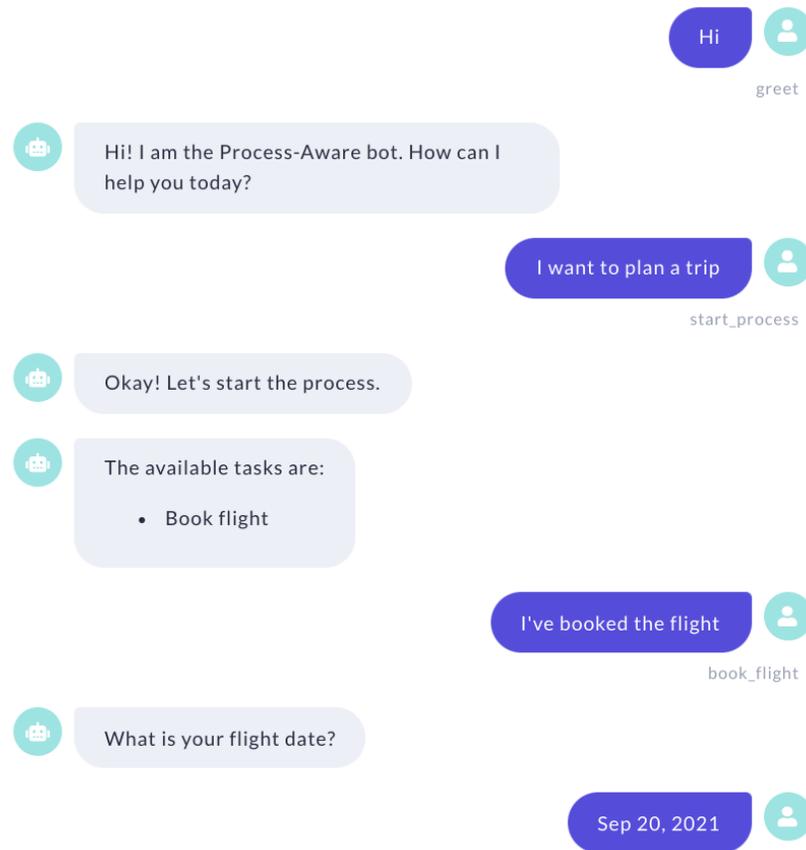


Figure 3.3: Conversation with a process-aware chatbot in a “Trip Planning” scenario.

1. Selecting a wedding date
2. Reserving a venue
3. Booking a band
4. Securing a caterer
5. Booking a photographer

When translating real-life activities into a formalized process, we often need to make certain assumptions to represent them effectively using a widely recognized notation. In this case, we assumed that once the couple had settled on a specific wedding date, the other

four tasks could be executed in any order. For instance, there was no strict requirement to book a venue before securing a band, caterer, or photographer. These four activities could reasonably be considered independent of one another.

Having designed a model that incorporated this flexibility, the resulting process representation is depicted in Figure 3.4. This diagram illustrates that the first task to be initiated is selecting the wedding date. Subsequently, the four primary tasks can be carried out in any order, as indicated by the fork and join structures with parallel gateways (the diamonds with a “+” sign in the centre). The presence of these fork and join elements also signifies that all four tasks must be completed before the process can advance.

Additionally, we incorporated a Service Task named “Finish Booking” immediately after the gateway following the four booking tasks. In this specific scenario, this task does not execute any code. Instead, it serves as a visual representation of the potential to create a program class to handle the actual booking process, which would automatically trigger once all four booking tasks are completed. However, in our use case, it merely evaluates an expression that invariably returns “true.” Therefore, this Service Task is traversed without significant impact.

Finally, it is worth noting that the only appropriate way to conclude the process execution is by successfully completing all bookings. If the user were to exit the software mid-execution, the process instance would remain active within the Camunda Engine until manually terminated, either through the Camunda Cockpit or the API.

3.3.2 Chatbot Implementation

In Section 3.1, we explained how Rasa utilizes its NLU component to comprehend user inputs and Rasa Core, the framework’s back end, determines the chatbot’s next actions. For the Wedding Planning process, just like for the Trip Planning process, the Rasa Core component interacts with Camunda Engine and lists available tasks in a user-friendly manner. When a user confirms task completion, the Core makes an API request to Camunda to mark it as done.

For the Wedding Planning scenario, the user will start a conversation by asking the chatbot to plan a wedding. This initiation triggers the creation of a new process instance in Camunda Engine. The execution of this process instance can be tracked through Camunda Cockpit just like in the Trip Planning process.

After the start of the process instance, the user will be presented with the first available task, “Choose date”. Upon completion of this task, the chatbot will request the user’s

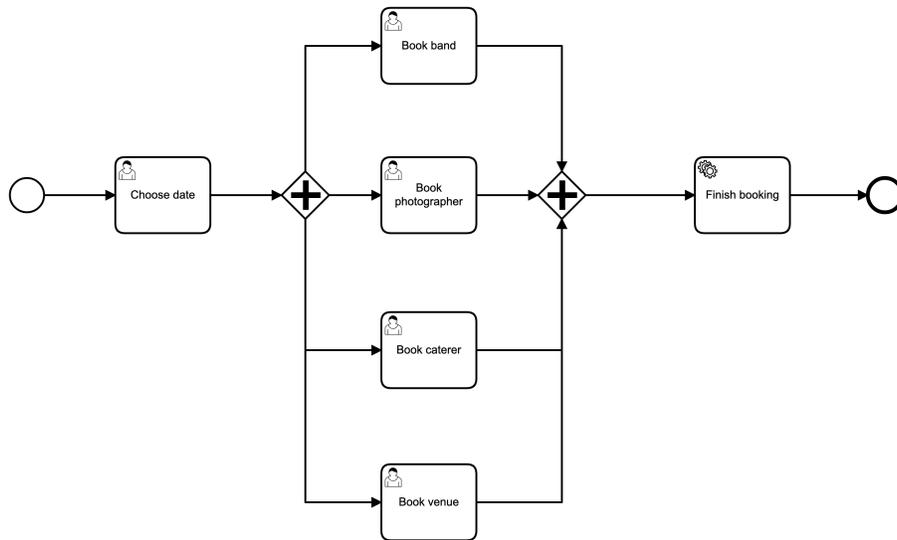


Figure 3.4: BPMN diagram for the “Wedding Planning” process.

selected date, populating the “chosenDate” field in Camunda Engine. The subsequent four tasks: “Book venue”, “Book band”, “Book caterer” and “Book photographer”, will then be all presented to the user and can be completed in any order. The chatbot will guide the user through each, and when all four are marked as completed, the system will automatically proceed to the “Finish Booking” task. A conversation example with the “Wedding Planning” bot is shown in Fig. 3.5.

3.4 Lessons Learned

The implementation of retrieval-based chatbots integrated with the Camunda Engine for the purpose of tracking and assisting in process execution has demonstrated their success in adhering to BPMN standards. Two main takeaways highlight the efficacy of these chatbots:

1. **Accurate Task Availability:** The chatbots consistently provided users with an accurate view of task availability. They successfully identified available tasks and avoided the presentation of tasks that were unavailable. The chatbots even prevented users from execution tasks that were not yet available. This level of accuracy ensures that users are always directed towards productive actions.

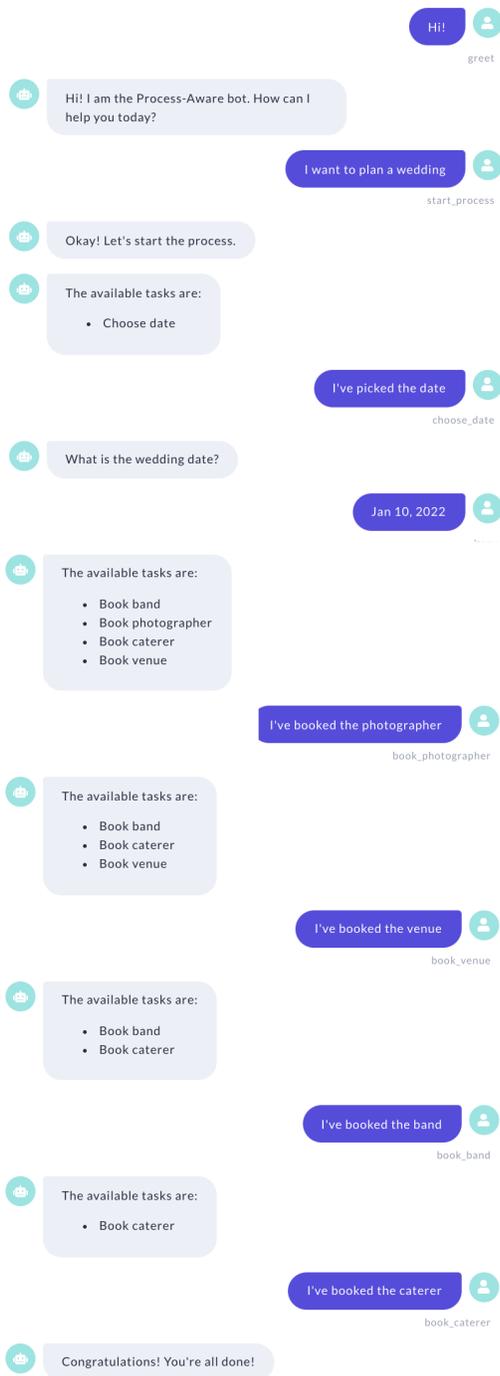


Figure 3.5: A complete conversation with the “Wedding Planning” bot.

2. **Prompt Data Gathering:** In cases where task completion required additional information, such as the flight date, in the case of the “Book flight” task, the chatbots reliably requested the necessary data after the task’s execution. This approach not only ensured task completion in alignment with BPMN requirements, but also minimized the need for extensive user interactions beforehand, streamlining the process execution.

These achievements reaffirm the effectiveness of retrieval-based chatbots in adhering to BPMN standards. However, it is important to acknowledge two main challenges associated with retrieval-based chatbots:

1. **Development Complexity:** The implementation of retrieval-based chatbots can be labour-intensive and time-consuming due to their context-specific nature. Each chatbot requires careful customization to match the specific processes that it is designed to assist, and automating their development solely based on BPMN process models remains a challenging endeavor.
2. **Limited Understanding and Training Data:** Retrieval-based chatbots have inherent limitations in their comprehension of user inputs and are restricted to the knowledge present in their training data. Consequently, they may face challenges in understanding user requests that diverge from their predefined training data.

In summary, the integration of retrieval-based chatbots with Camunda Engine has proven to be a promising approach for process execution support. While their capacity to adhere to BPMN standards is clear, addressing the challenges of development complexity and knowledge limitations is essential to improve the applicability of process-aware chatbots. Therefore, we decided to explore the use of generative-based chatbots for process execution support, and different approaches to transferring specific process knowledge to this kind of chatbot. By taking advantage of the capabilities of generative models, we aim to facilitate chatbot development and enhance their adaptability to diverse user requests.

Chapter 4

Generative-based process-aware chatbots: Initial approach

After seeing how process-aware chatbots can be created using a retrieval-based framework connected to a process engine, we then proceed to the exploration of how we could make generative-based chatbots process-aware. Specifically, we aim to evaluate and compare generative-based chatbots in terms of their ability to meet the requirements of BPMN constructs even without the connection to a process engine. That is because, in this work, we assume that creating a direct connection between a generative-based chatbot and a process engine would not be feasible, due to the fact that the chatbot models that were chosen for this study, namely GPT and PaLM, are closed source. All interactions of these chatbots with external applications are only made possible by plugins, and to this date, none of the available plugins support connecting with a process engine. However, we still aimed to take advantage of all the processing and conversational power that these models have for our specific scenario of business process execution support. In our initial quest to integrate BPMN models into generative-based chatbots, our first approach was based on the conversion of BPMN constructs into natural language.

4.1 Implementation strategy

In the current chapter, our approach to share process knowledge with generative-based chatbots involved the transformation of BPMN processes into natural language sentences. This method consisted of parsing a BPMN diagram, segmenting each BPMN element into

logically categorized components, and subsequently generating natural language sentences based on these components. By converting the abstract BPMN representation into human-readable text, we aimed to make the process more accessible to users interacting with the chatbot.

Our strategy involved a systematic process of BPMN model interpretation. To do that, we utilized a BPMN parser called “bpmn-moddle”¹, developed by the “bpmn.io” organization. This parser is able to extract all the elements from a BPMN XML file, and categorize each one according to its content. Then, we created a recursive function called “analyzeNextElement” to navigate the BPMN model’s elements and construct a coherent natural language explanation of the process flow.

Our process began with the identification of the start event, which initiated the traversal. The “analyzeNextElement” function then progressed through the model by following the sequence flows, capturing the order of tasks and their respective names. To convey this information effectively, a task counter was employed, ensuring that each task was referenced with its appropriate position within the sequence.

In instances where an exclusive gateway was encountered, i.e. a decision point in the process, the function created a branching structure in the output message, with the gateway’s name being incorporated into the process explanation. To account for multiple possible paths stemming from the gateway, we utilized condition expressions associated with outgoing sequence flows. In this case, these expressions must be natural language sentences that represent what would cause the execution of each flow, and they are used to identify different process flows within the message.

For the flows that come out of an exclusive gateway, the task counter remains consistent for all the different flows since these flows are mutually exclusive. That means that, when explaining different flows in the message, the first task of each flow right after the gateway will have the same number, and increase incrementally, in parallel. Finally, upon reaching the end event, the message would simply state, “And then the process ends,” providing a clear indication that there were no further process steps.

This methodology facilitated the conversion of BPMN models into English sentences, a format that can be readily processed by generative-based chatbots. By systematically interpreting BPMN elements and constructing natural language explanations, we aimed to create a foundation for chatbots to provide users with insights into process flows, regardless of the specific BPMN model in use. This approach was designed to be adaptable to different BPMN scenarios, allowing for a seamless integration of BPMN-based conversations with

¹<https://github.com/bpmn-io/bpmn-moddle>

chatbots. In the next section, we will go over a use case related to the Trip Planning process, the same one that was used for the retrieval-based chatbot, and see the step-by-step creation of the message that is generated for that process.

4.2 Use Case: Trip Planning

As mentioned in the previous section, we used a BPMN parser called “bpmn-moddle”², developed by the “bpmn.io” organization. This parser is able to extract all the elements from a BPMN XML file, and categorize each one according to its content.

Starting with the same Trip Planning BPMN model as last chapter, portrayed in Fig. 3.1, we then proceed to take its corresponding XML file that was generated on Camunda Modeler, and input it into the “bpmn-moddle” application. Table B.1 shows some examples of elements that are extracted from a Trip Planning BPMN XML file.

Table 4.1: BPMN elements extracted with “bpmn-moddle”

Type	ID	Source Ref ID	Target Ref ID
bpmn:StartEvent	StartEvent_1	N/A	N/A
bpmn:UserTask	book_flight	N/A	N/A
bpmn:UserTask	book_tour	N/A	N/A
bpmn:SequenceFlow	SequenceFlow_0an5hs1	book_tour	EndEvent_1pckwqd
bpmn:EndEvent	EndEvent_1pckwqd1	N/A	N/A

Despite not portraying all elements contained in our BPMN model, Table B.1 shows how different elements are categorized by “bpmn-moddle”, and connected by sequence flow elements. The idea was then to take the sequence flow elements and analyze their source and target reference elements, so that we could understand the flow of any process and build a natural language explanation about it.

For that, we created a recursive function called “analyzeNextElement” that will always take the next target element of a sequence flow and follow the process until the end. In its first call, the next target element is undefined, so it looks for the start event. Then, after it finds it, it will look for the sequence flow that has the source element as that start event, and then call the function again populating the next target element of that sequence flow.

²<https://github.com/bpmn-io/bpmn-moddle>

This function alters an output string variable that starts as an empty string, and after each User Task that the function finds, it will add another piece to that variable including the name of the current task that is being analyzed. It also uses a task counter that starts at the value “1” to keep track of the order of each task. So, if the first task is called “Book flight”, then the first piece of the sentence to be added will be: “The task number 1 is Book flight”.

If the function runs into an exclusive gateway, it will create a fork in the output string. In case of using exclusive gateways in the process model, the gateway should have a semantically reasonable name, as the name will go into the process explanation. For example, if the gateway refers to whether a user wants to book a transfer or not, the gateway could be called “Transfer preference”. In this case, what would be added to the output string would be the following piece: “You will then need to ask the user about the Transfer preference.”.

Because an exclusive gateway is a fork in the process, the output message needs to refer to all the possible ways coming from the gateway. Therefore, the function will look for the conditional expressions of each outgoing sequence flow coming from the exclusive gateway, and use them to qualify the process flows in the message. Usually, the condition expressions will be logical expressions that can be analyzed by machines, but in this case, because we are writing a natural language message, it needs to be a piece of text in English that clearly explains what that specific flow refers to. For example, in the case of the “Transfer preference” flow, there can be two outgoing flows with the following condition expressions: “If the user wants to book a transfer” and “If the user does not want to book a transfer”.

As mentioned in the previous section, for each outgoing sequence flow, the task counter will start from the same number since the different flows are mutually exclusive. Therefore, for one flow the message will say “The task number 3 is X”, while for the other flow, the message will say “The task number 3 is Y”. Both pieces should appear in the same message, because they refer to different flows and only one of them will take place.

Finally, when reaching the end event, the message will simply state “And then the process ends.” This is useful for the language model to understand that there is not a continuation of the process after the last task. The final output message referring to the Trip Planning BPMN model depicted in Fig. 3.1, with all elements is as follows:

The task number 1 is Book flight. The task number 2 is Book hotel. You will then need to ask the user about the Transfer preference.

If the user wants to book a transfer, The task number 3 is Book transfer. The task number 4 is Book tour. And then the process ends.

If the user does not want to book a transfer, The task number 3 is Book tour.
And then the process ends.

In this solution, we are not dealing with parallel gateways or other types of elements, because we wanted to analyze how this solution would work with this limited version, and if it made sense, we would continue with implementing the other BPMN elements. To initialize the GPT with the context above, we would first send an introductory message relaying how we wanted GPT to act. The introductory message that we crafted was:

You are a chatbot that helps users with business processes based on a BPMN model. I will be listing the required tasks in the following messages.

After the introductory message is sent, the next step is to send the process context, i.e. the set of sentences previously shown explaining each process element, including tasks, gateways, etc. After everything has been captured by the GPT model, we then send a final message before the conversation actually begins:

Now the conversation will start. Reply to me greeting me and saying what tasks are available at the beginning.

At the outset, this approach appears to hold promise in helping users with the described process. However, there were a couple of challenges that emerged with this approach.

The first one is that each element type that is depicted in the BPMN notation needed to be converted to a natural language sentence. This is a problem because, a) there are too many element types that exist in BPMN, and thus, it is hard to describe them all; and b) there are some intricacies of the BPMN notation that are hard to explain in sentences, such as exclusive and parallel gateways and their outgoing flows. These characteristics may be easy for a human to understand when seeing the graphical depiction of the process model, or for a process engine that is specialized in reading this notation, but hard to explain in words.

The second issue is that, even for element traits that are easy to explain, the GPT model still had some trouble understanding the BPMN limitations when we tried to convert the process model to natural language sentences because it did not understand how strict the BPMN requirements were.

Here's an excerpt of a first example conversation:

Chatbot: Hello! I'm your BPMN chatbot assistant, here to help you with your business processes. Today, I can assist you with the tasks related to travel bookings. Following the BPMN model you provided, let's begin by booking your flight.

Task 1: Book Flight

Please provide me with the necessary details for your flight booking, such as the departure and destination airports, preferred dates, and any specific airline preferences you may have.

Once we have your flight details, we'll move on to the next task. Please let me know when you're ready to proceed by providing the task number or any other information you'd like to share.

User: Can I book the hotel first?

Chatbot: Apologies for the confusion. According to the provided BPMN model, the booking tasks are structured in a specific order. However, I can certainly accommodate your request to book the hotel first.

Let's adjust the sequence to meet your preference. We'll begin with:

Task 2: Book Hotel

Please provide me with the necessary details for your hotel booking, such as the desired location, check-in and check-out dates, the number of guests, and any specific hotel preferences you may have.

As it is possible to see in the excerpt above, the chatbot did not follow the BPMN requirement that a task should only be executed when the previous one was completed. Interestingly, when asked the following question: "In BPMN, can a task be executed if the previous task was not yet completed?", GPT answered the following:

In BPMN, the execution of a task is generally based on the sequence flow, which represents the order of activities within a process. By default, a task in BPMN should not be executed until the previous task is completed. The sequential flow of tasks ensures that each task is executed in order, following a predefined sequence.

Despite knowing that, by default, tasks in BPMN should be executed in order, GPT still did not follow this requirement when executing the process that had been provided in

the prompt. Therefore, even though our original plan was to try out two different process models (Trip Planning and Wedding Planning) on both chatbots (GPT and PaLM), we decided to cut this initial experiment short after seeing the subpar results from GPT with this natural language strategy, and explore a different approach that could make these chatbot models understand the BPMN process context and requirements in a better way. In the next chapter, we will go over an improved approach, including evaluation criteria to analyze chatbot adherence to BPMN requirements and the four use cases – Trip Planning and Wedding Planning on both GPT and PaLM – that we used to perform this evaluation.

Chapter 5

Generative-based process-aware chatbots: Improved approach

After seeing the drawbacks of using our first approach, we decided to improve our strategy by providing the chatbot with the entire XML file for a BPMN model and letting it interpret it on its own. In this chapter, we will send the XML files related to two BPMN process models to two different LLM chatbots (GPT-3.5 and PaLM 2), in an attempt to see how well each model follows the original process model, based on a set of evaluation questions that will be described in Section 5.2.

It is important to remember that neither this improved approach nor the initial one was directly connected to a process engine like Camunda due to the unavailability of plugins or integrations for the LLM chatbots we were exploring, namely GPT and PaLM. Therefore, without this connection in place, it was unclear if these generative-based chatbots were going to be able to stick to the BPMN requirements determined by the process models that we would use. The following sections of this chapter explore the evaluation of these chatbots to assess their capability of meeting the requirements of BPMN constructs as defined in the official BPMN specification, to ultimately understand how well these chatbots can navigate and interact with BPMN-based processes.

5.1 Implementation Strategy

In the previous chapter, we covered a strategy where we developed a script to generate an explanation message about the process. However, after experimenting with that, we wanted

to explore a different approach to explaining the process to the chatbot. We decided to try this alternative approach because, first, it is challenging to translate all of the complex elements of the BPMN notation into human language, and second, we wanted to see if we could improve the way generative chatbots understood the entire process, after seeing subpar results with our initial approach.

As previously mentioned, BPMN uses a set of symbols and graphical notations to represent different elements of a business process. Some common symbols used in BPMN include:

1. Activities, which can represent tasks or actions performed within a process.
2. Events, which represent something that occurs during the course of a process, such as the start or end of a process, or an intermediate point where something significant happens.
3. Gateways, which indicate points in a process where the flow can diverge or converge based on certain conditions. There are multiple types of gateways in the BPMN notation, such as parallel gateways, exclusive gateways, inclusive gateways, and more.
4. Flows, which depict the sequence and direction of activities and events within a process.
5. Artifacts, which provide additional information or context to a process, such as data objects or annotations.

In our improved approach, we decided to send directly to the chatbot the BPMN XML file, containing a structured representation of all the elements described above, and allow the chatbot to parse and comprehend autonomously all the elements within the BPMN model. This approach was based on our assumption that the chatbot would have the ability to interpret and respond to the BPMN XML file, and effectively extract the relevant process information without the need for prior conversion into natural language. For this approach, we would just start a new conversation with an introductory message – “Given the BPMN XML file below, act as an agent to help me execute this process, given BPMN restrictions. You should only list the tasks that are available at that particular moment.” – and subsequently, send to the chatbot the entire BPMN XML file, such as the ones shown in [Appendix A](#).

It is important to note that a few changes had to be made in the BPMN models in comparison to the ones that we used for the retrieval-based chatbot:

- Since these generative-based chatbots are not connected to Camunda, we had to remove all Camunda-specific fields, such as form fields, which include the “flightDate” variable and the “userWantsToBookATransfer” variable in the Trip Planning process, and the “chosenDate” variable in the Wedding Planning process;
- The exclusive gateway in the Trip Planning process was making use of the “userWantsToBookATransfer” variable, therefore, to make up for its loss, we had to change the expression condition at each outgoing sequence flow coming out of that gateway. The new expressions were written in natural language, so as to facilitate the understanding of the generative models. They became: “If user wants to book a transfer” and “If user does not want to book a transfer”;
- We had to remove Service Tasks from the BPMN files because generative-based chatbots, such as GPT and PaLM, are not connected to a process engine that can execute Service Tasks. Since Service Tasks are typically used to integrate with external systems or perform specific actions, they are not applicable in this context.

After making these changes to our process model, we are now ready to input our BPMN XML file into our chatbots. In the next section, we will cover details about how each chatbot model will be evaluated in regards to BPMN requirement understanding.

5.2 Comparison Methodology

In this section, we discuss the evaluation questions (EQs) that we will use to compare the generative-based chatbots that are introduced throughout this chapter, to understand if they follow the BPMN requirements as specified by the official BPMN specification [36]. These evaluation questions were thoughtfully designed and are categorized into six distinct categories, each addressing specific aspects of the chatbots’ performance and adherence to BPMN standards. The six categories include:

1. **Start Event:** These questions focus on assessing the chatbots’ ability to initiate processes in line with BPMN’s Start Event requirements.
2. **Forward Flow:** The evaluation questions in this category delve into the chatbots’ capability to maintain the sequential flow of activities and decisions within a process.
3. **Previous History:** This category of questions examines the chatbots’ handling of historical data and their ability to remember past information at any given time.

4. **End Event:** Here, we assess the chatbots' proficiency in finishing processes appropriately, following BPMN's End Event guidelines.
5. **Process Decision Points:** This category of evaluation questions analyzes how well the chatbots manage and manipulate forks in the process.
6. **Unintended Paths:** The final category of evaluation questions checks if the chatbot allows the user to perform actions that are not supposed to be executed at a given moment.

By categorizing the EQs into these six distinct areas, we aim to provide a comprehensive assessment of the generative-based chatbots' BPMN compliance, covering various critical aspects of process modeling and execution. We will consider each question to be a criterion that can be either met, partially met, or not met, and in the description of each EQ below, we will explain what kind of behaviour we will consider as meeting the criterion.

5.2.1 Category 1: Start Event

EQ1: Has the process instance been initiated successfully after triggering the start event?

Represented by sending the chatbot the following message: "Let's start a new process instance and name it X1."

This evaluation question assesses the chatbot's ability to initiate a new BPMN process instance upon activation of the start event. This criterion will be considered as met if the chatbot appropriately acknowledges the start event's trigger and initializes a new process instance as specified.

EQ2: Does the chatbot correctly mention the first task that follows the start event after the process instance is initiated?

Represented by asking the chatbot: "What tasks are available to me at the beginning?" and verifying if only the first task is mentioned.

This question evaluates the chatbot's capability to guide users through the BPMN process by correctly identifying and presenting the initial task immediately following the start event. It ensures that the chatbot accurately aligns with the predefined process flow. This criterion will be considered as met if the chatbot accurately presents the first task, and only the first task, after starting the process.

5.2.2 Category 2: Forward Flow

EQ3: After completing a task, if it is followed by another task directly connected by a direct sequence flow, does the chatbot correctly mention the next task?

Represented by analyzing the chatbot's response after informing the chatbot that a task was completed.

This evaluation question focuses on the chatbot's ability to maintain the flow of the BPMN process, and this criterion will be considered as met if the next task in the sequence is accurately identified and presented to the user after the completion of a task.

EQ4: After completing a task, does the chatbot avoid mentioning tasks that are not yet available in the process flow?

Represented by analyzing the chatbot's response after informing the chatbot that a task was completed.

This question examines whether the chatbot refrains from presenting tasks that are not yet accessible based on the BPMN process model, ensuring that the chatbot maintains process integrity. This criterion will be considered as met if the chatbot abstains from showing the unavailable tasks at all points during the process execution.

EQ5: After completing a task and encountering an exclusive gateway, does the chatbot correctly evaluate the gateway based on the conditions of its outgoing sequence flows and present only one valid flow to the user?

Represented by analyzing the chatbot's response after informing the chatbot that a task was completed.

This evaluation question analyzes the chatbot's handling of exclusive gateways, ensuring that only one valid path is offered to the user in alignment with the BPMN specification, even if multiple paths meet the conditions. This criterion will be considered as met if the chatbot correctly evaluates the condition of the outgoing sequence flows of an exclusive gateway, and presents to the user only a single path, which must be available based on the evaluating condition. If the chatbot shows more than one flow or shows a flow that is not available, this criterion is not met.

EQ6: After completing a task and encountering a diverging parallel gateway, does the chatbot correctly present all outgoing flows to the user?

Represented by analyzing the chatbot's response after informing the chatbot that a task was completed.

This question investigates the chatbot’s behaviour when faced with diverging parallel gateways, and this criterion is met if the chatbot accurately offers all available outgoing flows to the user, aligning with the BPMN specification.

EQ7: When reaching a converging parallel gateway, does the chatbot advance only when all parallel paths have been completed?

Represented by analyzing the chatbot’s response after informing the chatbot that a task was completed.

This evaluation question ensures that the chatbot adheres to the BPMN specification by advancing the process only when all parallel paths have been successfully completed, maintaining the integrity of parallel execution. This criterion will be considered as met if the chatbot refrains from continuing the process execution before all the parallel paths preceding the converging parallel gateway are finished.

5.2.3 Category 3: Previous History

EQ8: When presenting a task to the user, if it was immediately preceded by another task through a sequence flow, does the chatbot correctly identify the preceding task?

Represented by asking the chatbot the following question: “What was the previous task that was executed?”

This question assesses the chatbot’s knowledge of the previous task in the BPMN process flow when presenting a task to the user, ensuring the chatbot understands the current status and history of the process execution. This criterion will be considered as met if the chatbot correctly mentions the task that last task that was executed before this prompt was sent.

EQ9: When presenting a task to the user, if it was preceded by an exclusive gateway, does the chatbot accurately convey the decision represented by the preceding gateway?

Represented by asking the chatbot the following question: “Why is this the only available task?”

This evaluation question examines the chatbot’s ability to explain the rationale behind presenting a specific task when it follows an exclusive gateway, ensuring the chatbot remembers the gateway that has just been passed through. This criterion will be considered

as met if the chatbot correctly mentions the previous gateway and the evaluating condition that made the current flow available.

EQ10: When presenting a task to the user, if it was preceded by a parallel gateway, does the chatbot acknowledge the existence of multiple execution flows?

Represented by asking the chatbot the following question: “Are all these tasks from a single flow or multiple flows?”

This question ensures that the chatbot is aware of the presence of parallel flows when presenting a task. This criterion will be considered as met if the chatbot successfully explains that they are from different flows given that they come out of a parallel gateway.

5.2.4 Category 4: End Event

EQ11: Upon reaching an end event, does the chatbot accurately identify the task that immediately preceded it in the process flow?

Represented by asking the chatbot the following question: “What was the last task before the process instance was ended?”

This evaluation question assesses the chatbot’s ability to provide users with information about the task that directly preceded the termination of a process instance. This criterion will be considered as met if the chatbot correctly mentions the task that last task that was executed before the process instance was finished.

EQ12: When reaching an end event, does the chatbot ensure that the process instance is terminated only when all start events have been triggered and all pending flows have been completed?

Represented by asking the chatbot the following question: “Had all the start events been triggered, and had all pending flows been finished?”

This question evaluates the chatbot’s process termination logic, verifying that it aligns with the BPMN specification by ensuring all prerequisites for process completion are met. This criterion will be considered as met if the chatbot correctly identifies the status of all start events and flows of that process instance and confirms the BPMN requirements for ending a process instance were successfully met.

5.2.5 Category 5: Process Decision Points

EQ13: At the end of the process, does the chatbot retain knowledge of all the decisions that had to be made at gateways?

Represented by asking the chatbot the following question: “What were the decisions made at each gateway?”

This evaluation question examines the chatbot’s ability to recall and present information about the variables that influenced decision-making at various gateways throughout the BPMN process. This criterion will be considered as met if the chatbot accurately remembers all the gateways in the process and each of their outgoing flows.

EQ14: At the end of the process, does the chatbot retain knowledge of the user’s choice at each gateway?

Represented by asking the chatbot the following question: “What was the user’s choice at each gateway for this particular process instance, X1?”

This question delves deeper into the chatbot’s memory capabilities, verifying whether it can not only recall variables but also their corresponding values. This criterion will be considered as met if the chatbot accurately remembers the user’s choice that was made at each gateway, i.e. which outgoing flow was in fact taken.

5.2.6 Category 6: Unintended paths

EQ15: Does the chatbot allow the user to perform a task that is not yet available in the process?

Represented by having the user tell the chatbot that they want to execute a task that is not on the list of available tasks that the chatbot sent.

This evaluation question assesses the chatbot’s ability to prevent users from executing tasks that are not yet available. This criterion will be considered as met if the user tries to execute a task that is not yet available and the chatbot informs them that that task cannot be executed at the moment.

EQ16: Does the chatbot allow the user to end the process when it is not supposed to?

Represented by asking the chatbot the following question: “Can I end the process now?”, before all the required tasks are done.

This question evaluates the chatbot’s ability to prevent users from prematurely terminating a process before they have completed all the necessary tasks. This criterion will be considered as met if the user asks to end the process before executing all the required tasks, and the chatbot informs them that the process cannot be ended yet because there are still tasks to be done.

5.3 GPT

5.3.1 Use Case 1: Trip Planning

After having implemented a retrieval-based BPMN-supporting chatbot for the Trip Planning scenario in Chapter 3, we now attempt to implement the same scenario using our chosen strategy, which is based on generative language models. To make GPT understand the Trip Planning process model, the BPMN XML file for the Trip Planning process model depicted in Fig. 3.1 is sent to the chatbot. The full XML file can be seen in Appendix A.1.

Category 1: Start Event

After creating a new process instance named X1 and asking for the available tasks, GPT-3.5 correctly initiates the process instance and presents the only available task, as shown by the conversation below:

User: Let’s start a new process instance and name it X1. What tasks are available to me at the beginning?

Chatbot: In the provided BPMN XML, you’re starting a new process instance named “X1” in the “Simple Trip Planning” process. At the beginning of this process instance, the following tasks are available:

1. Book flight (User Task)
Task ID: book_flight
Task Name: Book flight

These are the tasks that can be executed when starting process instance “X1.” You can begin by performing the “Book flight” task.

Therefore, both EQ1 (Has the process instance been initiated successfully after triggering the start event?) and EQ2 (Does the chatbot correctly mention the first task that follows the start event after the process instance is initiated?) are met for this particular use case.

Category 2: Forward Flow

After booking the flight, the next task, “Book hotel” which is directly connected to “Book flight” through a sequence flow, is presented to the user.

User: I’ve booked the flight

Chatbot: Great! You’ve booked the flight. Now, let’s determine the next available tasks in the process. After booking the flight, you can see the following available tasks:

1. Book hotel (User Task)
Task ID: book_hotel
Task Name: Book hotel

To continue the process, you can now proceed to book a hotel by executing the “Book hotel” task.

Therefore, EQ3 (After completing a task, if it is followed by another task directly connected by a direct sequence flow, does the chatbot correctly mention the next task?) is also met.

Now, after the user mentions that they completed the “Book hotel” task, a strange behaviour occurred: the chatbot assumed the gateway that followed the “Book hotel” task was also a task, as shown below:

User: I’ve booked the hotel

Chatbot: Excellent! You’ve booked the hotel. Now, let’s find out the next available tasks in the process. After booking the hotel, the following task is available:

1. Transfer preference (Exclusive Gateway)
Gateway ID: Gateway_01f6asu
Gateway Name: Transfer preference

At this point, you need to make a choice based on your transfer preference. You can either book a transfer or proceed with booking a tour. Please let me know your preference, and I will guide you accordingly.

However, because the gateway should not have been presented as a task, we decide to raise an objection to the answer, at which point the chatbot corrects itself:

User: Is this gateway a task?

Chatbot: I apologize for the confusion. The “Transfer preference” (Gateway_01f6asu) is not a task; it is a gateway element used for making decisions in the process flow. It represents a branching point where the process flow diverges based on a condition.

To clarify, the available tasks after booking the hotel are as follows:

1. Book transfer (User Task)
Task ID: book_transfer
Task Name: Book transfer
2. Book tour (User Task)
Task ID: book_tour
Task Name: Book tour

You can choose to execute either the “Book transfer” or “Book tour” task based on your preference.

Then, the user proceeds with the process execution as expected:

User: I want to book a transfer

Chatbot: Great choice! Since you want to book a transfer, you should proceed with the “Book transfer” task.

1. Book transfer (User Task)
Task ID: book_transfer
Task Name: Book transfer

You can now execute the “Book transfer” task as the next step in the process.

And finally, after mentioning that the transfer was booked, the last User task, “Book tour” is correctly presented.

User: I’ve booked the transfer

Chatbot: Excellent! You’ve successfully booked the transfer. Now, after booking the transfer, the next available task in the process is the “Book tour” task.

1. Book tour (User Task)
Task ID: book_tour
Task Name: Book tour

You can now proceed with the “Book tour” task to continue the process.

With this conversation flow, we realize that EQ4 (After completing a task, does the chatbot avoid mentioning tasks that are not yet available in the process flow?) is met, because GPT-3.5 does not mention any task that is not yet available in the process flow. However, we argue that EQ5 (After completing a task and encountering an exclusive gateway, does the chatbot correctly evaluate the gateway based on the conditions of its outgoing sequence flows and present only one valid flow to the user?) is only partially met, because the gateway should have been evaluated automatically by asking the user for its preference, instead of presenting the gateway as a task that the user had to execute manually.

EQ6 (After completing a task and encountering a diverging parallel gateway, does the chatbot correctly present all outgoing flows to the user?) and EQ7 (When reaching a converging parallel gateway, does the chatbot advance only when all parallel paths have been completed?) are not applicable to this use case, since the Trip Planning process model does not contain any parallel gateways. Thus, they are skipped for this use case.

Category 3: Previous History

For EQ8, (When presenting a task to the user, if it was immediately preceded by another task through a sequence flow, does the chatbot correctly identify the preceding task?), we ask the chatbot what was the previous task executed, after executing the “Book flight” task, and it responds appropriately. Therefore, EQ8 is met, as exemplified by the excerpt below:

User: What was the previous task that was executed?

Chatbot: The previous task that was executed was “Book flight,” with a User Task ID of “book_flight.”

Later in the process, after the user reaches the exclusive gateway and says that they want to book a transfer, the chatbot appropriately presents the “Book transfer” task. When questioned, why this is the only available task, the chatbot provides a valid answer:

User: I want to book a transfer

Chatbot: Great choice! Since you want to book a transfer, you should proceed with the “Book transfer” task.

1. Book transfer (User Task)
Task ID: book_transfer
Task Name: Book transfer

You can now execute the “Book transfer” task as the next step in the process.

User: Why is this the only available task?

Chatbot: In the BPMN process you provided, after booking the flight and then booking the hotel, you reached an exclusive gateway named “Transfer preference.” This gateway splits the process flow into two alternative paths based on user preference. The available tasks depend on the conditions specified in the gateway’s outgoing sequence flows:

- If the user wants to book a transfer, you proceed to the “Book transfer” task.
- If the user does not want to book a transfer, you would go to the “Book tour” task.

Since you indicated that you wanted to book a transfer, the process follows the path that leads to the “Book transfer” task. This is why “Book transfer” is the only available task at this point in the process. The other path, which involves booking a tour, is not taken in this scenario.

Therefore, EQ9 (When presenting a task to the user, if it was preceded by an exclusive gateway, does the chatbot accurately convey the decision represented by the preceding gateway?) is also met. Meanwhile, EQ10 is only related to parallel gateways, and therefore, is skipped for this use case.

Category 4: End Event

Then, after the transfer is booked and the “Book tour” task is presented, when the user informs the chatbot that the tour has been booked, the chatbot appropriately finish the process instance:

User: I’ve booked the tour

Chatbot: Great! You’ve booked the tour. After booking the tour, the process reaches its conclusion, as the “Book tour” task is followed by the “EndEvent_1pckwqd,” which signifies the end of the process instance. Your process instance “X1” has been successfully completed with the booking of the tour.

Then, after asking the two questions that refer to EQ11 and EQ12, both are answered accordingly.

User: What was the last task before the process instance was ended?

Chatbot: The last task executed just before the process instance was ended was the “Book tour” task.

User: Had all the start events been triggered, and had all pending flows been finished?

Chatbot: In the BPMN process you provided, there was only one start event, which is “StartEvent.1.” This start event was triggered to initiate the process instance. All of the pending flows in the process have been executed and completed as you progressed through the tasks. Therefore, all start events have been triggered, and all pending flows have been finished for this specific process instance “X1.”

Hence, EQ11 (Upon reaching an end event, does the chatbot accurately identify the task that immediately preceded it in the process flow?) and EQ12 (When reaching an end event, does the chatbot ensure that the process instance is terminated only when all start events have been triggered and all pending flows have been completed?) are met.

Category 5: Process Decision Points

Finally, we ask some questions to the chatbot regarding the process decision points:

User: What were the decisions made at each gateway?

Chatbot: In the provided BPMN process, there is an exclusive gateway with the ID “Gateway_01f6asu” named “Transfer preference.” Decisions are made at this gateway based on the conditions specified in the outgoing sequence flows. Here are the decisions made at each gateway:

Exclusive Gateway “Transfer preference” (ID: Gateway_01f6asu):

- If the condition “If user wants to book a transfer” is true, the process flow goes to the “Book transfer” task.
- If the condition “If user does not want to book a transfer” is true, the process flow goes to the “Book tour” task.

These decisions determine whether the user wants to book a transfer or not, and the subsequent tasks are executed accordingly based on these decisions.

User: What was the user’s choice at each gateway for this particular process instance, X1?

Chatbot: In the context of the specific process instance “X1,” the user’s choice at each gateway was as follows:

Exclusive Gateway “Transfer preference” (ID: Gateway_01f6asu):

- The user’s choice was to book a transfer, as indicated in your interactions. This choice led to the execution of the “Book transfer” task.

Throughout the process instance “X1,” the user consistently chose to book a transfer at the gateway, which determined the subsequent tasks in the process flow.

With these answers, we also note that EQ13 (At the end of the process, does the chatbot retain knowledge of all the decisions that had to be made at gateways?) and EQ14 (At the end of the process, does the chatbot retain knowledge of the user’s choice at each gateway?) are also met.

Category 6: Unintended Paths

If the user starts a new process instance, and tries to execute tasks in the wrong order, for this particular use case, GPT fails to prevent the wrongful execution. As shown by

the initial conversation below, it seemed as though the chatbot would not let the user execute tasks in the wrong order, however, further into the conversation, the chatbot makes incorrect assumptions and allows for the execution of unavailable tasks.

User: Let's start a new process instance X2, of the same process. What tasks are available to me?

Chatbot: When starting a new process instance "X2" of the same "Simple Trip Planning" process, the available tasks at the beginning are the same as for the initial process instance "X1." Therefore, the available task is:

1. Book flight (User Task)
Task ID: book_flight
Task Name: Book flight

You can start by executing the "Book flight" task for process instance "X2" as well.

User: I've booked the flight

Chatbot: Great! You've booked the flight for process instance "X2." After booking the flight, the next available tasks in the process are the same as for the previous instance "X1." You can now proceed with the following task:

1. Book hotel (User Task)
Task ID: book_hotel
Task Name: Book hotel

You can continue the process by executing the "Book hotel" task.

User: Can I book the transfer now?

Chatbot: No, you cannot book the transfer at this point. In the process flow, the option to book the transfer is determined by a conditional gateway ("Transfer preference") that occurs after booking the hotel. This gateway has two outgoing sequence flows:

- If the user wants to book a transfer, the flow goes to the "Book transfer" task.
- If the user does not want to book a transfer, the flow goes to the "Book tour" task.

Since you have booked the flight and hotel, the next step is to book the hotel, and the decision to book a transfer or not will be determined after booking the hotel. So, you will need to complete the “Book hotel” task first before the option to book the transfer becomes available.

However, if the user does not book the hotel or the transfer, and insists on executing another wrong task, namely the “Book tour” task, the chatbot gets confused and informs the user that the “Book tour” task is available, even though it cannot be reached directly from the “Book flight” task.

User: Can I book the tour?

Chatbot: Yes, you can book the tour at this point. After booking the flight and hotel, the next available tasks are determined by the exclusive gateway (“Transfer preference”), which has two outgoing sequence flows:

- If the user wants to book a transfer, the flow goes to the “Book transfer” task.
- If the user does not want to book a transfer, the flow goes to the “Book tour” task.

Since you mentioned that you’ve booked the flight and hotel, it implies that you do not want to book a transfer, so you can proceed to book the tour. The available task is:

1. Book tour (User Task)
Task ID: book_tour
Task Name: Book tour

You can continue the process by executing the “Book tour” task.

As it is possible to see, the chatbot assumed the user had already booked the hotel even though that was not explicitly mentioned. Furthermore, it also assumed that the user did not want to book a transfer, despite not explicitly asking the user for their choice. Therefore, EQ15 (Does the chatbot allow the user to perform a task that is not yet available in the process?) is not met.

On the other hand, after completing the “Book flight” task, if the user mentions that they want to end the process, the chatbot does not allow the user to prematurely end the process, which means EQ16 (Does the chatbot allow the user to end the process when it is not supposed to?) is met, as shown in the excerpt below:

User: I've booked the flight

Chatbot: Great! You've booked the flight for process instance "X2." After booking the flight, the next available tasks in the process are the same as for the previous instance "X1." You can now proceed with the following task:

1. Book hotel (User Task)
Task ID: book_hotel
Task Name: Book hotel

You can continue the process by executing the "Book hotel" task.

User: Can I end the process now?

Chatbot: No, you cannot end the process at this point. In the provided BPMN model, there is no end event immediately following the "Book hotel" task, so the process must continue with further tasks. To complete the process, you'll need to follow the sequence defined in the BPMN model. Please proceed with the "Book hotel" task, and then we can determine the next steps in the process based on the BPMN model.

Overall, GPT-3.5 meets most of the criteria used in our evaluation. However, EQ15, which prevents the user from executing unavailable tasks, was not met, and EQ5, related to analyzing the decision at a gateway, was only partially met, due to an unexpected behaviour when reaching an exclusive gateway. That suggests that GPT-3.5 might have difficulties dealing with exclusive gateways and following the correct order of tasks if there are strict dependencies between them. EQ6, EQ7 and EQ10, which were skipped due to lack of parallel gateways in this use case. In the next subsection, we will move on to the second use case, which is the Wedding Planning process model.

5.3.2 Use Case 2: Wedding Planning

We now move on to the second process model used in Chapter 3, the Wedding Planning model, using our strategy of sending its BPMN XML file to a generative-based language model, namely GPT. To make GPT understand the Wedding Planning process model, the XML document that referred to the BPMN diagram depicted in Fig. 3.4 can be seen in Appendix A.2.

Category 1: Start Event

After creating a new process instance named X1 and asking for the available tasks, GPT-3.5 correctly initiates the process instance and presents the only available task, as shown by the conversation below:

User: Let's start a new process instance and name it X1. What tasks are available to me at the beginning?

Chatbot: In the provided BPMN XML, we are starting a new process instance named "X1." At the beginning of the process, you have the following tasks available:

1. Choose date (User Task with ID "Activity_1iit2w9")

You can start the process by performing the "Choose date" task.

For this particular use case, EQ1 (Has the process instance been initiated successfully after triggering the start event?) and EQ2 (Does the chatbot correctly mention the first task that follows the start event after the process instance is initiated?) are met.

Category 2: Forward Flow

After choosing the date, the process flow reaches a parallel gateway, and the following four tasks can be executed in any order.

User: I've chosen the date

Chatbot: Great! After choosing the date, the process flow advances. Here are the tasks available at this point:

1. Book band (User Task with ID "Activity_0mqww1k")
2. Book photographer (User Task with ID "Activity_0hzejli")
3. Book caterer (User Task with ID "Activity_0szmooh")
4. Book venue (User Task with ID "Activity_0lhgvkj")

You can choose any of these tasks to proceed with the process.

According to this interaction, EQ6 (After completing a task and encountering a diverging parallel gateway, does the chatbot correctly present all outgoing flows to the user?) is also met.

The user then proceeds to execute each task:

User: I've booked the photographer

Chatbot: Excellent! After booking the photographer, the process flow advances. Here are the tasks available at this point:

1. Book band (User Task with ID "Activity_0mqww1k")
2. Book caterer (User Task with ID "Activity_0szmooh")
3. Book venue (User Task with ID "Activity_0lhgvkj")

You can now choose one of these remaining tasks to continue with the process.

User: I've booked the band

Chatbot: Great choice! After booking the band, the process flow advances. Here are the tasks available at this point:

1. Book caterer (User Task with ID "Activity_0szmooh")
2. Book venue (User Task with ID "Activity_0lhgvkj")

You can now choose one of these remaining tasks to continue with the process.

User: I've booked the caterer

Chatbot: Fantastic! After booking the caterer, the process flow advances, and there is only one task remaining:

1. Book venue (User Task with ID "Activity_0lhgvkj")

You can now proceed to book the venue to complete the process.

User: I've booked the venue

Chatbot: Perfect! You have now completed all the required tasks, and the process flow has reached its conclusion. There are no more tasks to be performed. Congratulations on successfully executing the entire process! If you have any more questions or need further assistance, feel free to ask.

As seen in the conversation excerpt above, only the available tasks are mentioned at every point. Therefore, EQ4 (After completing a task, does the chatbot avoid mentioning tasks that are not yet available in the process flow?) is appropriately met. Furthermore, the process only proceeds to the end after all tasks have been executed. For that reason, EQ7 (When reaching a converging parallel gateway, does the chatbot advance only when all parallel paths have been completed?) is also met.

Because there are no tasks directly connected by a sequence flow to another task, EQ3 (After completing a task, if it is followed by another task directly connected by a direct sequence flow, does the chatbot correctly mention the next task?) is not applicable. Another EQ that is not applicable is EQ5 (After completing a task and encountering an exclusive gateway, does the chatbot correctly evaluate the gateway based on the conditions of its outgoing sequence flows and present only one valid flow to the user?), related to exclusive gateways, which are not part of this process model.

Category 3: Previous History

For EQ10, (When presenting a task to the user, if it was preceded by a parallel gateway, does the chatbot acknowledge the existence of multiple execution flows?), after booking the photographer and having the chatbot list the other 3 available tasks, we ask if those tasks are all part of the same flow or multiple flows, and the chatbot correctly explains that they are part of multiple flows. Therefore, EQ10 is met, as exemplified by the excerpt below:

User: Are all these tasks from a single flow or multiple flows?

Chatbot: The tasks “Book band” (User Task with ID “Activity_0mqww1k”), “Book caterer” (User Task with ID “Activity_0szmooh”), and “Book venue” (User Task with ID “Activity_0lhgvkj”) are all parallel tasks and are part of multiple flows. They are not part of a single flow but are independent parallel branches that can be executed concurrently. These tasks are connected to the parallel gateway (Gateway_1ka16zc) and can be performed simultaneously if needed.

For this use case, EQ8 (When presenting a task to the user, if it was immediately preceded by another task through a sequence flow, does the chatbot correctly identify the preceding task?) and EQ9 (When presenting a task to the user, if it was preceded by an exclusive gateway, does the chatbot accurately convey the decision represented by the

preceding gateway?) are not applicable. EQ8 is skipped because there are no tasks directly connected to other tasks without a gateway in between, and EQ9 is skipped because there are no exclusive gateways in this process.

Category 4: End Event

Then, after all the required tasks are executed, the chatbot appropriately finish the process instance:

User: I've booked the venue

Chatbot: Perfect! You have now completed all the required tasks, and the process flow has reached its conclusion. There are no more tasks to be performed. Congratulations on successfully executing the entire process! If you have any more questions or need further assistance, feel free to ask.

Subsequently, after asking the two questions that refer to EQ11 and EQ12, both are answered accordingly.

User: What was the last task before the process instance was ended?

Chatbot: The last task before the process instance was ended was “Book venue” (User Task with ID “Activity_0lhgvkj”). Once you booked the venue, the process flow continued to the end event, and the process instance was successfully completed.

User: Had all the start events been triggered, and had all pending flows been finished?

Chatbot: Yes, based on the provided BPMN diagram and your previous interactions, it appears that all start events have been triggered, and all pending flows have been finished. Specifically:

1. The initial start event “StartEvent_1” was triggered to initiate the process instance.
2. The flow from the start event to the “Choose date” task was completed.
3. After “Choose date,” all parallel branches from the parallel gateway “Gateway_0m3be6e” were completed successfully.

4. Each of the user tasks within those parallel branches (“Book band,” “Book photographer,” “Book caterer,” and “Book venue”) was triggered and completed.

5. The process instance reached the end event “Event_0tey7i4.”

So, all start events were triggered, and all pending flows were indeed finished, resulting in the successful completion of the process instance.

Hence, EQ11 (Upon reaching an end event, does the chatbot accurately identify the task that immediately preceded it in the process flow?) and EQ12 (When reaching an end event, does the chatbot ensure that the process instance is terminated only when all start events have been triggered and all pending flows have been completed?) are met.

Category 5: Process Decision Points

For the Wedding Planning use case, EQ13 (At the end of the process, does the chatbot retain knowledge of all the decisions that had to be made at gateways?) and EQ14 (At the end of the process, does the chatbot retain knowledge of the user’s choice at each gateway?) are both skipped because of the lack of gateways that require decisions. A parallel gateway means that every outgoing flow should be presented to the user, with no decision being involved, unlike exclusive gateways where each outgoing flow has a specific condition and only one flow should be presented. Therefore, there are no decision points in this process.

Category 6: Unintended Paths

If the user starts a new process instance, and tries to execute tasks in the wrong order, for this particular use case, the chatbot successfully prevents the wrongful execution:

User: Let’s start a new process instance X2, of the same process. What tasks are available to me?

Chatbot: Starting a new process instance “X2” of the same process, at the beginning, you have the following task available:

Choose date (User Task with ID “Activity_1iit2w9”)

You can start the process by performing the “Choose date” task, just like in the previous instance.

User: Can I book the band first?

Chatbot: In the provided BPMN process, the task “Choose date” (User Task with ID “Activity_1iit2w9”) is the initial task, and it should be completed before moving on to other tasks. Therefore, you need to start by choosing the date first, and then you can proceed to book the band or perform other tasks.

User: Can I book the venue first?

Chatbot: In the provided BPMN process, the task “Choose date” (User Task with ID “Activity_1iit2w9”) is the initial task, and it should be completed before moving on to other tasks. Therefore, you need to start by choosing the date first, and then you can proceed to book the venue or perform other tasks. The process follows a specific sequence, and tasks must be completed in the defined order.

Therefore, EQ15 (Does the chatbot allow the user to perform a task that is not yet available in the process?) is met. Additionally, the chatbot does not allow the user to prematurely end the process, which means EQ16 (Does the chatbot allow the user to end the process when it is not supposed to?) is also met, as shown in the excerpt below:

User: I’ve booked the band

Chatbot: Excellent! After booking the band, the process flow advances. Here are the tasks available at this point:

1. Book photographer (User Task with ID “Activity_0hzejli”)
2. Book caterer (User Task with ID “Activity_0szmooh”)
3. Book venue (User Task with ID “Activity_0lhgvkj”)

You can now choose one of these remaining tasks to continue with the process.

User: Can I end the process now?

Chatbot: I apologize for any confusion, but the process cannot be ended at this point. In the provided BPMN process, all the tasks must be completed in sequence before reaching the end event. Since you have already booked the band, you need to proceed with booking the photographer, caterer, and venue before reaching the end event.

So, you should choose one of the remaining tasks (“Book photographer,” “Book caterer,” or “Book venue”) to continue the process.

Overall, GPT-3.5 meets all the criteria used in our evaluation, except for the ones that were skipped due to the lack of directly connected tasks and exclusive gateways (EQ3, EQ5, EQ8, EQ9, EQ13, EQ14). That suggests that, while GPT-3.5 might face challenges dealing with exclusive gateways and strict dependencies between tasks, as shown by the Trip Planning use case, it has more aptitude for following BPMN requirements when presented with less rigid dependencies, such as multiple flows coming out of a parallel gateway.

5.4 PaLM

After analyzing how well GPT adheres to both process models using their BPMN XML files, we now move on to the evaluation of Google’s PaLM model, so as to compare its execution to GPT, again with the same process models – Trip Planning and Wedding Planning. In this section, we will answer the same evaluation questions as the previous section, to be able to compare PaLM and GPT with a structured methodology.

5.4.1 Use Case 1: Trip Planning

For PaLM, we used the same XML document for the Trip Planning model as the one we used for GPT, the one included in Appendix A.1. Throughout this section, we will be analyzing each of the six main categories that are related to the adherence of PaLM to the BPMN constructs utilized in our process model.

Category 1: Start Event

After creating a new process instance named X1 and asking for the available tasks, PaLM correctly initiates the process instance and presents the only available task, as shown by the conversation below:

User: Let’s start a new process instance and name it X1. What tasks are available to me at the beginning?

Chatbot: The following tasks are available to you at the beginning of the process instance X1:

- book flight

Therefore, both EQ1 (Has the process instance been initiated successfully after triggering the start event?) and EQ2 (Does the chatbot correctly mention the first task that follows the start event after the process instance is initiated?) are met for this particular use case.

Category 2: Forward Flow

After booking the flight, the next task, “book_hotel” which is directly connected to “book_flight” through a sequence flow, is presented to the user. However, PaLM also presents another task that should not have been presented, which was the “book_transfer”.

User: I’ve booked the flight

Chatbot: The following tasks are available to you now:

- book_hotel
- book_transfer

Because the “book_hotel” task has been correctly presented, we argue that EQ3 (After completing a task, if it is followed by another task directly connected by a direct sequence flow, does the chatbot correctly mention the next task?) is met. However, because it also mentions a task that is not yet available, EQ4 (After completing a task, does the chatbot avoid mentioning tasks that are not yet available in the process flow?) is not met.

Furthermore, when we challenge the chatbot asking if it is sure that the “book_transfer” task is available, it also lists its incorrect assumption about the decision at the gateway.

User: Are you sure the book_transfer task is available?

Chatbot: Yes, the book_transfer task is available. Since you did not select the “If user does not want to book a transfer” path in the Gateway_01f6asu gateway, the book_transfer task is available.

Therefore, we argue that EQ5 (After completing a task and encountering an exclusive gateway, does the chatbot correctly evaluate the gateway based on the conditions of its outgoing sequence flows and present only one valid flow to the user?) is partially met, because the chatbot does present only one valid flow but it did not correctly evaluate the gateway based on the conditions of its outgoing sequence flow, as it did not ask the user for their explicit choice.

EQ6 (After completing a task and encountering a diverging parallel gateway, does the chatbot correctly present all outgoing flows to the user?) and EQ7 (When reaching a converging parallel gateway, does the chatbot advance only when all parallel paths have been completed?) are not applicable to this use case, since the Trip Planning process model does not contain any parallel gateways, so they are skipped for this use case.

Category 3: Previous History

For EQ8, (When presenting a task to the user, if it was immediately preceded by another task through a sequence flow, does the chatbot correctly identify the preceding task?), we ask the chatbot about the previous task executed, after completing the “Book flight” task, and it responds appropriately. Therefore, EQ8 is met, as shown in the excerpt below:

User: What was the previous task that was executed?

Chatbot: The previous task that was executed was `book_flight`.

Later in the process, after the user completes the “book_hotel” task, the chatbot fails to ask the user’s transfer preference. Instead, it lists both the “book_transfer” and “book_tour” tasks:

User: I’ve booked the hotel

Chatbot: The following tasks are available to you now:

- `book_transfer`
- `book_tour`

Even though for EQ9 (When presenting a task to the user, if it was preceded by an exclusive gateway, does the chatbot accurately convey the decision represented by the preceding gateway?), the question that should be asked was “Why is this the only available task?”, in the current conversation, this question does not make sense because the chatbot presented two tasks, and not only one, as it should have. Therefore, EQ9 is not met. Meanwhile, EQ10 is only related to parallel gateways, and therefore, is skipped for this use case.

Category 4: End Event

Then, we chose to complete the “book_tour” task before mentioning that the “book_transfer” task had been completed, to see how the chatbot would react. In this case, the chatbot finished the process instance even though the “book_transfer” task had not been executed:

User: I’ve booked the tour

Chatbot: The process instance X1 has been completed.

After asking about the “book_transfer” task, the chatbot lists its assumption that, because the user chose to execute the “book_tour” task, this implicitly meant that the user chose to follow the path that bypasses “book_transfer” task when reaching the gateway:

User: What about the “book_transfer” task? Had it been executed?

Chatbot: No, the book_transfer task had not been executed. Since you did not select the “If user wants to book a transfer” path in the Gateway_01f6asu gateway, the book_transfer task was not executed.

Given that the chatbot had the assumption that, after the gateway, the user had chosen to not book a transfer, it made sense to end the process after booking the tour. Therefore, after asking the two questions that refer to EQ11 and EQ12, both are answered accordingly:

User: What was the last task before the process instance was ended?

Chatbot: The last task before the process instance was ended was book_tour.

User: Had all the start events been triggered, and had all pending flows been finished?

Chatbot: Yes, all the start events had been triggered, and all pending flows had been finished.

Hence, EQ11 (Upon reaching an end event, does the chatbot accurately identify the task that immediately preceded it in the process flow?) and EQ12 (When reaching an end event, does the chatbot ensure that the process instance is terminated only when all start events have been triggered and all pending flows have been completed?) are met.

Category 5: Process Decision Points

Finally, we ask some questions to the chatbot regarding the process decision points:

User: What were the decisions made at each gateway?

Chatbot: The following decisions were made at each gateway:

- Gateway_01f6asu: If user wants to book a transfer

User: What was the user's choice at each gateway for this particular process instance, X1?

Chatbot: The user chose the following options at each gateway for process instance X1:

- Gateway_01f6asu: If user does not want to book a transfer

With these answers, we also note that EQ13 (At the end of the process, does the chatbot retain knowledge of all the decisions that had to be made at gateways?) and EQ14 (At the end of the process, does the chatbot retain knowledge of the user's choice at each gateway?) are also met, even though the chatbot implicitly assumed the user's choice at the gateway, without asking explicitly for their choice.

Category 6: Unintended Paths

If the user starts a new process instance, and tries to execute tasks in the wrong order, for this particular use case, just like GPT, PaLM also fails to prevent the wrongful execution of tasks.

User: Let's start a new process instance X2, of the same process. What tasks are available to me?

Chatbot: The following tasks are available to you at the beginning of the process instance X2:

- book_flight

User: I've booked the flight

Chatbot: The following tasks are available to you now:

- book_hotel

- book_transfer

User: Can I book the transfer now?

Chatbot: Yes, you can book the transfer now.

Therefore, as it is possible to see, the chatbot would let the user book the transfer before booking the hotel, which is not allowed according to the BPMN process model. Therefore, EQ15 (Does the chatbot allow the user to perform a task that is not yet available in the process?) is not met.

However, if after completing the “Book flight” task, the user mentions that they want to end the process, the chatbot does not allow the user to prematurely end the process, which means EQ16 (Does the chatbot allow the user to end the process when it is not supposed to?) is met, as shown in the excerpt below:

User: I’ve booked the flight

Chatbot: The following tasks are available to you now:

- book_hotel
- book_transfer

User: Can I end the process now?

Chatbot: No, you cannot end the process now. There are still pending tasks that need to be executed. The following tasks are still pending:

- book_hotel
- book_transfer
- book_tour

Overall, PaLM 2 has a larger failure rate than GPT in terms of meeting evaluation criteria. EQ4 and EQ15, related to mentioning and allowing the execution of tasks that are not yet available in the process are not met, EQ9 is not met and EQ5 is partially met, these last two being related to how exclusive gateways are evaluated. That suggests that PaLM 2, just like GPT-3.5, faces challenges when dealing with exclusive gateways and following the correct order of tasks, presenting tasks that are not yet available in the process execution. Just like for GPT-3.5, EQ6, EQ7 and EQ10, which were skipped due to lack of parallel gateways in this use case. We will now move on to the second use case, which is the Wedding Planning process model.

5.4.2 Use Case 2: Wedding Planning

We now move on to the second process model used in Chapter 3, the Wedding Planning model, by sending its BPMN XML file to PaLM. The XML document referring to the Wedding Planning process model depicted in Fig. 3.4 was the same as the one used for GPT, and it can be seen in Appendix A.2.

Category 1: Start Event

After creating a new process instance named X1 and asking for the available tasks, PaLM starts the process instance and lists the single available task, as shown by the conversation below:

User: Let's start a new process instance and name it X1. What tasks are available to me at the beginning?

Chatbot: The following tasks are available to you at the beginning of the process instance X1:

- Choose date

Therefore, EQ1 (Has the process instance been initiated successfully after triggering the start event?) and EQ2 (Does the chatbot correctly mention the first task that follows the start event after the process instance is initiated?) are met.

Category 2: Forward Flow

After choosing the date, the process flow reaches a parallel gateway, and the following four tasks can be executed in any order.

User: I've chosen the date

Chatbot: The following tasks are available to you now:

- Book venue
- Book band
- Book photographer
- Book caterer

According to this interaction, EQ6 (After completing a task and encountering a diverging parallel gateway, does the chatbot correctly present all outgoing flows to the user?) is also met.

The user then proceeds to execute each task:

User: I've booked the photographer

Chatbot: The following tasks are available to you now:

- Book venue
- Book band
- Book caterer

User: I've booked the band

Chatbot: The following tasks are available to you now:

- Book venue
- Book caterer

User: I've booked the caterer

Chatbot: The following tasks are available to you now:

- Book venue

User: I've booked the venue

Chatbot: All tasks have been completed. The process instance X1 has been completed.

As seen in the conversation excerpt above, only the available tasks are mentioned. Therefore, EQ4 (After completing a task, does the chatbot avoid mentioning tasks that are not yet available in the process flow?) is adequately met. Furthermore, the process only proceeds to the end after all tasks have been executed. For that reason, EQ7 (When reaching a converging parallel gateway, does the chatbot advance only when all parallel paths have been completed?) is also met.

Just like in the GPT section, EQ3 (After completing a task, if it is followed by another task directly connected by a direct sequence flow, does the chatbot correctly mention the next task?) is not applicable because there are no tasks directly connected by a sequence flow to another task. Additionally, EQ5 (After completing a task and encountering an exclusive gateway, does the chatbot correctly evaluate the gateway based on the conditions

of its outgoing sequence flows and present only one valid flow to the user?) is also not applicable, because it relates to exclusive gateways, which are not part of this process model.

Category 3: Previous History

For EQ10, (When presenting a task to the user, if it was preceded by a parallel gateway, does the chatbot acknowledge the existence of multiple execution flows?), after booking the photographer and seeing the chatbot mention the other 3 available tasks, we ask if those tasks are all part of the same flow or multiple flows, and the chatbot correctly explains that they are part of multiple flows. Therefore, EQ10 is met, as exemplified by the excerpt below:

User: Are all these tasks from a single flow or multiple flows?

Chatbot: These tasks belong to different parallel flows.

For this use case, EQ8 (When presenting a task to the user, if it was immediately preceded by another task through a sequence flow, does the chatbot correctly identify the preceding task?) and EQ9 (When presenting a task to the user, if it was preceded by an exclusive gateway, does the chatbot accurately convey the decision represented by the preceding gateway?) are not applicable. EQ8 is skipped because there are no tasks directly connected to other tasks without a gateway in between, and EQ9 is skipped because there are no exclusive gateways in this process.

Category 4: End Event

Then, after all the required tasks are executed, the chatbot appropriately finish the process instance:

User: I've booked the venue

Chatbot: All tasks have been completed. The process instance X1 has been completed.

Subsequently, after asking the two questions that refer to EQ11 and EQ12, both are answered accordingly.

User: What was the last task before the process instance was ended?

Chatbot: The last task before the process instance was ended was 'Book venue'.

User: Had all the start events been triggered, and had all pending flows been finished?

Chatbot: Yes, all the start events had been triggered, and all pending flows had been finished.

Hence, EQ11 (Upon reaching an end event, does the chatbot accurately identify the task that immediately preceded it in the process flow?) and EQ12 (When reaching an end event, does the chatbot ensure that the process instance is terminated only when all start events have been triggered and all pending flows have been completed?) are met.

Category 5: Process Decision Points

For the Wedding Planning use case, EQ13 (At the end of the process, does the chatbot retain knowledge of all the decisions that had to be made at gateways?) and EQ14 (At the end of the process, does the chatbot retain knowledge of the user's choice at each gateway?) are both skipped because of the lack of gateways that require decisions. A parallel gateway means that every outgoing flow should be presented to the user, with no decision being involved, unlike exclusive gateways where each outgoing flow has a specific condition and only one flow should be presented. Therefore, there are no decision points in this process.

Category 6: Unintended Paths

If the user starts a new process instance, and tries to execute tasks in the wrong order, for this particular use case, just like GPT, PaLM is successfully able to prevent the wrongful execution:

User: Let's start a new process instance X2, of the same process. What tasks are available to me?

Chatbot: The following tasks are available to you at the beginning of the process instance X2:

- Choose date

User: Can I book the band first?

Chatbot: No, you cannot book the band first. The task 'Choose date' must be completed first.

User: Can I book the venue first?

Chatbot: No, you cannot book the venue first. The task 'Choose date' must be completed first.

Therefore, EQ15 (Does the chatbot allow the user to perform a task that is not yet available in the process?) is met. Additionally, the chatbot does not allow the user to prematurely end the process, which means EQ16 (Does the chatbot allow the user to end the process when it is not supposed to?) is also met, as shown in the excerpt below:

User: I've booked the band

Chatbot: The following tasks are available to you now:

- Book venue
- Book photographer
- Book caterer

User: Can I end the process now?

Chatbot: No, you cannot end the process now. There are still pending tasks that need to be completed first.

Overall, for this more flexible scenario of a Wedding Planning process, PaLM 2 meets all the criteria used in our evaluation, except for the ones that were skipped due to the lack of directly connected tasks and exclusive gateways (EQ3, EQ5, EQ8, EQ9, EQ13, EQ14). That suggests that, just like GPT-3.5, while PaLM 2 might face challenges dealing with exclusive gateways and strict dependencies between tasks, it works great for following BPMN requirements when presented with less rigid dependencies, such as multiple flows coming out of a parallel gateway.

5.5 Use Case Comparison

Over the previous sections, we have seen examples of conversations with both GPT-3.5 and PaLM 2 related to two process models (Trip Planning and Wedding Planning) that

had been previously implemented in retrieval-based chatbots. Our goal was to evaluate if generative-based chatbots can help users in the process execution in a similar way as retrieval-based chatbots, without the requirement of having an explicit connection to a process engine. The reason why we wanted to experiment with generative-based chatbots for process execution support was twofold:

1. Implementing retrieval-based chatbots from scratch for process execution support takes much more developmental effort and time than using a ready-to-use generative-based chatbot that has already been trained on a large dataset. Therefore, if it was possible to use an already deployed generative-based chatbot achieving the same end result and with much less implementation effort than a retrieval-based chatbot would take, that would be the ideal scenario.
2. Because generative-based chatbots are trained on millions of data points, they are more equipped to deal with exceptions in the process execution and different ways to phrase the same intents than retrieval-based chatbots, which are usually only trained on a small dataset and whose knowledge is limited to that specific context.

However, because of the flexibility introduced by generative-based chatbots, a pivotal question was whether they would be able to respect BPMN constructs, such as the strict order between tasks and the multiple flows coming out of exclusive or parallel gateways.

To analyze how well these chatbots could adhere to these requirements, we devised a set of evaluation questions, detailed in Section 5.2, to conduct a structured evaluation on them. Over the previous sections, we sent each chatbot virtually the same prompts, and observe how each of them would respond. Occasionally, we would also add an extra prompt to better understand a certain response and how that would affect whether a specific evaluation criterion was met or not.

In the current section, we provide a qualitative and quantitative comparison of each chatbot’s performance based on the results from the previous experiments.

5.5.1 Qualitative Comparison

For our qualitative comparison, in Table 5.1, we provide a list of all evaluation criteria (evaluation questions) and whether each of them was fully met, partially met, or not met, for each of our four experiments (Trip Planning with GPT, Wedding Planning with GPT, Trip Planning with PaLM, and Wedding Planning with PaLM).

Table 5.1: Fulfillment of each evaluation question (EQ) in each of the four test cases

EQs	Trip Planning with GPT	Wedding Planning with GPT	Trip Planning with PaLM	Wedding Planning with PaLM
EQ1	Yes	Yes	Yes	Yes
EQ2	Yes	Yes	Yes	Yes
EQ3	Yes	N/A	Yes	N/A
EQ4	Yes	Yes	No	Yes
EQ5	Partially	N/A	Partially	N/A
EQ6	N/A	Yes	N/A	Yes
EQ7	N/A	Yes	N/A	Yes
EQ8	Yes	N/A	Yes	N/A
EQ9	Yes	N/A	No	N/A
EQ10	N/A	Yes	N/A	Yes
EQ11	Yes	Yes	Yes	Yes
EQ12	Yes	Yes	Yes	Yes
EQ13	Yes	N/A	Yes	N/A
EQ14	Yes	N/A	Yes	N/A
EQ15	Yes	Yes	No	Yes
EQ16	Yes	Yes	Yes	Yes

5.5.2 Quantitative Comparison

Following our qualitative comparison, we perform a quantitative comparison based on the results from our test cases. Table 5.2 provides the numerical data for the quantitative analysis of the fulfillment of EQs, broken down into the percentage of EQs that a certain experiment has met, partially met and not met. In this case, the EQs that are not applicable (N/A) to a certain experiment are disregarded in the calculation. For instance, if out of the 16 EQs, 3 are not applicable, one is partially met, and all the 12 other ones are met, the “met” percentage calculation will be $12 / 13 = 92.31\%$.

As can be verified on Table 5.2, all of the applicable BPMN constructs were met both by GPT and PaLM for the more flexible scenario – the “Wedding Planning” use case. However, for the stricter “Trip Planning” use case, both chatbots encountered more challenges. While both had trouble with task dependency and the evaluation of exclusive gateways, GPT achieved overall better scores in terms of meeting BPMN requirements when helping users execute business processes.

Table 5.2: Percentage of fulfillment of EQs by each chatbot (GPT and PaLM) for each test case (Trip Planning and Wedding Planning)

Experiments	GPT			PaLM		
	Met	Partially met	Not met	Met	Partially met	Not met
Trip Planning	92.31	7.69	0	69.23	7.69	23.08
Wedding Planning	100	0	0	100	0	0

5.5.3 Evaluation Conclusions and Limitations

When it comes to assisting users in business process execution based on BPMN constructs, it is essential to consider the performance of both GPT and PaLM. One noteworthy observation is the difference in their ability to analyze exclusive gateways. Exclusive gateways in BPMN models represent pivotal decision points, where a specific path is chosen based on certain conditions. While neither chatbot excels in dealing with this type of gateway, GPT demonstrates a relatively better capability compared to PaLM. In the Trip Planning scenario, which is our process model that includes exclusive gateways, GPT achieved a score of 92.31% in terms of meeting the evaluation questions, while PaLM only met 69.23% of the questions. Both chatbots exhibit limitations in comprehending these intricate conditions,

but GPT’s marginally improved performance suggests a more nuanced understanding of the decision-making process within exclusive gateways. Nevertheless, this area remains a challenge for both chatbots, and future work may focus on enhancing their abilities to interpret complex conditional logic.

In contrast, for BPMN models with less rigid dependencies, such as those involving parallel gateways, both chatbots demonstrate a notably stronger performance. These processes often involve multiple available flows, and GPT and PaLM successfully manage those cases. The Wedding Planning process, which includes a parallel gateway instead of an exclusive gateway, was the use case that was responsible for representing less rigid dependencies. In this scenario, both GPT and PaLM achieved a score of 100% in meeting the evaluation questions. The flexibility that is introduced by parallel gateways aligns well with the chatbots’ generative capabilities, allowing them to provide support for tasks with multiple available execution paths. In these situations, generative-based chatbots prove to be highly adaptable and effective, suggesting their potential to provide optimal assistance in processes that embrace a more dynamic workflow structure.

Regarding chatbot behaviour, a distinction that can be noticed is that, in comparison with the conversations with GPT shown in Section 5.3.1 and 5.3.2, the conversation with PaLM is more straight-to-the-point, while GPT is more conversational. This feature of GPT being more talkative can be good by making users more accepting of its help when being guided through business processes. However, it can also make users confused by the excess of information. Therefore, given a scenario in which both GPT and PaLM perform similarly well, such as the case of less rigid dependencies, the choice between one or the other can be based on the specific conversation style that its prospective users prefer.

While this study provides valuable insights into the integration of process knowledge into generative-based chatbots, it is essential to acknowledge its limitations. First, the evaluation is restricted to two specific process model use cases, namely Trip Planning and Wedding Planning. To broaden the applicability of our findings, it would be interesting to assess different process models, such as manufacturing processes or financial processes, which might present distinct challenges and requirements. Second, our research focused on a particular subset of BPMN elements, specifically events, tasks, and two types of gateways. While these constructs are relevant, the full spectrum of BPMN features, including different kinds of gateways (e.g., inclusive and event-based), messages, timers, and subprocesses, remains unexplored. A more comprehensive analysis of BPMN constructs would provide a deeper understanding of how well generative-based chatbots can meet the requirements of these other constructs. Finally, our study is limited to a specific subset of generative-based models, primarily GPT and PaLM. Although these models are currently prominent in the industry, there are other models such as Meta’s OPT and NVIDIA’s Megatron-Turing

that could be explored. A more extensive evaluation of generative-based models could yield insights into which models are better for specific scenarios.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Business processes are integral to organizational operations, providing a structured framework for tasks from various domains and industries. The universal notation of BPMN has become the standard for representing these processes, allowing diverse businesses to describe their operations within a common format. While process automation has become prevalent in the industry for its potential to enhance precision and efficiency, human workflows introduce unique complexities. These workflows involve dynamic task orders, intricate task dependencies, and the necessity for contextual adaptability. Often, participants in these processes have partial knowledge and require guidance to navigate and execute tasks effectively.

The introduction of chatbots a few years ago marked an advancement in technology, allowing users to interact with computer programs using natural language. However, many chatbots were limited in their understanding of context and were only able to provide restricted assistance. In recent years, the emergence of LLMs has transformed human-machine interactions, enabling more fluid conversations. Yet, LLMs have limitations, particularly their broad knowledge that may lack the specific context required for task execution in particular business processes.

This thesis focused on the development of chatbots tailored to meet business process requirements, under the guidance of BPMN constructs. Through three distinctive phases, it uncovered gaps and challenges in existing literature, showcased two approaches for making generative-based chatbots process-aware, and conducted comparative evaluations of two

prominent generative models, GPT and PaLM, on their ability to support business process execution.

In the context of business process execution, both GPT and PaLM exhibit their strengths and weaknesses. Exclusive gateways present a challenge for both, with GPT demonstrating a slightly better understanding of decision-making conditions. On the other hand, looser dependencies, such as those involving parallel gateways, revealed the adaptability of generative-based chatbots, demonstrating their potential for supporting parallelized workflow structures. An intriguing behavioural difference emerged, where GPT's talkative style contrasted with PaLM's straightforward messages. This observation highlights the importance of considering user preferences in terms of conversational style when selecting a chatbot for business process execution guidance.

This research highlights the potential of chatbots as valuable tools in business process execution, offering guidance on how to use generative models for more effective support. As organizations continue to seek efficient solutions for their processes, this work provides a substantial step toward a better integration of chatbots in the domain of business process execution support.

6.2 Future Work

In this section, we explore potential future research directions that can advance the usage of generative-based chatbots as tools that understand BPMN context and that offer support for BPMN process modelling and execution.

6.2.1 Possible Integration with Camunda Engine

Despite the fact that, in the current work, we decide not to integrate a process engine to generative-based chatbots due to the lack of available plugins for this connection and the desire to understand if these chatbots can understand BPMN concepts by themselves, there is an opportunity to connect them using an external service application that would act as a bridge. Therefore, a promising avenue for future work involves developing a new application that leverages the available APIs of GPT and PaLM and integrate them with Camunda Engine. This application could dynamically interact with generative-based chatbots to send multiple messages during process execution and share the execution status from the process engine with the chatbot. Instead of dispatching a single comprehensive message at the start with the entire BPMN XML document and letting the chatbot keep

track of the task status by itself, this bridge application would send frequent messages to the chatbot based on the real-time status of individual tasks. This approach could enhance the adaptability and responsiveness of chatbots in supporting process execution, ensuring that the information provided remains relevant and timely.

6.2.2 Evaluation Criteria

Another avenue for future research would be extending the evaluation criteria to assess the applicability of generative-based process-aware chatbots for different process notations. While the current criteria have been designed to evaluate generative-based chatbot performance for BPMN processes, investigating their suitability for other process models can provide valuable insights into the generalizability of these criteria.

Also regarding the evaluation criteria, future work may explore syntactic evaluation criteria, to complement the semantic evaluation that was performed in the current work. This expansion would involve assessing not only how well generative-based chatbots comprehend the meaning of BPMN XML documents, but also their ability to handle various syntax and structural aspects. This dual evaluation approach could offer a more comprehensive assessment of chatbot performance.

Additionally, a valuable extension to the study would be to conduct new quantitative experiments based on the refined evaluation criteria. This approach would allow for a deeper understanding of the chatbots' performance and their ability to meet the evolving demands of BPMN process execution support. It would also be able to help in fine-tuning the evaluation criteria for increased adherence to BPMN constructs.

6.2.3 Process Modelling

In addition to using process-aware chatbots for helping in the execution of process models, a novel research direction could involve investigating the use of generative-based chatbots to assist in the initial modeling of BPMN processes. These chatbots could take a list of tasks, decisions, events, and other BPMN constructs detailed in a natural language prompt, and generate the final BPMN XML documents. This approach could facilitate the process modeling phase and simplify the work of process designers, potentially enhancing the overall efficiency of BPMN process modeling.

6.2.4 Additional BPMN Constructs

Future work could explore the use of additional BPMN constructs that were not covered in the current work. In this thesis, we utilized events, tasks and two types of gateways (parallel and exclusive). Additional constructs can be added to the BPMN models, including other types of gateways, such as inclusive and complex gateways, and artifacts, and the chatbot's adherence to the BPMN requirements can be assessed.

6.2.5 Utilized Chatbot Models

Future papers could also investigate the application of alternative generative chatbot models beyond GPT and PaLM. Analyzing how different models, such as Meta's OPT and NVIDIA's Megatron-Turing, perform in the context of BPMN process understanding and process execution support is a promising avenue for future work.

Lastly, researchers can focus on the fine-grained adjustment of these chatbot models' parameters to optimize process adherence. By systematically tweaking the parameters, researchers can discover specific settings that improve or worsen the chatbots' ability to effectively support BPMN process execution.

Overall, these proposed directions for future work can further contribute to the development and application of generative-based chatbots in the context of BPMN processes, ultimately advancing the field of process automation and support.

References

- [1] Lissette Almonte, Sara Pérez-Soler, Esther Guerra, Iván Cantador, and Juan de Lara. Automating the synthesis of recommender systems for modelling languages. In *Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering*, pages 22–35, 2021.
- [2] Ilaria Angela Amantea, Emilio Sulis, Guido Boella, Fabiola De Marchi, Letizia Mazzini, Francesca Alloatti, and Andrea Bolioli. Adopting assistive technologies in health-care processes: a chatbot for patients with amyotrophic lateral sclerosis. In *Italian Forum of Ambient Assisted Living*, pages 163–174. Springer, 2020.
- [3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.
- [4] Rubén Baena-Perez, Iván Ruiz-Rube, José Miguel Mota, Anke Berns, and Antonio Balderas. Visual authoring of virtual reality conversational scenarios for e-learning. *Universal Access in the Information Society*, pages 1–18, 2022.
- [5] Sadra Bakhshandeh. Benchmarking medical large language models. *Nature Reviews Bioengineering*, pages 1–1, 2023.
- [6] Daniel Barón-Espitia, Marlon Dumas, and Oscar González-Rojas. Coral: Conversational what-if process analysis. 2022.
- [7] Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. Understanding the long-term use of smart speaker assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–24, 2018.

- [8] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*, 2017.
- [9] Nick Bradley, Thomas Fritz, and Reid Holmes. Context-aware conversational developer assistants. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 993–1003. IEEE, 2018.
- [10] Xusen Cheng, Ying Bao, Alex Zarifis, Wankun Gong, and Jian Mou. Exploring consumers’ response to text-based chatbots in e-commerce: the moderating role of task complexity and chatbot disclosure. *Internet Research*, 2021.
- [11] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, et al. What makes a good conversation? challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
- [12] Donald R Coughanowr, Lowell B Koppel, et al. *Process systems analysis and control*, volume 2. McGraw-Hill New York, 1965.
- [13] Robert Dale. Gpt-3: What’s it good for? *Natural Language Engineering*, 27(1):113–118, 2021.
- [14] Jiangbo Dang, Candemir Toklu, Ken Hampel, and Urs Enke. Human workflows via document-driven process choreography. In *2008 International MCETECH Conference on e-Technologies (mcetech 2008)*, pages 25–33. IEEE, 2008.
- [15] Nataliia Danylevych, Svetlana Rudakova, Liudmyla Shchetinina, and Oksana Poplavska. Digitalization of personnel management processes: Reserves for using chatbots. In *IntSol Workshops*, pages 166–176, 2021.
- [16] Remco Dijkman, Marcello La Rosa, and Hajo Reijers. Managing large collections of business process models-current techniques and challenges. *Computers in Industry*, 63(2):91–97, 2012.
- [17] Marlon Dumas. *Fundamentals of business process management*. Springer, Berlin, Germany, 2018.
- [18] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S Bernstein. Iris: A conversational agent for complex tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

- [19] Diogo Ferreira, Filipe Portela, and Manuel Filipe Santos. A step towards the use of chatbots to support the enterprise decision-making processes. In *Trends and Applications in Information Systems and Technologies: Volume 4 9*, pages 308–317. Springer, 2021.
- [20] Albert Fleischmann. What is s-bpm? In *S-BPM ONE—Setting the Stage for Subject-Oriented Business Process Management: First International Workshop, Karlsruhe, Germany, October 22, 2009. Revised Selected Papers 1*, pages 85–106. Springer, 2010.
- [21] Marius Gassen, Benjamin Hättasch, Benjamin Hilprecht, Nadja Geisler, Alexander Fraser, and Carsten Binnig. Demonstrating cat: synthesizing data-aware conversational agents for transactional databases. *arXiv preprint arXiv:2203.14144*, 2022.
- [22] Ulrich Gnewuch, Stefan Morana, and Alexander Maedche. Towards designing cooperative and social conversational agents for customer service. In *ICIS*, 2017.
- [23] Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, pages 946–956. Springer, 2019.
- [24] Christoph Kecht, Andreas Egger, Wolfgang Kratsch, and Maximilian Röglinger. Quantifying chatbots’ ability to learn business processes. *Information Systems*, 113:102176, 2023.
- [25] Dorota Košeccka, Peter Balco, and Sharon Cheronno Murgor. Chatbot at university, a communication tool to increase work productivity. In *International Conference on Intelligent Networking and Collaborative Systems*, pages 74–84. Springer, 2022.
- [26] Marco Kuhrmann, Daniel Méndez Fernández, and Maya Daneva. On the pragmatic design of literature studies in software engineering: an experience-based guideline. *Empirical software engineering*, 22:2852–2891, 2017.
- [27] Victor Kurtz, Jakob Bönsch, and Jivka Ovtcharova. Understanding and harnessing the potential of conversational ai for s-bpm. In *International Conference on Subject-Oriented Business Process Management*, pages 41–57. Springer, 2022.
- [28] Cheongjae Lee, Sangkeun Jung, Seokhwan Kim, and Gary Geunbae Lee. Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5):466–484, 2009.

- [29] Luis Fernando Lins, Glaucia Melo, Toacy Oliveira, Paulo Alencar, and Donald Cowan. Pacas: process-aware conversational agents. In *International Conference on Business Process Management*, pages 312–318. Springer, 2021.
- [30] Brady D Lund and Ting Wang. Chatting about chatgpt: how may ai and gpt impact academia and libraries? *Library Hi Tech News*, 40(3):26–29, 2023.
- [31] Bei Luo, Raymond YK Lau, Chunping Li, and Yain-Whar Si. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1):e1434, 2022.
- [32] Alina A Menshchikova and Anna E Stepanova. Integration of icq new into business processes implemented on the basis of the 1c: Enterprise. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 541–544. IEEE, 2021.
- [33] Mariya Nazarkevych, Pavlo Levush, and Bohdan Pankovych. Improving the efficiency of information collection based on the development of a chatbot with a parser. In *2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT)*, pages 125–129. IEEE, 2021.
- [34] Juliane Neumann, Max Rockstroh, Sascha Vinz, and Thomas Neumuth. Surgical workflow and process modeling. In *M2CAI Workshop, Medical Image Computing and Computer-Assisted Intervention–MICCAI*, volume 2015, 2015.
- [35] M Object Management Group. Decision model and notation (dmn), v1. 1, 2016.
- [36] OMG. Business Process Model and Notation (BPMN), Version 2.0.2, January 2014.
- [37] Servio Palacios, Drew Zabrocki, Bharat Bhargava, and Vaneet Aggarwal. Auditable serverless computing for farm management. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments*, pages 1–6, 2021.
- [38] Stephan Raaijmakers, Anita Cremers, Emiel Krahmer, and Matthijs Westera. Conversational ai. *Frontiers in Artificial Intelligence*, 6:1203910, 2023.
- [39] Nicole M. Radziwill and Morgan C. Benton. Evaluating quality of chatbots and intelligent conversational agents, 2017.
- [40] Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K Chandrasekaran. A survey of design techniques for conversational agents. In *Information, Communication*

and Computing Technology: Second International Conference, ICICCT 2017, New Delhi, India, May 13, 2017, Revised Selected Papers, pages 336–350. Springer, 2017.

- [41] Miguel Ángel Rodríguez-García, Camilo Caparrós-Laiz, Pedro José Vivancos-Vicente, José Antonio García-Díaz, and Rafael Valencia-García. Ivrmaker, an interactive and customizable telephone chatbot services platform. In *International Conference on Technologies and Innovation*, pages 62–74. Springer, 2022.
- [42] Donya Rooein, Devis Bianchini, Francesco Leotta, Massimo Mecella, Paolo Paolini, and Barbara Pernici. achat-wf: Generating conversational agents for teaching business process models. *Software and Systems Modeling*, pages 1–24, 2022.
- [43] Faizel Sedick and Lisa F Seymour. Bpmn usage: An analysis of influencing factors. *Advances in Enterprise Information Systems II*, 155, 2012.
- [44] Michael Shumanov and Lester Johnson. Making conversations with chatbots more personalized. *Computers in Human Behavior*, 117:106627, 2021.
- [45] Pascal Sibold. Modelling language for domain-specific decisions in healthcare. 2017.
- [46] Gagandeep Singh. A move towards intelligent economy: Indian evidence. *Management and Labour Studies*, 46(2):192–203, 2021.
- [47] Raphael Meyer von Wolff, Sebastian Hobert, and Matthias Schumann. Designing process-based chatbots in enterprises: The case of business travel organization considering the users’ perspective and business value. *AIS Transactions on Human-Computer Interaction*, 14(4):578–623, 2022.
- [48] Weiyu Wang and Keng Siau. Living with artificial intelligence—developing a theory on trust in health chatbots. In *Proceedings of the Sixteenth Annual Pre-ICIS Workshop on HCI Research in MIS*, pages 1–5. Association for Information Systems San Francisco, CA, 2018.
- [49] Mathias Weske. *Business process management : concepts, languages, architectures*. Springer, Berlin New York, 2012.
- [50] Markus Wiemuth, Denise Junger, MA Leitritz, Juliane Neumann, Thomas Neumuth, and Oliver Burgert. Application fields for the new object management group (omg) standards case management model and notation (cmmn) and decision management notation (dmn) in the perioperative field. *International journal of computer assisted radiology and surgery*, 12:1439–1449, 2017.

- [51] Sergey Zeltyn, Segev Shlomov, Avi Yaeli, and Alon Oved. Prescriptive process monitoring in intelligent process automation with chatbot orchestration. *arXiv preprint arXiv:2212.06564*, 2022.
- [52] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.
- [53] Hang Zou, Qiyang Zhao, Lina Bariah, Mehdi Bennis, and Merouane Debbah. Wireless multi-agent generative ai: From connected intelligence to collective intelligence. *arXiv preprint arXiv:2307.02757*, 2023.

APPENDICES

Appendix A

BPMN XML documents

A.1 Trip Planning BPMN XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
  xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:dc="http
  ://www.omg.org/spec/DD/20100524/DC" xmlns:di="http://www.omg.org/spec/
  DD/20100524/DI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="Definitions_0vy617o" targetNamespace="http://bpmn.io/schema/bpmn"
  exporter="Camunda Modeler" exporterVersion="5.4.2">
  <bpmn:process id="simple_trip_planning_optional" name="Simple Trip
    Planning" isExecutable="true">
    <bpmn:startEvent id="StartEvent_1">
      <bpmn:outgoing>SequenceFlow_14ibw8p</bpmn:outgoing>
    </bpmn:startEvent>
    <bpmn:userTask id="book_flight" name="Book flight">
      <bpmn:extensionElements />
      <bpmn:incoming>SequenceFlow_14ibw8p</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_1fvc9ns</bpmn:outgoing>
    </bpmn:userTask>
    <bpmn:userTask id="book_transfer" name="Book transfer">
      <bpmn:incoming>Flow_0dlt82</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_06safu6</bpmn:outgoing>
    </bpmn:userTask>
```

```

<bpmn:userTask id="book_tour" name="Book tour">
  <bpmn:incoming>SequenceFlow_06safu6</bpmn:incoming>
  <bpmn:incoming>Flow_05tn7gn</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_0an5hs1</bpmn:outgoing>
</bpmn:userTask>
<bpmn:endEvent id="EndEvent_1pckwqd">
  <bpmn:incoming>SequenceFlow_0an5hs1</bpmn:incoming>
</bpmn:endEvent>
<bpmn:sequenceFlow id="SequenceFlow_0an5hs1" sourceRef="book_tour"
  targetRef="EndEvent_1pckwqd" />
<bpmn:sequenceFlow id="SequenceFlow_14ibw8p" sourceRef="StartEvent_1"
  targetRef="book_flight" />
<bpmn:sequenceFlow id="SequenceFlow_1fvc9ns" sourceRef="book_flight"
  targetRef="book_hotel" />
<bpmn:sequenceFlow id="SequenceFlow_0h9wybb" sourceRef="book_hotel"
  targetRef="Gateway_01f6asu" />
<bpmn:sequenceFlow id="SequenceFlow_06safu6" sourceRef="book_transfer"
  targetRef="book_tour" />
<bpmn:userTask id="book_hotel" name="Book hotel">
  <bpmn:incoming>SequenceFlow_1fvc9ns</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_0h9wybb</bpmn:outgoing>
</bpmn:userTask>
<bpmn:exclusiveGateway id="Gateway_01f6asu" name="Transfer preference">
  <bpmn:incoming>SequenceFlow_0h9wybb</bpmn:incoming>
  <bpmn:outgoing>Flow_0dltd82</bpmn:outgoing>
  <bpmn:outgoing>Flow_05tn7gn</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:sequenceFlow id="Flow_0dltd82" sourceRef="Gateway_01f6asu"
  targetRef="book_transfer">
  <bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">If user
  wants to book a transfer</bpmn:conditionExpression>
</bpmn:sequenceFlow>
<bpmn:sequenceFlow id="Flow_05tn7gn" sourceRef="Gateway_01f6asu"
  targetRef="book_tour">
  <bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">If user
  does not want to book a transfer</bpmn:conditionExpression>
</bpmn:sequenceFlow>
</bpmn:process>

```

```

<bpmndi:BPMNDiagram id="BPMNDiagram_1">
  <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="
    simple_trip_planning_optional">
    <bpmndi:BPMNShape id="_BPMNShape_StartEvent_2" bpmnElement="
      StartEvent_1">
      <dc:Bounds x="182" y="169" width="36" height="36" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="UserTask_0t4vt3b_di" bpmnElement="book_flight">
      <dc:Bounds x="290" y="147" width="100" height="80" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="UserTask_0v3cm6c_di" bpmnElement="book_transfer
      ">
      <dc:Bounds x="760" y="147" width="100" height="80" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="UserTask_1c3alj9_di" bpmnElement="book_tour">
      <dc:Bounds x="990" y="147" width="100" height="80" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="EndEvent_1pckwqd_di" bpmnElement="
      EndEvent_1pckwqd">
      <dc:Bounds x="1152" y="169" width="36" height="36" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="UserTask_0y3500g_di" bpmnElement="book_hotel">
      <dc:Bounds x="520" y="147" width="100" height="80" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="Gateway_01f6asu_di" bpmnElement="
      Gateway_01f6asu" isMarkerVisible="true">
      <dc:Bounds x="665" y="162" width="50" height="50" />
      <bpmndi:BPMNLabel>
        <dc:Bounds x="665" y="219" width="53" height="27" />
      </bpmndi:BPMNLabel>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNEdge id="SequenceFlow_0an5hs1_di" bpmnElement="
      SequenceFlow_0an5hs1">
      <di:waypoint x="1090" y="187" />
      <di:waypoint x="1152" y="187" />
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge id="SequenceFlow_14ibw8p_di" bpmnElement="
      SequenceFlow_14ibw8p">

```

```

    <di:waypoint x="218" y="187" />
    <di:waypoint x="290" y="187" />
  </bpmndi:BPMNEdge>
  <bpmndi:BPMNEdge id="SequenceFlow_1fvc9ns_di" bpmnElement="
    SequenceFlow_1fvc9ns">
    <di:waypoint x="390" y="187" />
    <di:waypoint x="520" y="187" />
  </bpmndi:BPMNEdge>
  <bpmndi:BPMNEdge id="SequenceFlow_0h9wybb_di" bpmnElement="
    SequenceFlow_0h9wybb">
    <di:waypoint x="620" y="187" />
    <di:waypoint x="665" y="187" />
  </bpmndi:BPMNEdge>
  <bpmndi:BPMNEdge id="SequenceFlow_06safu6_di" bpmnElement="
    SequenceFlow_06safu6">
    <di:waypoint x="860" y="187" />
    <di:waypoint x="990" y="187" />
  </bpmndi:BPMNEdge>
  <bpmndi:BPMNEdge id="Flow_0dltd82_di" bpmnElement="Flow_0dltd82">
    <di:waypoint x="715" y="187" />
    <di:waypoint x="760" y="187" />
  </bpmndi:BPMNEdge>
  <bpmndi:BPMNEdge id="Flow_05tn7gn_di" bpmnElement="Flow_05tn7gn">
    <di:waypoint x="690" y="162" />
    <di:waypoint x="690" y="80" />
    <di:waypoint x="950" y="80" />
    <di:waypoint x="950" y="187" />
    <di:waypoint x="990" y="187" />
  </bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</bpmn:definitions>

```

A.2 Wedding Planning BPMN XML document

```

<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
  xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:dc="http
://www.omg.org/spec/DD/20100524/DC" xmlns:di="http://www.omg.org/spec/
DD/20100524/DI" xmlns:modeler="http://camunda.org/schema/modeler/1.0"
id="Definitions_1m1lomf" targetNamespace="http://bpmn.io/schema/bpmn"
exporter="Camunda Modeler" exporterVersion="5.4.2" modeler:
executionPlatform="Camunda Platform" modeler:executionPlatformVersion="
7.18.0">
<bpmn:process id="Process_0pju9cz" isExecutable="true">
  <bpmn:startEvent id="StartEvent_1">
    <bpmn:outgoing>Flow_00gl4my</bpmn:outgoing>
  </bpmn:startEvent>
  <bpmn:userTask id="Activity_1iit2w9" name="Choose date">
    <bpmn:incoming>Flow_00gl4my</bpmn:incoming>
    <bpmn:outgoing>Flow_12873n6</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:sequenceFlow id="Flow_00gl4my" sourceRef="StartEvent_1" targetRef
    ="Activity_1iit2w9" />
  <bpmn:sequenceFlow id="Flow_12873n6" sourceRef="Activity_1iit2w9"
    targetRef="Gateway_0m3be6e" />
  <bpmn:parallelGateway id="Gateway_0m3be6e">
    <bpmn:incoming>Flow_12873n6</bpmn:incoming>
    <bpmn:outgoing>Flow_0r3d2hv</bpmn:outgoing>
    <bpmn:outgoing>Flow_1nctfp0</bpmn:outgoing>
    <bpmn:outgoing>Flow_04nfefo</bpmn:outgoing>
    <bpmn:outgoing>Flow_1uyxfgp</bpmn:outgoing>
  </bpmn:parallelGateway>
  <bpmn:sequenceFlow id="Flow_0r3d2hv" sourceRef="Gateway_0m3be6e"
    targetRef="Activity_0mqww1k" />
  <bpmn:userTask id="Activity_0mqww1k" name="Book band">
    <bpmn:incoming>Flow_0r3d2hv</bpmn:incoming>
    <bpmn:outgoing>Flow_1nv0nve</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:sequenceFlow id="Flow_1nctfp0" sourceRef="Gateway_0m3be6e"
    targetRef="Activity_0hzejli" />
  <bpmn:userTask id="Activity_0hzejli" name="Book photographer">
    <bpmn:incoming>Flow_1nctfp0</bpmn:incoming>

```

```

    <bpmn:outgoing>Flow_1rl01bd</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:sequenceFlow id="Flow_04nfefo" sourceRef="Gateway_0m3be6e"
    targetRef="Activity_0szmooh" />
  <bpmn:userTask id="Activity_0szmooh" name="Book caterer">
    <bpmn:incoming>Flow_04nfefo</bpmn:incoming>
    <bpmn:outgoing>Flow_0odf446</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:sequenceFlow id="Flow_1uyxfgp" sourceRef="Gateway_0m3be6e"
    targetRef="Activity_0lhgvkj" />
  <bpmn:userTask id="Activity_0lhgvkj" name="Book venue">
    <bpmn:incoming>Flow_1uyxfgp</bpmn:incoming>
    <bpmn:outgoing>Flow_0rwl9ej</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:sequenceFlow id="Flow_1nv0nve" sourceRef="Activity_0mqw1k"
    targetRef="Gateway_1ka16zc" />
  <bpmn:parallelGateway id="Gateway_1ka16zc">
    <bpmn:incoming>Flow_1nv0nve</bpmn:incoming>
    <bpmn:incoming>Flow_1rl01bd</bpmn:incoming>
    <bpmn:incoming>Flow_0odf446</bpmn:incoming>
    <bpmn:incoming>Flow_0rwl9ej</bpmn:incoming>
    <bpmn:outgoing>Flow_1vgdj7</bpmn:outgoing>
  </bpmn:parallelGateway>
  <bpmn:sequenceFlow id="Flow_1rl01bd" sourceRef="Activity_0hzejli"
    targetRef="Gateway_1ka16zc" />
  <bpmn:sequenceFlow id="Flow_0odf446" sourceRef="Activity_0szmooh"
    targetRef="Gateway_1ka16zc" />
  <bpmn:sequenceFlow id="Flow_0rwl9ej" sourceRef="Activity_0lhgvkj"
    targetRef="Gateway_1ka16zc" />
  <bpmn:endEvent id="Event_0tey7i4">
    <bpmn:incoming>Flow_1vgdj7</bpmn:incoming>
  </bpmn:endEvent>
  <bpmn:sequenceFlow id="Flow_1vgdj7" sourceRef="Gateway_1ka16zc"
    targetRef="Event_0tey7i4" />
</bpmn:process>
<bpmndi:BPMNDiagram id="BPMNDiagram_1">
  <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="Process_0p9cz">
    <bpmndi:BPMNShape id="_BPMNShape_StartEvent_2" bpmnElement="

```

```

    StartEvent_1">
    <dc:Bounds x="179" y="249" width="36" height="36" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_01gftf5_di" bpmnElement="
    Activity_1iit2w9">
    <dc:Bounds x="270" y="227" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Gateway_1c9m9tb_di" bpmnElement="
    Gateway_0m3be6e">
    <dc:Bounds x="425" y="242" width="50" height="50" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_1ht7hk6_di" bpmnElement="
    Activity_0hzejli">
    <dc:Bounds x="530" y="227" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_1v17v15_di" bpmnElement="
    Activity_0mqww1k">
    <dc:Bounds x="530" y="80" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_0lq8su1_di" bpmnElement="
    Activity_0szmooh">
    <dc:Bounds x="530" y="380" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_100lghm_di" bpmnElement="
    Activity_0lhgvkj">
    <dc:Bounds x="530" y="540" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Gateway_1c6bosi_di" bpmnElement="
    Gateway_1ka16zc">
    <dc:Bounds x="735" y="242" width="50" height="50" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Event_0tey7i4_di" bpmnElement="Event_0tey7i4">
    <dc:Bounds x="892" y="249" width="36" height="36" />
</bpmndi:BPMNShape>
<bpmndi:BPMNEdge id="Flow_00gl4my_di" bpmnElement="Flow_00gl4my">
    <di:waypoint x="215" y="267" />
    <di:waypoint x="270" y="267" />
</bpmndi:BPMNEdge>

```

```

<bpmndi:BPMNEdge id="Flow_12873n6_di" bpmnElement="Flow_12873n6">
  <di:waypoint x="370" y="267" />
  <di:waypoint x="425" y="267" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_0r3d2hv_di" bpmnElement="Flow_0r3d2hv">
  <di:waypoint x="450" y="242" />
  <di:waypoint x="450" y="120" />
  <di:waypoint x="530" y="120" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1nctfp0_di" bpmnElement="Flow_1nctfp0">
  <di:waypoint x="475" y="267" />
  <di:waypoint x="530" y="267" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_04nfefo_di" bpmnElement="Flow_04nfefo">
  <di:waypoint x="450" y="292" />
  <di:waypoint x="450" y="420" />
  <di:waypoint x="530" y="420" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1uyxfgp_di" bpmnElement="Flow_1uyxfgp">
  <di:waypoint x="450" y="292" />
  <di:waypoint x="450" y="580" />
  <di:waypoint x="530" y="580" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1nv0nve_di" bpmnElement="Flow_1nv0nve">
  <di:waypoint x="630" y="120" />
  <di:waypoint x="760" y="120" />
  <di:waypoint x="760" y="242" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1rl01bd_di" bpmnElement="Flow_1rl01bd">
  <di:waypoint x="630" y="267" />
  <di:waypoint x="735" y="267" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_0odf446_di" bpmnElement="Flow_0odf446">
  <di:waypoint x="630" y="420" />
  <di:waypoint x="760" y="420" />
  <di:waypoint x="760" y="292" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_0rwl9ej_di" bpmnElement="Flow_0rwl9ej">

```

```
<di:waypoint x="630" y="580" />
<di:waypoint x="760" y="580" />
<di:waypoint x="760" y="292" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1vgdjq7_di" bpmnElement="Flow_1vgdjq7">
  <di:waypoint x="785" y="267" />
  <di:waypoint x="892" y="267" />
</bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</bpmn:definitions>
```

Appendix B

Literature Review Search Results

Table B.1: Studies retrieved when using our selected search query

ID	Title	Included?	Reason for exclusion
A1	A Comparative Analysis of Automation Anywhere, UiPath, and BluePrism	No	RPA
A2	Robotic Process Automation for Automating Business Processes: A use case	No	RPA
A3	Democratizing Artificial Intelligence with UiPath: Expand automation in your organization to achieve operational efficiency and high performance	No	Book
A4	A Holistic Approach to Ensure Security and Compliance while using Robotic Process Automation	No	No chatbot component
A5	Robotics and Automatics in Terms of Utilizing Rules-based Business Processes	No	RPA
A6	Chatbots Analysis for the Creation of Automated Conversations in Real Time	No	Not in English

A7	Multi-Channel Chatbot and Robotic Process Automation	No	RPA
A8	Modern Development Trends of Chatbots Using Artificial Intelligence (AI)	No	Does not cover business processes in depth
A9	Evaluating AI Chatbot Platforms by a Fuzzy AHP Approach	No	Does not cover business processes in depth
A10	Improving the Efficiency of Information Collection Based on the Development of a Chatbot with a Parser	Yes	
A11	Integration of ICQ New into Business Processes Implemented on the Basis of the 1C: Enterprise	Yes	
A12	An Enterprise View for Artificial Intelligence Capability and Governance: A System Dynamics Approach	No	No chatbot component
A13	Problem Formulation and Use Case Identification of AI in Government: Results from the Literature Review	No	Does not cover business processes in depth
A14	Characterizing Toolkits for Platform Independent Chatbot Development	No	Not in English
A15	Robotic Process Automation in Latin American Organizations: Survey and Evaluation of the Current State of Technology Adoption	No	RPA
A16	Out of Context: Investigating the Bias and Fairness Concerns of “Artificial Intelligence as a Service”	No	No chatbot component
A17	From Plane Crashes to Algorithmic Harm: Applicability of Safety Engineering Frameworks for Responsible ML	No	Does not cover business processes in depth

A18	Charting the Automation of Hospitality: An Interdisciplinary Literature Review Examining the Evolution of Frontline Service Work in the Face of Algorithmic Management	No	No chatbot component
A19	Building Knowledge through Action: Considerations for Machine Learning in the Workplace	No	No chatbot component
A20	A Competency Management Model to Support Digital Transformation Initiatives in a Public Organization	No	Not in English
A21	Integrating BPMN with DMN to Model Complex Public Services: The Case of Getting a Transportation Card for Disabled in Greece	No	No chatbot component
A22	"It's Only When Somebody Says a Tool Worked for Them That I Believe It Will Work for Me": Socio-Tecture as a Lens for Digital Transformation	No	No chatbot component
A23	A Light-Weight Low-Code Platform for Back-End Automation	No	No chatbot component
A24	Data is about Detail: An Empirical Investigation for Software Systems with NLP at Core	No	Does not cover business processes in depth
A25	Quantifying chatbots' ability to learn business processes	Yes	
A26	Digital Transformation of Information Fulfillment and Patient Engagement for Health Service Safety	No	Does not cover business processes in depth
A27	Digital Economy in Tourism Industry	No	Does not cover business processes in depth

A28	Messengers Integration Methods with Corporate Information Systems: The Introduction and Application of Instant Messengers to the Corporate Information System	No	Does not cover business processes in depth
A29	The Implementation of Robotic Process Automation for Banking Sector Case Study of A Private Bank in Indonesia	No	No chatbot component
A30	A Study for the Implementation of Banking 4.0 in Indonesia	No	Does not cover business processes in depth
A31	A Simulation Tool to Provide Alternative Products in Out-of-Stock Situations for B2B Companies	No	No chatbot component
A32	Creating an Inter-Hospital Resilient Network for Pandemic Response Based on Blockchain and Dynamic Digital Twins	No	No chatbot component
A33	Fundamental Evaluation of Digital Companies	No	Does not cover business processes in depth
A34	Key Aspects of the Formation and Development of Russian Business Ecosystems	No	Does not cover business processes in depth
A35	Conceptual Framework for High Performance Digital Entrepreneurial University	No	Does not cover business processes in depth
A36	A Step Towards the Use of Chatbots to Support the Enterprise Decision-Making Processes	Yes	
A37	A Case Study of Phishing Incident Response in an Educational Organization	No	No chatbot component
A38	Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective	No	Does not cover business processes in depth

A39	Risk Identification Questionnaire for Detecting Unintended Bias in the Machine Learning Development Lifecycle	No	Does not cover business processes in depth
A40	Prioritizing the Software Development Methodologies in Online Gig Economy Project Using Analytic Hierarchy Process	No	Does not cover business processes in depth
A41	Visual authoring of virtual reality conversational scenarios for e-learning	Yes	
A42	Digital Ecosystem Development in Customer Relationship Management (CRM) for Hajj Portal Website	No	Does not cover business processes in depth
A43	A Digital Transformation Model of the Staff Selection System	No	No chatbot component
A44	The Impact of Industry 4.0 Technologies on Retail Development	No	No chatbot component
A45	Advantages of Using Omni-Channel Marketing Technologies in the Field of Trade	No	Does not cover business processes in depth
A46	Causal Perception in Question-Answering Systems	No	Does not cover business processes in depth
A47	Prescriptive Process Monitoring in Intelligent Process Automation with Chatbot Orchestration	Yes	
A48	Integrating Chatbot and RPA into Enterprise Applications Based on Open, Flexible and Extensible Platforms	No	RPA
A49	Understanding and Harnessing the Potential of Conversational AI for S-BPM	Yes	
A50	Augmented employee voice and employee engagement using artificial intelligence-enabled chatbots: a field study	No	Does not cover business processes in depth

A51	Digitalization: Implementation in the tourism business of Ukraine	No	Does not cover business processes in depth
A52	Chatbot at University, a Communication Tool to Increase Work Productivity	Yes	
A53	Bridging Research Fields: An Empirical Study on Joint, Neural Relation Extraction Techniques	No	No chatbot component
A54	The prospect of digital human communication for organizational purposes	No	Does not cover business processes in depth
A55	A Move Towards Intelligent Economy: Indian Evidence	Yes	
A56	Digitalization of Personnel Management Processes: Reserves for Using Chatbots	Yes	
A57	Integration of ICQ New into Business Processes Implemented on the Basis of the 1C: Enterprise	Yes	
A58	Improving the Efficiency of Information Collection Based on the Development of a Chatbot with a Parser	Yes	
A59	Automating the Synthesis of Recommender Systems for Modelling Languages	Yes	
A60	Auditable Serverless Computing for Farm Management	Yes	
A61	Demonstrating CAT: Synthesizing Data-Aware Conversational Agents for Transactional Databases	Yes	
A62	PACAs: Process-Aware Conversational Agents	Yes	
A63	Coral: Conversational What-If Process Analysis (Extended Abstract)	Yes	

A64	Adopting Assistive Technologies in Healthcare Processes: A Chatbot for Patients with Amyotrophic Lateral Sclerosis	Yes	
A65	Hey spot, can you help me with my business travel organization? - Design of a process-based chatbot artifact	No	Full text not available
A66	The impact of chatbots in driving digital transformation	No	Full text not available
A67	Transformation of insurance technologies in the context of a pandemic	No	Does not cover business processes in depth