Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Julia Caroline Seufert

Chatbot-Assisted Business Process Modelling - From User Interaction to BPMN Diagrams

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Julia Caroline Seufert

Chatbot-Assisted Business Process Modelling - From User Interaction to BPMN Diagrams

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. Ulrike Steffens
Zweitprüfer: Prof. Dr. Olaf Zukunft

Abgabedatum: 15.04.2025

Kurzreferat

Julia Caroline Seufert

Thema der Arbeit

Chatbot-gestützte Geschäftsprozessmodellierung – Von der Benutzerinteraktion zum BPMN-Diagramm

Stichworte

Prozessmodellierung, BPMN, Chatbots

Kurzzusammenfassung

In dieser Masterarbeit wird die Entwicklung eines Chatbots für die Geschäftsprozessmodellierung vorgestellt, der es Nutzern ermöglicht, BPMN-Diagramme über einen interaktiven Chat in natürlicher Sprache zu erstellen. Zu diesem Zweck wurde ein regelbasierter Bot auf Basis von Botpress implementiert, der die Nutzer Schritt für Schritt durch den BPMN-Erstellungsprozess führt und die beschriebenen Prozesse in ein korrektes BPMN-Modell überführt. Die Lösung wurde mit Nutzer:innen unterschiedlicher Vorkenntnisse evaluiert und zeigte, dass auch Nicht-Expert:innen mit diesem Tool korrekte BPMN-Diagramme erstellen können.

Julia Caroline Seufert

Title of Thesis

Chatbot-Assisted Business Process Modelling - From User Interaction to BPMN Diagrams

Keywords

Business Process Modelling, BPMN, Chatbots

Abstract

This master's thesis presents the development of a chatbot for business process modelling that enables users to create BPMN diagrams via an interactive chat interface using natural language. For this purpose, a rule-based chatbot was implemented using Botpress, guiding users step by step through the BPMN generation process and converting the described processes into valid BPMN models. The solution was evaluated with users of varying levels of prior knowledge and showed that even non-experts can create correct BPMN diagrams using the tool.

Contents

List of Figures

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
B2B	Business-to-Business
B2C	Business-to-Consumer
BPM	Business Process Management
BPMN	Business Process Model and Notation
CMMN	Case Management Model and Notation
DMN	Decision Model and Notation
EPC	Event-driven Process Chain
HCI	Human-Computer Interaction
ID	Identifier
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
ML	Machine Learning
MVP	Minimum Viable Product
NLP	Natural Language Processing
NLU	Natural Language Understanding
OMG	Object Management Group
SMEs	Small and Medium-sized Enterprises
UI	User Interface
YAWL	Yet Another Workflow Language

1 Introduction

The evolving landscape of modern enterprises necessitates efficient and innovative approaches to managing and optimizing business processes. Consequently, Business Process Management (BPM) has become an indispensable discipline, concentrating on the recording, design, analysis and enhancement of these processes. Business Process Model Notation (BPMN) has emerged as the de facto standard for business process modelling, providing a graphical representation of business processes through detailed and standardized diagrams (Silver, 2011). These BPMN recordings act as visual tools that facilitate better communication and understanding among stakeholders, enhancing the clarity and efficiency of process management (Figl, Mendling, and Strembeck, 2013).

Traditionally, the creation of process models involves a collaborative effort between domain experts, who have an in-depth understanding of the processes and process modellers or analysts, who possess the skills required for process modelling and analysis. This collaborative process, while crucial for developing accurate as-is models, is often time-consuming. According to Friedrich et al. (Friedrich, Mendling, and Puhlmann, 2011), the generation of these models can consume up to 60% of the time allocated for process management projects. This highlights the need for more efficient methodologies to streamline the modelling process.

The quality of process models often depends on the expertise of process analysts (Figl, Koschmider, and Kriglstein, 2013). BPMN is complex and prone to errors, particularly when created by individuals with varying levels of expertise. Human factors such as inadequate skill levels, distractions, misunderstandings and cognitive biases can lead to mistakes that compromise the accuracy and effectiveness of the created BPMN models. These errors can cause faults and inefficiencies in process management, underscoring the necessity for automated solutions that can aid in the creation and validation of BPMN diagrams.

BPM as an information systems discipline seems a viable candidate to benefit from chatbots, which can support users in creating and improving process-related content, most prominently process models. Chatbots can facilitate the process of model creation by enabling domain experts to articulate their knowledge directly into process models without the need for extensive intervention from process modellers, thus reducing the reliance on specialized modelling experts and making process management more accessible (Kourani1, Berti1, and Aalst1, 2024).

The rise of chatbots is evident across various business domains, including healthcare, customer service and e-commerce, where they provide significant improvements in efficiency and user experience (Huang and Rust, 2018). Chatbots are increasingly used to handle customer inquiries, provide personalized recommendations, answer FAQs and assist in troubleshooting issues, demonstrating their versatility and effectiveness in different sectors (Adam, Wessel, and Benlian, 2020).

In the context of BPM, the overarching question of this work is how and to what degree chatbots can replace the BPMN specialist when creating process models through conversational modelling with the domain expert. This research aims to explore the potential of chatbots to streamline BPMN creation, enhance model accuracy and democratize process management practices.

1.1 Problem Statement

Creating BPMN diagrams typically requires specialized knowledge and technical skills, which many organizations lack. This creates a barrier for small and medium-sized enterprises (SMEs), especially as they often cannot afford to hire BPMN specialists, thus limiting their ability to efficiently document their business processes (Leopold, Mendling, and Gunther, 2016). SMEs often struggle with BPMN modelling due to a lack of in-house expertise and the high costs associated with hiring external consultants. According to a study by Tam et al. (Tam, Chu, and Sculli, 2001), only around 40% of SMEs model the majority of their business processes, while one-third document only some processes. This indicates a significant gap in comprehensive process documentation capabilities, with the remaining SMEs not documenting any business processes at all.

In larger organisations, BPMN modelling is more prevalent, with 60% of BPMN modellers working in private sector companies with more than 1,000 employees (Polančič, 2022). This suggests that larger organizations are more equipped to leverage BPMN for comprehensive process management, unlike SMEs that often face barriers due to a lack of specialized knowledge and financial constraints. This gap can hinder their operational efficiency and strategic planning, making it challenging to compete with larger organisations that have more resources to invest in process management tools and expertise.

An additional challenge faced by both SMEs and larger organisations is that the tools available for BPMN modelling are not self-explanatory and can be overwhelming for non-experts (Friedrich, Mendling, and Puhlmann, 2011). This can lead to errors and formal inconsistencies in the process documentation and subsequent BPMN generation. Models created by novice analysts often exhibit issues such as control flow errors, incorrect labels, untidy layouts or soundness problems caused by mismatched gateways (Leopold, Mendling, and Gunther, 2016). These deficiencies impair the comprehensibility of process models, affecting their usability and leading to various problems in the management tasks they support (Krogstie, 2012; Wesenberg, 2011). Poorly designed models can result in miscommunication among stakeholders and increased costs due to the need for revisions and corrections. Therefore, high-quality process models are crucial for effective process management, yet achieving this quality is challenging without expert knowledge and experience (H. A. Reijers, Mendling, and Recker, 2014).

Process modelling guidelines, such as the Seven Process Modelling Guidelines (7PMG) proposed by Mendling et al., offer rules to improve model quality by enhancing comprehensibility and comparability (Mendling, H. Reijers, and W. Van der Aalst, 2010). However, since these guidelines require correct interpretation and application by skilled analysts, they do not inherently reduce errors. This further emphasises the need for more robust and accessible solutions that can assist in the creation and validation of BPMN diagrams.

While NLP-based BPMN generation has been explored as a means to simplify the modelling process, this approach also has significant disadvantages. Purely NLP-based systems often struggle with the complexity and specificity required in BPMN modelling (Sintoris and Vergidis, 2017). For instance, these systems may misinterpret user input, resulting in incorrect or incomplete models. Additionally, NLP models may at least not yet effectively handle the constraints and specific syntax of BPMN language, leading to errors in the generated diagrams (Kourani, Berti, and Schuster, 2024). The ambiguity and variability in natural language can further exacerbate these issues, making it difficult to ensure the accuracy and consistency of the models produced.

Adding onto the described challenge, even if NLP-based process modelling correctly extracts the text, it cannot guarantee that the BPMN model will be accurate. The domain expert providing the input process description might not deliver a complete and formally correct BPMN description (e.g., a condition might be missing, or a gateway might not be properly closed) (Klievtsova, Benzin, and Kampik, 2023). Formal verification remains a significant challenge with purely NLP-based approaches, as the user is not guided through the input sequence (Sintoris and Vergidis, 2017).

The integration of chatbots into BPM modelling presents a promising solution to these challenges. Chatbots can offer a solution by allowing domain experts to input their knowledge directly into process models via a conversational interface. The bot can guide users through the generation process without requiring them to understand the intricacies of BPMN. The underlying logic of the chatbot can then translate the user's input into the corresponding BPMN diagram. This reduces the reliance on specialized modelling experts and makes BPMN creation more accessible (Kourani1, Berti1, and Aalst1, 2024). Apart from democratizing the BPMN generation process, this approach can help mitigate common modelling errors and ensure the accuracy and completeness of BPMN diagrams. By providing interactive guidance and support during the modelling process, chatbots can make BPMN tools more user-friendly and accessible to non-experts.

Therefore, the primary goal of this thesis is to develop a chatbot that enables non-technical users to document business processes with ease and develop a tool that subsequently generates a formally correct BPMN diagram. The generated diagram, within the scope of this thesis, will include only the main elements of BPMN diagrams, specifically start and end events, process flow, tasks and both exclusive and parallel gateways. Additionally, the BPMN diagram will be editable post-generation, allowing users to make necessary corrections and refinements.

1.2 Motivation

The motivation for focusing my thesis on the integration of chatbots for BPMN generation is driven by both personal and academic interests, as well as the recognition of significant practical needs in the field of business process management. Personally, my academic journey has been centred around the study of process optimization and automation. Throughout my bachelor's and master's degrees, I have extensively explored various facets of process management, gaining a comprehensive understanding of the theoretical and practical aspects of this field. My two master projects were specifically focused on the field of business process management, giving me a solid academic basis and reinforcing my passion for this area. During one of my projects, I encountered the limitations of purely NLP-based approaches for generating business processes, particularly the lack of guidance and direct validation of user input for formal correctness. This issue, also highlighted in the problem statement, was further supported by empirical evidence collected during my master's project (Friedrich, Mendling, and Puhlmann, 2011; Sintoris and Vergidis, 2017). Addressing these challenges is a central goal of my thesis.
In addition to my academic background, my personal experiences working in different organizational settings have highlighted the pervasive nature of processes and the critical need for effective tools to manage them. Processes are integral to a wide variety of tasks, from scheduling trips and hiring new employees to admitting students into university programs and producing and delivering products (Leopold, Mendling, and Gunther, 2016). These experiences have underscored the importance of optimizing these processes to enhance organizational efficiency and effectiveness. The widespread nature of these processes showcases the critical importance of effective tools for managing and optimizing them.

The decision to focus on chatbots for BPMN generation is also influenced by the limitations I have observed in traditional BPMN tools. These tools often lack interactive guidance and can be overwhelming for non-experts, leading to errors and inconsistencies in process documentation. By integrating chatbots, I aim to provide a more user-friendly and accessible solution that can guide users through the process of creating BPMN diagrams, regardless of their technical expertise (Friedrich, Mendling, and Puhlmann, 2011; Kourani1, Berti1, and Aalst1, 2024). This approach not only addresses the practical challenges faced by organizations but also aligns with the objectives of my academic research.

Chatbots offer several compelling advantages that make them an ideal choice for BPMN modelling, particularly in addressing the issues outlined in the problem statement. One of the primary reasons for their adoption is their ability to engage in natural language conversations with users. Unlike traditional BPMN tools that can be overwhelming for non-experts and often lead to errors and inconsistencies, chatbots provide guidance, ensuring that users are supported throughout the modelling process. By providing a conversational interface with natural language conversations, chatbots can reduce the cognitive load on users, making the process of creating BPMN diagrams less daunting and more efficient (Kourani1, Berti1, and Aalst1, 2024; Sintoris and Vergidis, 2017).

In addition to providing iterative guidance throughout the conversation flow with the chatbot, they can help mitigate common modelling errors and help maintain the formal correctness of BPMN models. In contrast to purely NLP-based solutions, the bot can prompt users for necessary information and check for common modelling mistakes, addressing issues such as missing conditions or unclosed gateways (Klievtsova, Benzin, and Kampik, 2023). This interactive approach not only enhances the quality of the models but also reduces the need for costly revisions and corrections.

Chatbots reduce the need for specialized BPMN knowledge or BPMN analysts by directly giving domain experts a means to model their processes, as no knowledge of the underlying BPMN syntax is required. This capability is particularly beneficial for SMEs that often lack the resources to hire BPMN specialists, thus democratizing access to effective process management tools (Tam, Chu, and Sculli, 2001; Polančič, 2022). The ability of chatbots to provide coherent and context-aware responses further enhances their suitability for BPMN modelling. They can adapt to the user's level of expertise, offering more detailed explanations and support for novices while allowing more experienced users to proceed with minimal interruption. This flexibility makes chatbots a versatile tool that can cater to a wide range of users and use cases.

In conclusion, my motivation for integrating chatbots into BPMN generation is multifaceted. It is driven by a combination of personal academic interests and first hand observations of the limitations of traditional BPMN tools. By focusing on the approach to use a chatbot to model business processes, I aim to contribute to the field of business process management by developing a solution that simplifies BPMN creation, making it accessible and effective for a broader range of users.

1.3 Scope of the thesis

The focus of my research is clearly defined to address specific challenges in BPMN modelling without extending into broader areas of process management. Firstly, this thesis does not aim to improve or redesign existing business processes through process analysis or process redesign. The primary objective is to establish a robust foundation in process modelling. Future work can build on this foundation to incorporate process redesign and optimization. The scope of this research is also limited to the fundamental elements of BPMN, including events, gateways, the process flow and tasks. This focus ensures that the chatbot can effectively manage the basic constructs of BPMN before any expansion to more complex elements in future research.

Additionally, my approach does not involve using existing documentation or written records as input to create an initial framework for BPMN diagrams. This method is excluded due to the identified problem of guiding users through the process. Instead, the goal is to engage users directly and interactively in the modelling process.
Another key boundary of this research is the utilization of existing chatbot tools rather than developing new chatbot technology. The market offers several competent chatbot solutions that are suitable for this purpose, allowing the research to leverage these tools without the need for developing a new chatbot system from scratch.
Due to time and resource constraints, the research will include limited user testing and feedback. While some user testing will be conducted, a comprehensive user testing process, with multiple user testing sessions over the time period of development, will not be performed. Additionally, although user feedback will be incorporated into the chatbot, there will not be a thorough evaluation of the chatbot's performance through an extensive user evaluation process, but rather the user testing will be kept to a minimum.

Furthermore, this research will not involve the use of the Camunda API for generating BPMN diagrams. This decision is based on the complexity and cost associated with the pro version of the Camunda API, which would exceed the scope and resources available for this thesis. By narrowing the focus, the research can remain manageable and concentrated on the core objective of creating an easy-to-use BPMN modelling tool guided by a chatbot.

1.4 Thesis outline

The following chapters are structured to address the theoretical background, system design and empirical testing of the developed chatbot prototype. Chapter 2 provides an overview of the relevant background, introducing both the fundamentals of BPMN and the basic principles of chatbot technologies, including their classification and use in business contexts. Chapter 3 reviews existing literature on automatic BPMN generation, with a focus on NLP-based and chatbot-supported approaches and identifies gaps that this thesis aims to address.
Chapter 4 examines established methods for testing chatbots and evaluates their suitability for the present research. Chapter 5 presents the Bot2BPMN prototype, outlining the rationale for choosing a rule-based system and describing the designed conversation flow. This chapter also details the technical implementation, including the system architecture and integration of backend modules for diagram generation, as well as the applied testing procedures.
Chapter 6 covers the user testing phase, reporting on both technical performance and qualitative feedback, with particular attention to language settings, user experience and interaction patterns. Chapter 7 discusses the results in light of the research questions and reflects on the system's strengths and limitations. The thesis concludes with an outlook on future development, practical applications and areas for further research.

2 Related Areas

Given the objective of developing a chatbot that assists non-technical users in creating BPMN diagrams, it is essential to provide context on two primary components of the solution: business processes and chatbots. For business processes, various notations commonly used in literature and industry to represent these processes will be outlined. Subsequently, a focus on BPMN will be maintained, detailing its significance as a de-facto standard for business process documentation. The reasons for choosing BPMN will be explained, particularly highlighting its widespread adoption, standardization and effectiveness in accurately capturing the complexities of business workflows. Furthermore, the specific BPMN elements that will serve as the basis for the BPMN model created by the chatbot will be covered. Understanding BPMN's structure and benefits will clarify why BPMN was chosen as the modelling language for this project.
For chatbots, various classifications of chatbots will be explored, with particular emphasis on the different types of chatbots and the conversation flow. This understanding will be beneficial for the project as it will inform the design and implementation choices of the chatbot. This knowledge will aid in selecting the most suitable type of chatbot and in designing an intuitive conversation flow that can assist users in accurately creating BPMN diagrams.

2.1 BPMN and Business Process Management

2.1.1 Business Process Documentation and Notations

Business Process Management is a structured and systematic approach that focuses on the continuous improvement of organizational processes to enhance efficiency, effectiveness and adaptability (Brocke and Rosemann, 2010). It involves the careful analysis of existing workflows, the thoughtful design of optimized processes and the implementation of strategies that ensure their seamless execution. Additionally, BPM enables organizations to monitor operations in real-time, make data-driven adjustments and foster an environment of ongoing refinement. By integrating various methodologies and tools, it supports the design, enactment and management of operational business processes, helping organizations respond dynamically to both internal challenges and external market demands (W. M. P. Van der Aalst, n.d.).

One of the critical aspects of BPM is the documentation of business processes. This documentation serves several essential purposes. Firstly, it helps establish standardized procedures across an organization, ensuring consistency in execution and output quality (Brocke and Rosemann, 2010). Additionally, process documentation acts as a repository of organizational knowledge, facilitating knowledge transfer and reducing dependency on individual employees (Karagiannis, Junginger, and Strobl, 1996). Documented processes provide a baseline for analysis, enabling organizations to identify inefficiencies and areas for improvement (Friedrich, Mendling, and Puhlmann, 2011).
Furthermore, process documentation supports regulatory compliance and internal auditing by providing clear evidence of established procedures and controls (Sadiq, Governatori, and Namiri, 2007). It also serves as valuable training material for new employees, accelerating onboarding and reducing the learning curve (Hammer, 2014). Well-documented processes are essential for the successful implementation of process automation and digital transformation initiatives. Research indicates that organizations with mature process documentation practices demonstrate higher operational performance and agility in responding to market

changes (Wesenberg, 2011). Additionally, a study by Kohlbacher (2010) found a positive correlation between the degree of process documentation and overall organizational performance (Kohlbacher, 2010).
In conclusion, thoroughly documenting processes ensures clarity, consistency and transparency in organizational operations. Businesses have historically employed flowcharts to graphically represent their processes. However, the introduction of BPMN by the Business Process Management Initiative (BPMI) in 2004, later maintained by the Object Management Group (OMG), provided a more comprehensive notation. BPMN 2.0, the current version, offers a structured notation for developing graphic representations of business processes. Currently, it is designed so it is mostly understood by business analysts and technical developers (Object Management Group, 2011b).

Other notations for process modelling include Case Management Model and Notation (CMMN) and Decision Model and Notation (DMN), both developed by OMG. CMMN allows representation of weakly structured processes through the definition of a Case, a high-level activity comprising actions towards a known goal without specifying the exact path (Object Management Group, 2011a). However, CMMN is less suitable for workflows with a fixed order and lacks functionalities such as task synchronization and control flow patterns.
DMN, on the other hand, is designed for decision handling and representation (Object Management Group, 2015). It models decisions and their requirements, facilitating decision automation. However, DMN cannot represent workflows or model entire processes.
Two other alternatives to BPMN are Event-driven Process Chain (EPC) and Yet Another Workflow Language (YAWL). EPC, primarily used in German-speaking regions, focuses on events and functions but offers fewer features than BPMN (Mendling, Neumann, and Nüttgens, 2005). YAWL, while more technical and less graphical, requires considerable programming effort. Compared to YAWL, BPMN is more intuitive and widely used (YAWL Foundation, 2022).

A recent study by Compagnucci et al. (2023) highlights BPMN's extensive usage and its role as the preferred notation for business process modelling in most organizations (Compagnucci et al., 2023). Given this consideration, BPMN was chosen as the notation for this thesis, as it is widely regarded as the industry standard and has the most widespread adoption among business process modelling notations. Its comprehensive feature set and standardization make it an ideal choice for documenting business processes. Additionally, my experience with BPMN modelling further supports this choice, as I am able to judge the models generated more thoroughly. BPMN's broad acceptance and familiarity in the business process management community make it the most suitable notation for achieving the goals of this thesis.

2.1.2 BPMN elements included in the thesis

BPMN processes can incorporate a wide variety of elements, but in this thesis, we will focus on a few of the most common ones: events, tasks, sequence flows and gateways.

Events: In BPMN, events are occurrences that affect the flow of the process and can be classified as start, intermediate or end events. A start event signifies the beginning of a process, triggering the workflow to commence. Intermediate events occur between the start and end of a process, representing changes or updates that impact the process flow. End events signal the completion of a process. These events are crucial for defining the lifecycle of a process and ensuring proper flow control (Dumas et al., 2018).

Tasks: Tasks are the fundamental units of work within a BPMN process. They represent nuclear activities performed by human actors or automated systems (Dumas et al., 2018). Tasks can be of various types, such as user tasks, service tasks, script tasks and manual tasks, each indicating the nature of the activity. Proper task definition is vital for clear communication and effective process execution (Weske, 2007). In this thesis the various tasks types are not distinguished.

Sequence Flows: Sequence flows are connectors that define the order in which tasks and events are executed within a process. They establish the flow of control and ensure that activities are performed in the correct sequence. Sequence flows are depicted as solid arrows and are essential for mapping out the progression of the workflow, enabling a clear understanding of the process dynamics (W. M. P. Van der Aalst, n.d.).

Gateways: Gateways are decision points within a BPMN process that control the divergence and convergence of sequence flows. They determine the path a process will take based on conditions or events (Dumas et al., 2018). The two types of Gateways that are most commonly used are parallel gateways and exclusive gateways. These elements are pivotal for managing complex process logic and ensuring that processes adapt to varying scenarios (Decker and Mendling, 2008).

This brief overview of the elements incorporated into the chatbot's conversation flow will define the BPMN diagram of the underlying business process. Understanding the importance of these elements and their purpose is therefore vital.

2.2 Chatbots

Chatbots are software applications designed to simulate human conversation through text or voice interactions, utilizing natural language processing, rules and machine learning techniques to interpret user inputs and generate appropriate responses. The development of chatbots can be traced back to early experiments in artificial intelligence, with notable examples such as ELIZA in 1966 and PARRY in 1972 Weizenbaum, 1966 Colby, Weber, and Hilf, 1971. Modern chatbots have evolved significantly, incorporating advanced AI technologies to provide more sophisticated and context-aware interactions (Adamopoulou and Moussiades, 2020).
The effectiveness of chatbots is often evaluated based on their ability to understand user intent, provide relevant responses, solve a user's query and maintain coherent conversations. Researchers have developed various metrics and evaluation frameworks to assess chatbot performance, including task completion rates and the ability to handle complex queries (Scuotto, 2022). Recent statistics indicate that 88% of users had at least one conversation with a chatbot in 2022, demonstrating the widespread adoption of this technology. Interestingly, 58% of companies in the B2B sector actively use chatbots, compared to 42% of B2C companies. This higher adoption rate in B2B is primarily due to chatbots' success in lead generation, which is extremely valuable for the B2B sector (Pollock, 2024).

As of 2024, the chatbot landscape has undergone a transformation, with the integration of AI and machine learning revolutionizing their capabilities. According to a study by Gartner, conversational AI will reduce contact centre agent labour costs by $80 billion by 2026 (Gartner, 2022). This significant cost-saving opportunity highlights the importance of leveraging advanced technologies to enhance chatbot performance.

Chatbots serve various purposes across different domains, including customer service, information retrieval and task automation. In the business context, chatbots are increasingly employed to enhance customer engagement and provide 24/7 support (Adamopoulou and Moussiades, 2020). By handling routine inquiries and automating repetitive tasks, chatbots allow businesses to focus on more complex issues and improve overall efficiency. They can also personalize customer interactions, offering tailored responses based on previous interactions and preferences, thereby boosting customer satisfaction and loyalty (Pollock, 2024).

The integration of chatbots has shown significant benefits in terms of efficiency and customer satisfaction. By 2024, chatbots are expected to save businesses up to 2.5 billion hours of work. Moreover, 50% of customers report liking interactions with AI chatbots, appreciating their speed and 24/7 availability. As chatbot technology continues to advance, we can expect to see even more sophisticated applications across various industries.

The top five industries currently utilizing chatbots are real estate (28%), travel (16%), education (14%), healthcare (10%) and finance (5%). However, the potential for chatbot implementation extends far beyond these sectors, with businesses across all industries exploring ways to leverage this technology to improve customer interactions, streamline operations and drive growth (Shewale, 2024a).

As businesses continue to recognize the value of chatbots in enhancing customer experience and operational efficiency, further innovations and wider adoption of this technology across various sectors in the coming years are expected. In the realm of Business Process Management (BPM), the application of chatbots is gaining traction. Chatbots are currently mainly used to facilitate the understanding of business processes by providing real-time assistance and guidance. They can help users navigate complex BPM tools, answer questions regarding existing business processes and guide users through task completions. By integrating with BPM systems, chatbots enhance process efficiency and accuracy, reducing the time and effort required for process management tasks (Melnyk et al., 2024).

2.2.1 Types of Chatbots

In the realm of conversational AI, various chatbot types have been developed to address different needs and applications. This analysis categorizes chatbots into rule-based, generative-based, transactional and hybrid models, each possessing unique challenges and benefits. The objective of this examination is to determine the most suitable approach for this thesis.

Rule-based Chatbots: Rule-based chatbots, also known as deterministic or scripted chatbots, operate on pre-defined rules and structures to interact with users. These chatbots use a set of if-then-else logic to understand and respond to user inputs. Unlike AI-driven chatbots, which leverage machine learning to interpret user queries, rule-based chatbots follow a strict set of guidelines crafted by developers to simulate conversation.

Rule-based chatbots function through a combination of pattern recognition and pre-programmed responses. They rely on pattern matching algorithms to identify keywords or phrases in user input and then trigger the corresponding scripted response. The fundamental components include pattern matching, where the chatbot identifies patterns in user inputs using techniques such as Regular Expressions or string comparison algorithms. This enables the chatbot to detect specific keywords or phrases (Jain et al., 2018). Upon recognizing a pattern, the chatbot retrieves a pre-defined response associated with that pattern. The responses are typically stored in a database or a set of scripts that the chatbot references during interactions. These chatbots often employ decision trees to guide the conversation flow. Each node in the tree represents a decision point based on user input, leading to different branches and responses. Some rule-based chatbots use finite state machines to manage the conversation state. Each state corresponds to a specific part of the conversation and transitions between states occur based on user inputs (Jurafsky and Martin, 2008).

Rule-based chatbots offer several benefits. The deterministic nature of these chatbots ensures predictable behaviour, making them ideal for applications where consistency is crucial. Creating rule-based chatbots is relatively straightforward, requiring minimal computational resources compared to AI-based chatbots. Developing and maintaining rule-based chatbots is generally less expensive, as they do not require extensive data training or complex algorithms (Jain et al., 2018).

Despite their advantages, rule-based chatbots have notable limitations. As the complexity of interactions increases, the number of rules and patterns that need to be managed grows exponentially, making scalability challenging. Rule-based chatbots lack the ability to understand context or handle ambiguous queries, often leading to frustration when dealing with unexpected inputs (Shawar and Atwell, 2007). The conversations are highly scripted, which can result in a lack of naturalness and flexibility in interactions (Jurafsky and Martin, 2008).

Rule-based chatbots remain a fundamental technology in the realm of automated communication. Their deterministic nature ensures predictability and control, making them suitable for applications requiring consistency.

Generative-based Chatbots: Generative-based chatbot models, such as those powered by LLMs like OpenAI's ChatGPT, represent a significant advancement in natural language processing and conversational AI. These models utilize deep learning and neural network architectures, particularly transformers, to generate human-like responses in real-time based on the context of user inputs (Vaswani et al., 2017). Unlike rule-based chatbots that rely on predefined rules and responses, generative models are trained on vast datasets that encompass a wide range of conversational examples, allowing them to produce more natural and contextually appropriate interactions.

The underlying mechanism of generative-based chatbots involves sequence-to-sequence frameworks and advanced transformer architectures like GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers) (Vaswani et al., 2017). These models excel in understanding the syntactic and semantic nuances of language, which enables them to maintain coherence over extended conversations and handle complex dialogue scenarios (Radford et al., 2019).

Key components of these models include an encoder that processes the input and converts it into a meaningful representation and a decoder that generates responses based on this representation. The attention mechanisms within transformers further enhance their capability by focusing on relevant parts of the input text, thus improving the relevance and quality of the responses generated (Vaswani et al., 2017).

Generative-based chatbots offer several significant advantages. They can handle a wide array of conversational scenarios without the need for extensive scripting, providing more fluid and human-like interactions. This adaptability makes them suitable for diverse applications such as customer service, virtual assistance and interactive entertainment (Roller et al., 2020).

Generative-based chatbot models currently face several challenges, especially in specialized and well-defined scenarios. These models require substantial computational resources and extensive training data, which can be prohibitive for many organizations (Roller et al., 2020). Without proper safeguards, they can produce inappropriate or biased responses, raising ethical and safety concerns (Bender et al., 2021). In specialized domains, their generality can lead to responses that lack the necessary precision and depth, making them less effective for tasks requiring detailed knowledge. Additionally, integrating and maintaining these models requires significant customization and continuous updates, adding to their complexity and cost (Gururangan et al., 2020).

The flexibility and advanced capabilities of models like ChatGPT demonstrate the significant potential of generative-based chatbots in enhancing user experiences across various domains. While these models are continually improving, challenges in scalability and training data persist.

Transactional Chatbots: Transactional chatbots are designed to facilitate and automate specific business transactions, providing a seamless interface between users and backend systems. These chatbots are highly structured, leveraging predefined scripts and workflows to execute tasks such as booking appointments, processing orders and handling customer service requests. Unlike generative or rule-based chatbots, transactional chatbots integrate deeply with databases, APIs and enterprise systems to perform real-time transactions (Shewale, 2024b).

Key components of transactional chatbots include natural language understanding (NLU) to interpret user intents and robust backend integration to access and update databases or execute transactions. For instance, a banking chatbot might handle fund transfers by verifying user credentials, querying account balances and executing the transfer, all within a conversational interface. This high level of integration requires meticulous design to ensure security and compliance with relevant regulations (Melnyk et al., 2024). Additionally, these chatbots often use context management to track ongoing transactions and maintain relevant information throughout the conversation, ensuring a smooth and coherent user experience (Jurafsky and Martin, 2008).

Transactional chatbots offer significant advantages, such as improving efficiency by automating repetitive tasks and enhancing user experience through immediate responses. They also provide scalability, allowing businesses to handle large volumes of transactions without proportional increases in customer service resources. However, they also face challenges, including the need for sophisticated error handling to manage unexpected user inputs and the requirement for continuous updates to accommodate changing business processes and user needs (Shewale, 2024b). Ensuring data privacy and security is another critical concern, especially when dealing with sensitive information such as financial or personal data (Melnyk et al., 2024).

Hybrid Chatbots: Hybrid chatbots combine the strengths of both rule-based and generative models, aiming to deliver more robust and flexible conversational experiences. These chatbots use predefined rules and scripts to handle structured, predictable interactions while leveraging machine learning and NLP capabilities to manage more complex and dynamic conversations (Shewale, 2024b). This dual approach allows hybrid chatbots to offer both the

precision of rule-based systems and the adaptability of generative models.
Key components of hybrid chatbots include an initial rule-based layer that quickly addresses common queries and transactions using decision trees and finite state machines. For more nuanced or ambiguous interactions, the system can escalate the conversation to a generative model layer powered by advanced machine learning algorithms such as transformers (Vaswani et al., 2017). This layered architecture ensures that routine tasks are handled efficiently while still providing the flexibility to manage more complex queries (Jurafsky and Martin, 2008).

Hybrid chatbots offer significant advantages. They can improve accuracy and user satisfaction by ensuring that simple queries are resolved quickly and effectively through rule-based methods while leveraging AI to handle more sophisticated conversations. This approach enhances scalability and allows for continuous learning and adaptation, improving over time as the AI component is exposed to more data and interactions (Shewale, 2024b).
However, the implementation of hybrid chatbots presents challenges. Integrating rule-based and AI components requires careful design to ensure seamless interaction between the two systems. Additionally, maintaining such systems involves managing both the rule sets and the training data for the AI, which can be resource-intensive (Roller et al., 2020). Ensuring data privacy and security remains a critical concern, particularly when the chatbot deals with sensitive information (Bender et al., 2021).
In summary, hybrid chatbots combine the strengths of rule-based and generative models, offering efficient and flexible conversational experiences. Their dual-layer architecture enables them to handle diverse interactions, making them valuable for customer service and technical support. However, integrating and maintaining both systems can increase development costs and pose challenges in ensuring seamless interaction and high performance.

2.2.2 Different Intents of Chatbots

Chatbots are designed to fulfil various intents, which define the purpose and goals of their interactions with users. These intents can be broadly categorized into informational, transactional and conversational intents, each serving distinct roles in user engagement and service provision (Adamopoulou and Moussiades, 2020).

Informational Intents: These chatbots are primarily focused on providing information. They are designed to answer questions, provide updates or share knowledge on specific topics. For example, a weather chatbot provides current weather conditions and forecasts, while a healthcare chatbot offers information on symptoms and treatments (Shewale, 2024b). Informational chatbots use NLP or rules to understand user queries and retrieve relevant information from databases or predefined scripts.

Transactional Intents: Transactional chatbots facilitate and automate specific business processes, such as booking appointments or processing orders. These chatbots integrate deeply with backend systems to perform real-time transactions, ensuring security, accuracy and compliance. For instance, a banking chatbot can manage fund transfers by verifying user credentials and executing transactions through secure APIs (Melnyk et al., 2024). The high level of integration with enterprise systems is a defining characteristic of transactional chatbots.

Conversational Intents: Conversational chatbots aim to engage users in natural and dynamic dialogue, often leveraging advanced AI and machine learning techniques. These chatbots are designed to simulate human-like conversations, making them suitable for customer service and virtual assistance. Generative models, such as OpenAI's GPT, are typically used to enhance the conversational abilities of these chatbots, allowing them to understand context and generate coherent and relevant responses (Radford et al., 2019).

Supportive Intents: Additionally, some chatbots are designed with supportive intents, where they assist users in decision-making processes or provide recommendations based on user preferences and data analysis. For example, an e-commerce chatbot might suggest products based on past purchases and browsing history (Adamopoulou and Moussiades, 2020).

By aligning chatbot functionalities with user requirements, developers can enhance user satisfaction, improve efficiency and ensure seamless interaction across various applications.

2.2.3 Conversation Flow Design

Designing an effective conversation flow for chatbots is critical to ensuring a seamless and intuitive user experience. The conversation flow refers to the structured sequence of interactions between the user and the chatbot, guiding users through various dialogue stages to achieve specific objectives. A well-designed flow enhances user engagement and satisfaction, making interactions feel natural and efficient (Følstad and Brandtzæg, 2017; Shneiderman et al., 2018).

At the core of conversation flow design is the concept of visual design, which involves the arrangement and presentation of elements to communicate information effectively and appealingly. In the context of conversational interfaces, visual design plays a crucial role in guiding users through dialogue flows. It employs elements such as colours, typography and layout to make interactions intuitive and easy to follow (Shneiderman et al., 2018). Clear and structured dialogue flows ensure that users can navigate the conversation without confusion, enhancing the overall user experience (Følstad and Brandtzæg, 2017).

Conversational process modelling is another fundamental concept, describing the iterative creation and refinement of process models through exchanges between domain experts and chatbots through feedback cycles (Shneiderman et al., 2018). The iterative nature of this process allows for continuous improvement and alignment with user needs.

A crucial aspect of conversation flow design is the dialogue management component, which processes user inputs and determines the next system action based on the system's belief state. This component is responsible for maintaining the coherence and relevance of the conversation, dynamically adjusting the dialogue tree (Moore and Arar, 2019). This dynamic adjustment ensures that the system can select the most relevant topics for discussion, providing a coherent and engaging interaction flow (Grassi, Recchiuto, and Sgorbissa, 2022).

Moreover, behavioural programming offers a novel approach to designing conversational agents by specifying small, independent pieces of behaviour that can be combined to form complex interactions. This method contrasts with traditional dialogue graph techniques and has been shown to create more accurate and robust conversational agents, albeit requiring more time and effort from designers (Rosenfeld and Haimovich, 2022).

Finally, user-centred design principles are paramount throughout the development process,

focusing on the needs and preferences of users to create intuitive and engaging conversational agents. This involves defining clear goals, developing a unique identity for the chatbot, selecting the appropriate delivery interface and continuously testing and refining the system based on user feedback and performance metrics (Silva and Canedo, 2022).
The diagram shown in Figure 2.1 outlines key principles for developing effective chatbot interactions. It emphasizes building relationships such as retention or intimacy tailored to the chatbot's purpose. By designing conversations with the principles in mind, chatbots can enhance user engagement and satisfaction. The diagram also highlights practices to avoid, ensuring a seamless and positive user experience (Silva and Canedo, 2022).

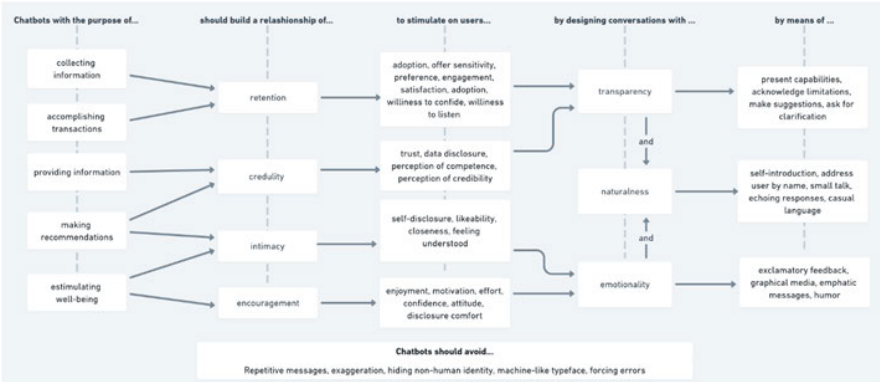

Figure 2.1: Example of a Conversation Flow Design. Source: Silva and Canedo, 2022

In conclusion, the design of conversation flows within chatbots is a multifaceted process that integrates visual design, conversational process modelling, dialogue management, behavioural programming and user-centred design. By focusing on these elements, designers can create chatbots that provide a natural, intuitive and satisfying user experience.

3 Literature review: Approaches to Automatic BPMN Generation

In recent years, the automation of BPMN generation has garnered significant attention within the field of Business Process Management. Bordignon et al. conducted a systematic literature review that highlights this growing interest and explores the potential of Natural Language Processing in automating business process modelling (De Almeida Bordignon, 2016).
In this literature review on “Approaches to Automatic BPMN Generation,” a comprehensive examination of the main existing methodologies will be conducted, along with an analysis of current trends and identification of gaps in the field. Additionally, the potential relevance of these findings to this thesis will be evaluated.

3.1 NLP for Business Process Modelling

Natural Language Processing has emerged as a powerful tool in Business Process Management, offering innovative solutions for creating, analysing and optimizing process models. NLP techniques enable the automatic extraction of process knowledge from unstructured text, facilitating the conversion of textual process descriptions into formal process representations. This approach bridges the gap between human-readable documentation and machine-interpretable models, making process modelling more accessible to non-technical users (Alkhatib, Jo, and Albustanji, 2024).

A study by Mendling et al. explored various methods for extracting process elements from textual descriptions and mapping them to BPMN constructs. Their study demonstrated the effectiveness of part-of-speech tagging, dependency parsing and named entity recognition in identifying activities, flows, events and gateways in business processes, showcasing the potential of NLP in automating BPMN generation (Friedrich, Mendling, and Puhlmann, 2011).

Nasiri et al. (2023) proposed an automated approach for generating UML activity diagrams from user stories and acceptance criteria in agile development. Their method employs natural language processing techniques and Prolog rules to analyze *"Given-When-Then* acceptance criteria, extracting activities, actions and control flows. The approach uses Stanford CoreNLP for preprocessing, coreference resolution and dependency parsing, followed by artifact extraction through defined Prolog rules. Testing on multiple case studies showed 94% accuracy in activity extraction and diagram generation, though with some limitations in detecting antonyms and decision nodes. Their work demonstrates the feasibility of automating business process model generation from agile requirements, reducing manual effort in requirements modelling (Nasiri, Adadi, and Lahmer, 2023).

Similar to the aforementioned study, Ferreira et al. (2021) proposed a semi-automatic approach to identify business process elements in natural language texts. They developed 33 mapping rules to detect BPMN elements like activities, events, gateways and swimlanes from text descriptions. The approach combines natural language processing tools with their mapping rules and includes steps for syntactic analysis, logic analysis and element identification. Their evaluation using 56 texts containing 387 sentences achieved 91.9% accuracy in identifying process elements. A survey of process experts showed 93.33% agreement with the mapping rules. The work aims to reduce manual effort in extracting process models from unstructured text documentation. However, the approach is limited by manually defined rules and focuses only on basic BPMN elements (R. C. B. Ferreira, Thom, and Fantinato, 2017).

Sintoris and Vergidis also investigated the use of NLP to automatically create business process models from organizational documentation, extracting components from the syntax and grammar of sentences. This is achieved through a series of techniques including part-of-speech tagging and dependency parsing. These techniques help identify the main verbs (representing actions), subjects (indicating the actors performing the actions) and objects (indicating the entities acted upon) (Sintoris and Vergidis, 2017).

Research by Mustansir et al. focused on extracting redesign suggestions from end-user feedback for business process redesign, using NLP to identify improvement areas. Their findings indicated that deep learning methods significantly outperformed traditional approaches in identifying redesign suggestions, highlighting the potential of NLP in business process improvement (Mustansir, Shahzad, and Malik, 2022).

Camargo et al. proposed a hybrid approach combining process mining and deep learning to improve business process simulation models. Their method, DeepSimulator, extracts a stochastic process model from event logs using Data-Driven Simulation (DDS) techniques and enhances it with a generative deep learning model to predict activity timestamps. The approach introduces activity embeddings instead of one-hot encoding, allowing for better generalization and the inclusion of unseen activities. An evaluation on multiple event logs showed that the hybrid model achieves temporal accuracy comparable to deep learning models while retaining the interpretability needed for what-if analysis. However, the approach relies on embedding quality and requires tuning for optimal performance (Camargo et al., 2023).

A proof of concept for GPT Codegen, a tool that integrates BPMN generation with OpenAI's ChatGPT APIs was introduced. The system employs an intermediate JSON schema to facilitate communication between the language model and the BPMN framework, enhancing the accuracy of the generated process models. To ensure robustness, GPT Codegen incorporates mechanisms such as fuzzy matching to handle extraneous model outputs and automated validation against the JSON schema to maintain BPMN correctness. The tool also utilizes chain-of-thought prompting, guiding the model to produce step-by-step natural language representations before constructing the intermediate data format. This approach aims to improve the quality and consistency of the generated models. While GPT Codegen demonstrates the potential for automating BPMN generation, it remains a proof-of-concept and is currently under active development. Future evaluations are necessary to assess its effectiveness and scalability in real-world applications (RobJenks, 2022).

Similarly, Finn Dohrns "Speech2Process" framework enables the direct transformation of verbal descriptions into structured models, leveraging process mining and NLP to simplify business process documentation. This tool is currently limited to easy sentence structure and the core BPMN elements (Dohrn, 2022).

Several studies have highlighted the potential of NLP in improving the efficiency of BPMN generation, while finding ways to improve the accuracy of machine-generated diagrams. For instance, a 2022 study introduced a machine translation-like approach to convert text descriptions into BPMN diagrams, achieving over 81% similarity to manually created models

(Sonbol, Rebdawi, and Ghneim, 2022).

Automating the generation of BPMN)diagrams from textual descriptions has gained increasing attention due to its potential to improve process modelling efficiency. A recent approach integrates NLP with BPMN to translate textual process descriptions into structured diagrams. The proposed technique applies Probabilistic Latent Semantic Analysis (PLSA) in the preprocessing stage to handle natural language ambiguity and improve element identification. The method processes textual requirements through NLP pipelines, extracting fact types that are mapped onto BPMN elements before generating a structured diagram. Evaluation demonstrates improved accuracy in representing business processes compared to previous rule-based or spreadsheet-driven methods. However, the technique relies on the effectiveness of semantic extraction and requires refinement for handling complex sentence structures and implicit process elements (Alkhatib, Jo, and Albustanji, 2024).

The thesis “Artificial Intelligence for Generation and Verification of BPMN Diagrams” by Anxhela Asllani explored the capabilities of AI models, specifically GPT-3.5 and GPT-4.0, in generating and verifying BPMN diagrams from descriptive problem statements. The study found that while AI models show promise, they do not yet match human performance in accuracy and error rates, emphasizing the need for further development (Asllani, 2024).

In conclusion, integrating NLP with business process modelling, particularly BPMN, offers significant potential for automating and enhancing process model generation. This reduces manual effort and improves diagram quality through innovative techniques and tools. However, a key challenge is the relatively low accuracy of NLP-generated models compared to human-created ones (Asllani, 2024).
Future trends point towards hybrid approaches that combine NLP with process mining, enhancing model reliability by refining initial text-based extractions against actual process data (Camargo et al., 2023). Additionally, advancements in deep learning and probabilistic semantic analysis are expected to further improve precision and context-awareness in model generation (Alkhatib, Jo, and Albustanji, 2024).

3.1.1 Modelling Business Processes with Process Mining Techniques

As organizations strive for greater efficiency and accuracy in their workflows, the application of process mining techniques has become crucial in modelling business processes with precision and insight. Process mining bridges the gap between traditional model-based process analysis and data-centric techniques, involving the extraction of knowledge from event logs to discover, monitor and improve real processes (W. Van der Aalst, 2016).
This literature review explores advancements in process mining methodologies applicable to business process modelling, with a particular focus on BPMN. Process mining can be categorized into three main types: process discovery, conformance checking and enhancement. Process Discovery aims to create a model from an event log without prior information, using techniques such as the alpha algorithm, heuristic mining and inductive mining (Leemans, Fahland, and W. M. P. Van der Aalst, 2013). Conformance Checking compares the event log with an existing process model to identify deviations, ensuring compliance and highlighting areas for improvement (W. Van der Aalst, Adriansyah, and Van Dongen, 2012). Enhancement focuses on extending or improving an existing process model using data from the event log to incorporate real-world variability and performance metrics.

De Weerdt et al. and Conforti et al. contributed significantly to the development of BPMN Miner, a refined technique for discovering hierarchical BPMN models from event logs. Traditional process mining techniques often produce flat models that do not leverage BPMN's advanced constructs such as subprocesses and boundary events. BPMN Miner addresses this by using functional and inclusion dependency discovery to create a hierarchy of processes and subprocesses. This technique applies existing flat process discovery methods, like Heuristics Miner and the -algorithm, to each node in the hierarchy, filtering out noise using approximate dependency discovery. This results in more accurate and less complex models, capable of capturing real-life complexities such as interrupting events and multi-instance activities (Miner, 2016; De Weerdt, Vanden Broucke, and Caron, 2015).

Another significant contribution is the Fuzzy-BPMN mining approach, which combines Fuzzy mining algorithms, adept at handling large numbers of activities and unstructured data, with BPMN. This method, applied to event logs from the IEEE CIS Task Force on Process Mining, effectively generates process models and accurately classifies traces. Validation experiments have shown a high percentage of correctly classified traces, demonstrating the method's ability to balance between overfitting and underfitting in process discovery (Okoye et al., 2017).

Furthermore, Kalenkova et al. explored how BPMN can enhance the understandability and applicability of process mining techniques. Conversion algorithms were developed to translate traditional process models like Petri nets and causal nets into BPMN models, preserving the behavioural properties of the original models. This integration allows for performance and conformance checking by converting BPMN models back into Petri nets for detailed analysis, providing a robust framework for discovering, verifying and enhancing business process models using BPMN. Their approach also introduces structural complexity metrics to compare discovered BPMN models with manually created ones, highlighting the advantages of BPMN for process visualization. However, the conversion process may introduce additional model elements, potentially increasing complexity in certain cases(Kalenkova et al., 2015).

The integration of process mining with traditional BPM practices has also been explored. Rebuge and Ferreira demonstrate how process mining can complement BPM by offering a data-driven perspective, providing a holistic view of business processes. Their methodology involves using process mining to extract process models from event logs and integrating these models with BPM practices to identify inefficiencies and areas for improvement (Rebuge and D. R. Ferreira, 2012). Van der Aalst et al. further illustrate this potential through conformance checking, comparing event logs with existing process models to identify deviations and provide insights into process compliance and performance (W. Van der Aalst, 2016). This integration is particularly crucial for organizations in highly regulated industries where adherence to prescribed processes is mandatory.

Emerging tools like Microsoft's Power Automate Process Mining further enhance the field. Microsoft's tool uses AI to automate process discovery and generate detailed BPMN models by analysing data from various sources, providing a comprehensive view of business processes and integrating seamlessly with other Microsoft Power Platform tools (Microsoft, 2024).

In conclusion, advancements in process mining methodologies have significantly enhanced the capability to model business processes accurately. However, future research should aim to improve the scalability, accuracy and integration of process mining tools to maximize their potential across various organizational contexts. While these advancements have addressed many challenges, ensuring that every touchpoint in the process leaves a digital trace remains a critical area for further development.

3.2 Modelling Business Processes with Chatbots

The integration of chatbots into business process modelling marks an advancement in simplifying and automating the creation of BPMN diagrams. These chatbots use conversational interactions to convert textual descriptions into structured models, reducing the manual effort and expertise required for creating process models.

The literature highlights a growing trend of leveraging chatbots for the automation of business process modelling, a movement supported by their ability to interact intuitively with users. For instance, López et al. demonstrated how chatbots can facilitate the transformation of BPMN models into interactive guides, effectively decomposing textual descriptions into structured BPMN elements and reducing the cognitive load on users (López et al., 2019).

Additionally, a study by Klievtsova et al. revealed that chatbots not only simplify the modelling process but also enhance model accuracy through iterative feedback mechanisms, which are critical in refining business processes. Their research systematically analysed existing chatbot applications for conversational process modelling and identified key scenarios where chatbots contribute, such as paraphrasing process descriptions, extracting process elements and assessing model quality. By evaluating chatbots on real-world process descriptions from the higher education domain, they demonstrated that AI-powered chatbots can improve process discovery efficiency and reduce human effort in manual modelling. However, their findings also highlight the limitations of chatbots in handling complex process semantics without human intervention(Klievtsova, Benzin, Kampik, et al., 2023).

BPMN-GPT is a specialized version of ChatGPT designed to assist users in creating and interpreting BPMN diagrams. Through a straightforward chat interface, BPMN-GPT converts textual descriptions of business processes into BPMN code and vice versa. Users can choose from predefined options that trigger specific conversation flows within the bot, including tasks like generating BPMN from text or validating a BPMN diagram. The generated BPMN code can be exported as either raw BPMN code or a .BPMN file, which can then be visualized using BPMN tools. This functionality enables non-technical users to create BPMN process models, enhancing the accessibility of creating BPMN diagrams (YesChat.AI, 2024).

Declo is a chatbot designed to help users easily create declarative process models using natural language input. The chatbot simplifies the process of defining constraints, especially for knowledge-intensive processes, by allowing users to provide statements in vocal or textual form. Declarative process models are a way to describe how things should be done without focusing on the specific steps to get there. In these models, verbs like "to create" show actions, while subjects and objects of verbs specify who does the action and on what it is done. It addresses the challenges faced with the existing Declare framework, making it more accessible to users without deep knowledge of temporal logic (Alman et al., 2020).

The SOCIO chatbot is designed to facilitate the creation of UML class diagrams through natural language interactions, making it accessible to users without extensive technical expertise. By integrating with social networks like Twitter and Telegram, SOCIO allows users to issue commands and descriptive messages in natural language to build and manipulate models. For instance, commands such as *"add class X"* directly manipulate the diagram, while descriptive messages like *"the shop contains products"* help build the context. SOCIO's natural language processing capabilities are supported by the Stanford parser and WordNet, ensuring accurate interpretation of user inputs. The system's usability was evaluated through a family of experiments comparing it with the Creately online collaborative tool, a frequently used tool to manually model UML diagrams. Results indicated that participants were faster at building class diagrams with SOCIO and more satisfied with its use, although the diagrams were slightly less complete. This research highlights SOCIO's potential for enhancing collaborative modelling through natural language processing and social network integration (Sara Pérez-Soler, 2018).

In addition to their role in modelling business processes, chatbots can further enhance existing workflows by optimizing parameters such as execution time, financial expenditures and other critical performance metrics. Barón-Espitia et al. detail the development of Coral, a chatbot created using the Rasa framework, which specializes in generating What-If scenarios within a business context. This chatbot intakes either manually constructed or data-driven process simulation models and interacts with users via a textual interface to formulate and test various simulation scenarios. Coral initiates simulations based on user-defined modifications, delivering projections of process performance and facilitating comparisons between diverse scenarios. This tool empowers business users to experiment with changes in demand, resource allocation, task timings and automation levels, thereby offering a platform for rapid decision-making and process enhancement without requiring deep expertise in simulation tools. It is important to note, however, that Barón-Espitia et al.'s study does not include an evaluation section, which limits the documented evidence of Coral's effectiveness in real-world applications (Barón-Espitia, Dumas, and González-Rojas, 2022).
The integration of chatbots into model generation, particularly for BPMN, remains a significantly under-explored area of research. Except for the BPMN-GPT tool, there are no other chatbots identified that assist in the creation of business process models. This gap in research highlights the need for further investigation into the development and application of chatbots for business process modelling. Future research could focus on designing and evaluating new chatbot systems that facilitate the creation of BPMN diagrams with user-friendly interfaces, thus guiding users through the modelling process.

4 Testing the Chatbot

Testing the chatbot is a crucial step in ensuring its reliability, accuracy and user satisfaction. Two primary methods are employed in this thesis: Key Performance Indicators (KPIs) and user testing. KPIs provide a structured, quantifiable approach to measure the chatbot's ability to generate accurate and complete BPMN diagrams, focusing on specific dimensions such as completeness, correctness and task accuracy. User testing complements this by capturing subjective feedback on the chatbot's usability and overall user experience. Together, these methods create a comprehensive framework to evaluate and refine the chatbot's performance, aligning it with both technical standards and user expectations.

4.1 Testing the Chatbot with KPIs

KPI testing serves to evaluate the chatbot's technical performance in generating process models by focusing on three critical dimensions: completeness, correctness and user engagement.

- **Completeness KPIs** assess whether the chatbot captures all essential elements of the process:
  - *Task Coverage* evaluates the proportion of tasks included in the process model compared to the source description, reflecting the chatbot's ability to identify and incorporate all relevant activities (Moody et al., 2022).
  - *Control Flow Coverage* measures the extent to which the generated model includes all necessary control constructs, such as sequences and branches, ensuring the structural integrity of the process flow.
- **Correctness KPIs** analyse the accuracy and logical coherence of the generated model:
  - *Task Accuracy* assesses whether tasks are correctly named and aligned with their descriptions in the input, ensuring fidelity to the source material.
  - *Control Flow Accuracy* evaluates the logical relationships between tasks, such as order and conditions, ensuring that transitions are modelled correctly.
  - *Semantic Alignment* examines whether the language and terminology used in the model are consistent with domain-specific vocabulary, ensuring relevance and contextual accuracy (Wang and Zhong, 2023).
- **User Engagement KPIs** evaluate how effectively the chatbot engages users and measures their interaction behaviour:
  - *User Engagement Rate* measures the percentage of users who interact with the chatbot after initiation. A higher engagement rate indicates that the chatbot effectively captures user interest and encourages interaction (He et al., 2024).
  - *Fallback Rate* assesses the frequency at which the chatbot fails to understand user inputs, leading to default or error responses. A lower fallback rate signifies better natural language understanding and response accuracy (Abd-Alrazaq et al., 2020).
  - *Conversation Length* analyses the average length of interactions to provide insights into user satisfaction and the chatbot's efficiency in resolving queries. Optimal conversation lengths vary depending on the chatbot's purpose; excessively long or short interactions may indicate issues (Santos, 2025).

- *Retention Rate* evaluates the proportion of users who return to use the chatbot after their initial interaction. A higher retention rate suggests that users find the chatbot valuable and are willing to engage with it multiple times (Dilmegani, 2025).
- *Customer Satisfaction Score (CSAT)* collects direct user feedback through post-interaction surveys to gauge satisfaction levels. High CSAT scores reflect positive user experiences and effective chatbot performance (Santos, 2025).

These KPIs offer a quantitative basis for identifying gaps in the chatbot's process modelling capabilities. By systematically applying these metrics, evaluators can measure the chatbot's alignment with user expectations and process modelling standards, as outlined in studies on conversational AI evaluation (P, 2023).
In addition to KPIs, benchmarks such as the Human-Computer Interaction (HCI) standards (ISO, 2018) can be integrated into the evaluation framework to assess usability dimensions like task completion rates, error rates and user satisfaction. These combined approaches provide a robust mechanism for refining the chatbot's performance in both technical and user-centric terms.

4.2 User Testing

User testing is a cornerstone of software development and product design, aimed at evaluating a system's usability and functionality from the end user's perspective (Rubin and Chisnell, 2008). For chatbots, user testing extends beyond traditional methods to focus on the nuances of conversational interaction, like user engagement and natural language understanding (Xu et al., 2017). Unlike software with static interfaces, chatbots rely on dynamic exchanges to fulfil user tasks, which necessitates a unique approach to testing.
User testing is a research method used to evaluate how real users interact with a system or product, focusing on identifying issues and gathering feedback to improve its usability and functionality (Nielsen, 2010). It involves observing participants as they complete tasks, capturing their experiences and analysing their behaviours to assess the system's effectiveness. This process ensures that the product meets user needs and expectations by addressing potential pain points and optimizing the user experience. In essence, user testing bridges the gap between developers and end-users, ensuring a product is intuitive and well aligned with its intended purpose. The objectives of user testing for chatbots are twofold. The first objective is to identify usability and performance issues, including errors in natural language understanding, response generation and task management (Radziwill and Benton, 2019). The second goal is to assess user satisfaction and long-term interaction quality. A well-tested chatbot is not only technically sound but also meets the expectations of its intended user base, ensuring effective communication and task fulfilment (Følstad and Brandtzaeg, 2018). Testing methodologies range from quantitative metrics like task completion rates and user retention to qualitative techniques such as observational studies and think-aloud protocols (Albert and Tullis, 2013). Combining these approaches allows developers to refine chatbots in alignment with both technical benchmarks and user-centric goals.

4.3 Methods of User Testing for Chatbots

When evaluating chatbots diverse methods can be employed to get the users feedback (Zamora, 2017). Methods such as think-aloud protocols, A/B testing and user satisfaction surveys delve into understanding user behaviour and satisfaction, while tools like Wizard-of-Oz testing and longitudinal studies assess chatbot adaptability and long-term effectiveness. By leveraging these varied techniques, developers can address both immediate usability challenges and broader design considerations, ensuring chatbots meet technical requirements and user-centric goals.

4.3.1 Think-Aloud Protocols

Think-aloud protocols are a qualitative research method in which participants verbalize their thoughts, feelings, frustrations and decision-making processes while interacting with a system, such as a chatbot (Ericsson and Simon, 1984). This approach provides real-time insights into how users perceive and navigate the chatbot, helping to uncover usability issues that can cause misunderstandings or lead to frustrations. For example, a participant interacting with a chatbot to book a flight might express confusion over vague prompts or difficulty understanding the sequence of steps. These verbalizations allow researchers to pinpoint areas of friction or ambiguity, providing actionable insights for improving chatbot design and functionality.

One significant advantage of think-aloud protocols is their ability to capture qualitative data on user thought processes and decision-making (Van Someren, Barnard, and Sandberg, 1994). This method sheds light on how users interpret chatbot responses, whether they find prompts intuitive and what mental models they form during interactions. Think-aloud testing is particularly useful in identifying instances where the chatbot fails to meet user expectations, such as delivering unclear instructions or providing irrelevant answers. However, the method is not without its limitations. Its effectiveness heavily depends on the participant's ability to articulate their thoughts clearly, which can vary significantly among users. Additionally, requiring participants to verbalize their thoughts may disrupt natural interaction patterns, potentially influencing how they engage with the chatbot. To mitigate this, researchers often use pre-session training to familiarize participants with the think-aloud process, ensuring more accurate and reliable results.
Recent advancements in user testing research have extended think-aloud protocols to hybrid approaches, such as combining them with eye-tracking or post-task interviews (Holmqvist et al., 2011). These enhancements allow researchers to correlate verbalized thoughts with physical behaviours, such as gaze direction, to gain a deeper understanding of user interactions.

4.3.2 A/B Testing

A/B testing, also known as split testing, is a quantitative evaluation method used to compare two or more versions of a chatbot to identify which design performs better based on predefined metrics such as engagement, task efficiency or user satisfaction (Kohavi, Longbotham, et al., 2009). By systematically altering specific elements, such as response phrasing, conversation flows or interface designs—A/B testing allows developers to determine how different variations impact user behaviour and interaction quality (Kohavi, Deng, et al., 2013). For example, one chatbot version might use formal language while another adopts a casual tone. Testing these variants can reveal which tone resonates more with users, guiding design decisions toward a more user-friendly chatbot experience.

One of the key advantages of A/B testing is its ability to provide clear, data-driven comparisons between design alternatives, helping developers make informed decisions about improvements. This method scales effectively for large user bases and is well-suited for iterative optimization, as it allows continuous refinement of specific chatbot features based on user feedback and observed behaviours. Furthermore, A/B testing is particularly useful for evaluating changes in isolated variables, such as button placement or message timing, enabling developers to fine-tune specific aspects of the user interface or conversation structure. However, A/B testing also has limitations. Achieving statistically significant results requires substantial user traffic, which may not always be feasible for smaller-scale deployments or niche chatbot applications. Additionally, while A/B testing excels at identifying surface-level preferences, it may overlook deeper usability or structural issues, as the method focuses on incremental changes rather than holistic evaluations. To address this, A/B testing is often complemented by qualitative methods, such as user interviews or think-aloud protocols, to gain a more comprehensive understanding of user needs and preferences (Deng et al., 2017).

Recent research highlights the growing importance of personalization in A/B testing for chatbots, emphasizing the need to account for user demographics, cultural differences and interaction contexts. By tailoring tests to specific user segments, developers can generate more targeted insights, ensuring that chatbot improvements align with diverse user expectations.

4.3.3 User Satisfaction Surveys

User satisfaction surveys are a widely used method for evaluating chatbot performance, focusing on collecting post-interaction feedback from users regarding their experiences (Molich, 2018). These surveys provide valuable insights into user perceptions of the chatbot's usability. ability to meet user expectations and overall satisfaction (Bangor, Kortum, and Miller, 2009). Typically, they employ standardized tools such as the System Usability Scale (SUS), a ten-item Likert-scale questionnaire designed to evaluate the usability of systems, or the Customer Satisfaction Score (CSAT), which measures user satisfaction with specific features or the overall interaction (Lewis, 2018). Questions in these surveys often include prompts like, *"How satisfied were you with the chatbot's response accuracy?"* or *"Was the chatbot easy to use?"* to gauge specific aspects of the chatbot's performance.

One of the major strengths of user satisfaction surveys is their ability to scale to large participant groups, making them particularly suitable for gathering diverse user feedback (Sauro, 2016). They are cost-effective and straightforward to administer, offering direct insights into user satisfaction and perceived usability. These surveys are also flexible and can be tailored to focus on particular aspects of the chatbot, such as response accuracy or efficiency, which helps in identifying targeted areas for improvement.

However, user satisfaction surveys have certain limitations. Responses can be subjective and influenced by individual user biases or recent interactions, making it challenging to correlate the feedback directly with measurable performance metrics (Albert and Tullis, 2013). For example, users who had a positive last interaction may rate the chatbot more favourably overall, even if previous interactions were suboptimal. Additionally, while these surveys effectively capture user sentiments, they may lack depth, failing to provide actionable insights into specific usability issues or interaction bottlenecks.

To address these challenges, surveys are often combined with other methods, such as task-based evaluations or think-aloud protocols, to provide a more comprehensive understanding of chatbot performance. Additionally, incorporating open-ended questions alongside Likert-scale items can help gather qualitative insights, enabling developers to better interpret user concerns and satisfaction levels. Overall, user satisfaction surveys remain a critical component of chatbot evaluation, providing essential feedback that can guide iterative design improvements and enhance user experiences.

4.3.4 Wizard-of-Oz Testing

Wizard-of-Oz testing is a user testing methodology in which a human operator secretly simulates the responses of a chatbot, allowing developers to evaluate interaction flows, user behaviour and overall usability without requiring a fully implemented system (Kelley, 1999). This approach is particularly valuable during the early stages of chatbot development when functionality may still be incomplete or under refinement. By having a human operator manage the interactions, developers can focus on assessing the conversation design and language choices without the constraints of technical limitations.

One of the primary strengths of Wizard-of-Oz testing is its flexibility (Dahlbäck, Jönsson, and Ahrenberg, 1993). It enables developers to explore complex scenarios that the chatbot might not yet be capable of handling autonomously, such as multi-step workflows or intricate decision trees. This allows for iterative testing and rapid prototyping of conversational flows before committing to development resources. Furthermore, the method provides insights into user behaviour, preferences, language choice and expectations, which can inform design decisions and ensure alignment with user needs.
However, Wizard-of-Oz testing also has notable limitations. It is resource-intensive, as it requires skilled human operators who must manage interactions in real-time while maintaining the illusion of an autonomous system. Additionally, because the chatbot's responses are controlled by a human, the results may not fully reflect the eventual capabilities of the automated system. Users may perceive a higher level of intelligence or flexibility during testing than the final chatbot can deliver, potentially leading to misaligned expectations.
To mitigate these challenges, researchers often combine Wizard-of-Oz testing with other methods, such as Think-aloud-Protocols evaluations or user satisfaction surveys, to gain a holistic understanding of chatbot usability. This hybrid approach can provide a balance between the qualitative insights gained from simulated interactions and the quantitative metrics derived from more structured testing. Despite its limitations, Wizard-of-Oz testing remains a valuable tool for refining chatbots during the design and prototyping phases, ensuring that the final product is both user-friendly and functionally robust.

4.3.5 Longitudinal Studies

Longitudinal studies are a valuable method for evaluating chatbots by tracking user interactions over an extended period to analyse trends such as learning curves and retention rates (Howard and Gutwin, 2022). Unlike short-term evaluations that provide a snapshot of usability or performance, longitudinal studies offer a deeper understanding of how user behaviour evolves with repeated interactions and how the chatbot's features are adopted and utilized over time (Følstad and Brandtzaeg, 2018). For instance, these studies can reveal whether users become more efficient in navigating the chatbot's interface or if they discover and begin utilizing previously overlooked functionalities as they gain familiarity with the system.

One of the key strengths of longitudinal studies is their ability to provide insights into long-term user satisfaction and feature adoption (Sauro, 2016). By observing patterns across multiple sessions, researchers can assess whether users consistently find value in the chatbot and whether its design supports sustained engagement. Additionally, longitudinal studies capture trends that may not be apparent in short-term evaluations, such as the gradual emergence of user frustrations, shifts in expectations or the impact of updates and new features on user behaviour (Albert and Tullis, 2013).
However, longitudinal studies also present significant challenges. They are time-consuming and resource-intensive, often requiring researchers to maintain access to participants and collect data over weeks or months (Molich, 2018). Furthermore, controlling for external variables—such as changes in user goals or environmental factors, potentially complicating the interpretation of results. For example, a decline in user engagement might be attributed to an unrelated factor, such as a seasonal drop in demand for the chatbot's services, rather than an issue with the chatbot itself.
To address these challenges, researchers often use a combination of qualitative and quantitative methods within longitudinal studies (Lewis, 2018). For example, periodic surveys can provide insights into user satisfaction, while interaction logs offer objective data on task completion rates, session durations and feature usage. This multi-method approach ensures that longitudinal studies deliver both depth and breadth of understanding, enabling developers to refine chatbots in ways that align with long-term user needs and expectations.

4.3.6 Pros and Cons of User Testing with Chatbots

User testing methods for chatbots vary significantly in their applicability, scope and the types of insights they provide (Radziwill and Benton, 2019). Quantitative methods, such as task-based evaluations and A/B testing, are valuable for generating measurable and objective data, such as task completion rates, response times and engagement metrics. These methods are particularly effective for benchmarking chatbot performance and identifying trends in user behaviour across large datasets. However, they often lack the ability to capture the nuanced experiences and emotional responses of users. In contrast, qualitative methods, including think-aloud protocols and Wizard-of-Oz testing, excel in exploring the subtleties of user interactions and uncovering areas of confusion or frustration that might not be evident through quantitative analysis. These approaches provide rich, contextual feedback but can be time-consuming and require skilled facilitators to interpret the data effectively.
Each method comes with its own strengths and limitations, making it essential to select the right approach based on the specific goals and constraints of the testing process. Combining quantitative and qualitative methods, alongside longitudinal evaluations, provides a holistic understanding of a chatbot's performance. This hybrid approach with different selected methods allows developers to address both immediate usability challenges and broader, systemic issues, ensuring the chatbot not only meets technical standards but also aligns with user expectations and long-term engagement goals. By leveraging a diverse set of testing methodologies, developers can create chatbots that are robust, intuitive and capable of delivering meaningful user experiences.

5 The Bot2BPMN Chatbot

The focus of this thesis is to create a chatbot that assists non-technical users in generating BPMN diagrams. This chatbot aims to bridge the gap between non-experts and the technical demands of BPMN, which traditionally requires specialized knowledge and skills. The goal is to make process modelling more accessible, efficient and user-friendly by guiding users step by step through the BPMN documentation process. This approach addresses the need for effective process documentation tools in organizations, especially small and medium-sized enterprises that may lack the resources to hire BPMN specialists.

This section is structured as follows: First, an introduction provides an overview of the chatbot and its purpose. The rationale follows, explaining why a chatbot was chosen for BPMN diagram generation. Next, the selection of a rule-based chatbot is discussed. The tools used, including Botpress, Ngrok and Graphviz, are then introduced. The chatbot's functionality is described, detailing its goals and expected outcomes. The conversation flow structure and its evolution are outlined. The system and conversation architecture are explained, followed by an overview of testing methods ensuring accuracy and compliance. Finally, the section concludes with a summary of the chatbot's implementation.

5.1 Reasons for Selecting a Chatbot for BPMN Diagram Generation

The evolving landscape of modern enterprises demands efficient business process management. BPMN has become the standard for modelling business processes through detailed diagrams that enhance communication and process clarity (H. A. Reijers, Mendling, and Recker, 2014; Dumas et al., 2018). However, creating BPMN diagrams typically requires specialized skills, which many SMEs lack (Tam, Chu, and Sculli, 2001). This creates a barrier for these organizations to document their processes effectively. Additionally, traditional BPMN modelling is time-consuming, taking up to 60% of project time (Friedrich, Mendling, and Puhlmann, 2011). This thesis proposes a chatbot that guides users in describing their processes step-by-step, automatically generating BPMN diagrams. This tool aims to make BPMN accessible to non-technical users, reducing the need for BPMN specialists and cutting costs.

The decision to utilize a chatbot for this task is driven by its potential to ensure BPMN standards and language criteria are met and to improve user experience by reducing the complexity of the BPMN generation process by navigating users through the business process documentation step by step. In a previous project, I encountered limitations with purely NLP-based approaches for generating business processes, particularly the lack of guidance and direct validation of user input for formal correctness. BPMN standards dictate specific graphical notations and rules that must be followed to create valid process diagrams, making this a crucial step in the generation of a BPMN diagram. These standards include the correct usage of elements such as events, activities, gateways and sequence flows, as well as ensuring proper syntax and structure of the models (Dumas et al., 2018).

Traditional methods of ensuring adherence to these standards are especially complex for non-experts and require knowledge of BPMN notation. Acquiring this knowledge can be time consuming and diagrams generated by non-experts are prone to errors (H. A. Reijers, Mendling, and Recker, 2014). By utilizing a chatbot, the conversation flow guides users through the process of generating a BPMN diagram, presenting specific options for answers and selections. This approach effectively acts as a way of conformance checking, ensuring that the basic syntax of BPMN standards is adhered to by only allowing the use of appropriate elements and ensuring that flows within the chatbot are BPMN compliant.

For example, the chatbot can ensure that an exclusive gateway, once opened, is closed before the user can add an event to end the business process, which is a requirement of BPMN standards. This process is mainly performed through the correct conversation flow of the bot and only allowing users to select appropriate answer or prompt options. However, while the chatbot can enforce adherence to some BPMN standards and syntax, it cannot eliminate all errors or ensure that the underlying business process is accurately depicted. This limitation arises because the chatbot relies on user input for generating the process flows. Users provide their textual descriptions and are in control of modelling the flows. The chatbot does not have access to underlying process logs or empirical data about the actual business processes. Consequently, it cannot verify whether the flows generated by the user accurately match the real-world business processes. This dependency on user input means that any inaccuracies or omissions in the user's description will be reflected in the BPMN diagram, making it crucial for users to provide precise and comprehensive information during the modelling process.

Another reason for choosing a chatbot is its ability to guide users through the BPMN generation process iteratively, making it particularly user-friendly for non-experts. The chatbot breaks down the complex task of BPMN diagram creation into manageable steps, prompting users for specific information at each stage. This step-by-step approach reduces cognitive load and minimizes the risk of errors by ensuring users focus on one aspect of the diagram at a time. By providing limited answer options for what to do next, such as selecting an event, activity or gateway—the chatbot prompts users to only think of the next step and not the process as a whole. This divide-and-conquer strategy makes the task less overwhelming and more manageable, as users can build the process incrementally.

A chatbot was also chosen for its capability to provide additional information and answer user questions about the process and BPMN in general while building the process with the user. The chatbot can offer explanations about various BPMN elements and this can help ensure that users understand the components they are working with. This educational aspect is particularly valuable for non-experts, as it helps them gain a deeper understanding of the BPMN modelling process and improves their ability to create accurate and effective process diagrams. By serving as both a guide and a source of information, the chatbot enhances the overall user experience, making BPMN modelling more accessible and informative.

Lastly, the chatbot's ability to gather user feedback easily and facilitate corrections after the BPMN diagram is generated and displayed to the user is a significant advantage. Once the diagram is created, the chatbot can prompt the user for feedback, asking if any corrections or adjustments are needed. If the user identifies an area for correction, the chatbot can gather detailed information about the specific changes required and assist in making those corrections. This post-generation feedback loop ensures that the final BPMN diagram accurately reflects the user's intent and needs, further enhancing the accuracy and usability of the process model. This iterative correction capability makes the chatbot an invaluable tool for continuous improvement and refinement of BPMN diagrams.

5.2 Selection of rule based chatbot for BPMN generation

For this thesis, I chose a rule-based chatbot developed using the Botpress platform due to several reasons (Botpress, 2024). Rule-based chatbots are designed to operate based on predefined rules and decision trees, making them highly suitable for applications where control and predictability are paramount. In the context of BPMN modelling, where accuracy and adherence to formal syntax are essential, rule-based systems ensure consistent and error-free outputs. Unlike generative models, which may yield unpredictable or contextually inappropriate responses, rule-based chatbots provide reliable performance by strictly adhering to predefined rules and structures, making them ideal for scenarios that require high precision (Caldarini, Jaf, and Mcgarry, 2022).

Another significant advantage of a rule-based approach is its relative simplicity in implementation and maintenance. Botpress is a framework that allows users to easily orchestrate and host these bots and does not require extensive coding knowledge or training data. The rule-based chatbot can be easily edited and amended should the conversation flow change, without having to adjust much of the underlying logic (Botpress, 2024).

Moreover, a key factor influencing the decision to opt for a rule-based chatbot is the lack of specific training data for a generative model specialized in BPMN generation. Generative models typically require large, domain-specific datasets to perform optimally in a given context (López et al., 2019). This specific training data was not readily available, making the rule-based approach more feasible.

Additionally, current generative models, as of December 2024, such as Gemini and ChatGPT, have not yet demonstrated the capability to produce accurate BPMN diagrams. While these models excel in handling complex language-based tasks, they struggle to generate structured outputs like BPMN models. These generative models often face challenges in maintaining the formal syntax and structure required for process modelling, making them less suitable for the precise and rule-bound nature of BPMN diagram creation (Asllani, 2024).

While rule-based chatbots offer significant advantages in terms of control, precision and simplicity, there are notable limitations that must be considered. One key downside is the inherent inflexibility of rule-based systems, which are constrained by their predefined rules and decision trees. This lack of adaptability can lead to limited scalability and difficulty in handling complex or unexpected user queries, especially as the conversation scenarios become more varied. Unlike generative models, rule-based systems cannot learn from new data or adapt to changes in user behaviour over time, requiring manual updates whenever new requirements arise (Jurafsky and Martin, 2008).

Another disadvantage of the current rule-based chatbot system is the limitation in the BPMN symbols it can handle. At present, only a restricted set of symbols (exclusive gateways, parallel gateways, events, start and end events) are included. For any additional symbols, such as a timer, the complexity of the bot would need to increase significantly. This is because the bot sends labelled data to the backend to generate the BPMN diagram and incorporating new symbols would require adjustments to both the conversation flow and the backend system. Specifically, a new dropdown or secondary multiple-choice prompt would need to be introduced for the user to clarify their input, adding more steps to the interaction. While this approach ensures precision, it also risks making the bot less user-friendly as the system and conversation flow becomes more complex. This added complexity could hinder the chatbot's ease of use.

5.3 Tool Usage

5.3.1 Botpress:

Botpress is a platform specifically designed for building and hosting chatbots. The company provides an open-source environment focused on the creation of rule-based chatbots, offering developers a user-friendly interface to design and manage bots without requiring extensive coding knowledge. Botpress features a modular architecture that allows for seamless integration of functionalities such as dialogue management and connections to external systems. The platform emphasizes control and user experience, making it particularly well-suited for applications where strict conversation flow management, like process modelling, is essential (Botpress, 2024).

In my thesis, I utilize Botpress to structure the conversation flow, which is essential for ensuring that the correct elements and steps of the BPMN are accurately captured from the user input. This structured approach allows the chatbot to guide users through a series of predefined steps, ensuring that the BPMN models generated are both consistent and complete. Additionally, Botpress supports the hosting of the chatbot, ensuring stable deployment and access, while its modular architecture allows for easy updates and adjustments to the conversation logic as needed. Furthermore, the platform provides the user interface, ensuring communication between users and the system.
While there are other companies that offer similar services, I chose Botpress as it was free and offered the functionality I needed. The editor for designing the conversation flow, Botpress Studio, was particularly easy to use and included many pre-built elements and integrations. Another reason for selecting Botpress was the availability of extensive tutorials and the option for 1-on-1 support via their Discord channel, which proved invaluable during the development process.
The free version of Botpress includes all necessary backend modules to design the conversation flow. At the core of this process is Botpress Studio, a drag-and-drop editor that enables the creation of conversation flows by linking different nodes. Within each node, developers can add cards, which define the specific tasks or actions the bot performs. Botpress Studio offers a range of pre-configured card functionalities that streamline development:

- **Send Messages Cards:** These cards allow the bot to send various types of messages to users, from simple text to rich media such as images, videos or documents.

- **Input Cards:** These cards collect user input and store it in variables. Pre-built options include multiple-choice input cards, which allow users to select from predefined choices and email input cards, which can validate user responses directly. There is a total of 12 different input cards alone.

- **Execute Cards:** These cards are used to perform specific actions, such as running custom code, creating timers or writing data to a database. These were particularly critical for backend functionality in my project.

- **Flow Logic Cards:** These regulate the conversation flow by enabling conditional branching, executing HTTP requests, or directing the user to specific subflows. They were essential for implementing the logic behind BPMN gateway creation and navigation through different modelling steps.

- **Additional Card Types:** Botpress also supports cards for AI tasks, logging and other advanced features, though these were less relevant for the focus of this project.

The captured user data or other variables can be stored in a relational database that Botpress also offers. This database was used to store the captured data regarding the BPMN events that have been captured by the user. It is the data that is added to the database that is sent out in a JSON format and builds the basis for the BPMN generation.
On the frontend, Botpress provides a web chat interface that allows users to interact with the bot through a browser. Once development is complete, the bot can be published and accessed via a URL, making it readily available to end users without additional infrastructure requirements.

Beyond the features used in this project, Botpress offers additional capabilities such as the creation of knowledge bases and pre-built flows for common tasks like HTTP requests. While I did not use the knowledge base feature, as the focus of my chatbot was on generating BPMN diagrams rather than disseminating information, these additional functionalities highlight the platform's versatility.
The modular design of Botpress allowed me to efficiently structure and manage the chatbot's components, ensuring both functionality and maintainability. Its user-friendly interface, extensive documentation and supportive community resources made it an ideal choice for this project. Combining robust backend tools with a seamless user interface, Botpress enabled the development of a chatbot that is both technically efficient and accessible for BPMN modelling.

5.3.2 Ngrok

Ngrok is a networking tool that facilitates the creation of secure tunnels from a public endpoint to a locally running service. It enables developers to expose local applications, running on their machines, to the internet by providing a temporary, public URL. This URL redirects external traffic to a specific local port, effectively bypassing network firewalls and allowing for remote access to the local environment (ngrok, n.d.).

In my thesis, I utilize Ngrok to establish a connection between my locally running Python application and external systems, in this instance Botpress. As the chatbot captures user input, Ngrok acts as a secure intermediary, transmitting the data from the chatbot to my local server. By doing so, I am able to retrieve, store and process this data on my localhost in real-time, without the need to deploy the entire application to an external server. This setup facilitates efficient testing and iteration during the development phase while ensuring data flows smoothly between the chatbot and the backend system.

5.3.3 Graphviz

Graphviz is a commonly utilized open-source software that specializes in the automatic generation of graph-based visualizations from structured data. It employs the DOT language to define the layout and relationships between nodes and edges, making it highly suitable for the representation of various types of structured information. The software provides precise control over the visualization, allowing for customization of elements such as node shapes, edge styles and hierarchical structures. Its efficiency in generating accurate and scalable visual outputs from raw data makes it particularly useful for applications that require clear and organized representations of complex systems (Graphviz, n.d.).

In the context of my thesis, Graphviz is employed for the visualization of the BPMN diagrams. The chatbot provides the necessary input as a JSON file, which encapsulates the BPMN elements and their relationships. Graphviz then processes this input, transforming the raw data into a visual diagram that accurately reflects the BPMN syntax and the structure and flow of the business process. By utilizing Graphviz, the raw data is transformed into a JPG that accurately depicts the process in BPMN notation.

5.4 AI Tools:

In addition to the technical tools used for system development and diagram generation, language assistance tools such as ChatGPT and DeepL were occasionally employed during the writing process. They supported the formulation and translation of specific passages.

5.5 What Should the Chatbot Be Able to Do?

The primary goal of the developed chatbot is to enable users without prior knowledge of BPMN modelling to model business processes effectively and for the bot to then output these processes accurately as BPMN diagrams. This is achieved through an intuitive conversational interface that guides the user step by step, ensuring that the generated diagrams are both correct and conform to BPMN standards.

To facilitate this, the chatbot employs simple and understandable language, making it accessible to users unfamiliar with BPMN concepts. Before initiating the modelling process, the bot offers explanations and tutorials on BPMN elements, providing a foundational understanding necessary for effective process modelling. This educational component ensures that users gain insights into the symbols and syntax of BPMN, which enhances their ability to articulate their business processes accurately.
Throughout the modelling process, the chatbot uses iterative prompts to guide the user, ensuring that each step is correctly captured and that the final diagram reflects the described process accurately. After the business process is described by the user, the bot outputs the generated BPMN diagram as an image. This ensures that the user can check the generated process visually for correctness.
An additional key functionality is allowing users to modify their modelled processes after generation without the need to restart the entire modelling sequence. These changes can be iteratively captured and implemented, focusing on renaming elements, adding new elements and deleting existing process components. This feature enhances the flexibility and usability of the chatbot, enabling users to refine their processes after the initial generation of the diagram.
In terms of user experience, the chatbot aims to be as user-friendly as possible. This will be evaluated through user tests on the one hand and also through testing KPIs. These include:

- **Task Accuracy:** A KPI that assesses whether the elements in the gateway are correctly named according to standards and user input.

- **Control Flow Coverage:** A KPI to measure whether all necessary elements, such as branches and conditions were present in the final BPMN diagram.

- **Conversation Length:** This KPI refers to the average number of interactions between a user and the chatbot.

- **User Satisfaction Survey:** A quick survey post usage, to capture the users satisfaction with the chatbot.

These metrics are based on established usability standards and methodologies, such as ISO 9241-11:2018, which defines effectiveness, efficiency and satisfaction as key components of usability (ISO, 2018). The importance of these metrics is underscored in foundational works by Ren et al., who emphasize the significance of usability evaluation in chatbot design (Ren et al., 2019). This focus on human interface design when using chatbots is underscored by works by Schneiderman et al. (Shneiderman et al., 2018).
To prevent overwhelming the user and to streamline the modelling process, the chatbot optimizes the number of questions asked during the generation process. This approach ensures a smooth interaction flow, contributing to a positive user experience.
Moreover, the chatbot's conversation flow is designed to allow only the creation of formally correct BPMN diagrams. This strict adherence to BPMN standards ensures that the outputs are not only accurate but also valid for practical applications.
In summary, the chatbot is designed to provide an intuitive and efficient platform for BPMN modelling, catering to users with no prior BPMN knowledge. The main goals of this chatbot are to facilitate users with no prior modelling experience to generate formally correct BPMN models.

5.6 Conversation Flow Design Design

The conversation flow design for the chatbot aims to ensure that users can intuitively and efficiently model business processes while generating BPMN diagrams that are accurate and adhere to BPMN conventions. To achieve this, several key principles and methodologies have been applied, focusing on user engagement, error prevention and goal achievement.

The primary objective of the chatbot is to guide users effectively toward generating BPMN diagrams. To support this, the chatbot uses a friendly and approachable tone, setting clear expectations and explaining the purpose of the interaction. By articulating the end goal clearly, the chatbot ensures that users understand not only what they are expected to do but also why each step is necessary, reinforcing a sense of purpose throughout the interaction. Research highlights that users are more likely to engage and complete tasks when they feel confident about the process and its relevance to their objectives (Følstad and Brandtzæg, 2017). Additionally, a positive tone can mitigate user frustration and foster a more enjoyable experience, which is particularly critical in tasks involving structured processes like BPMN modelling.
To streamline the process, the chatbot minimizes unnecessary user interactions. Every dialogue step is designed to elicit only the essential information required for the subsequent modelling phase. By avoiding redundant or overly complex prompts, the chatbot reduces the risk of user fatigue, which can occur when interactions feel overly repetitive or demanding. This approach not only improves task efficiency but also enhances the user experience by fostering a sense of clarity and momentum throughout the process. Research demonstrates that systems which maintain a focused and minimalistic interface are better at sustaining user engagement and reducing errors caused by decision fatigue (Shneiderman et al., 2018). By focusing on necessary inputs, the chatbot ensures that users progress efficiently without being overwhelmed by excessive choices or questions.

One of the core challenges in business process modelling is adherence to BPMN conventions. For example, exclusive gateways must include conditions and be properly closed to comply with BPMN standards. To achieve this, the chatbot is designed with a conversation flow that strictly guides users through valid modelling choices. By presenting only valid BPMN constructs via dropdown menus or multiple-choice inputs, the chatbot ensures that users can select options that are compliant with these conventions. This approach not only reduces the likelihood of common errors, such as incomplete or mismatched gateways, but also makes the modelling process more intuitive for users with limited experience. By embedding these constraints into the interaction design, the chatbot enables adherence to BPMN standards while simplifying the user experience, ensuring that the resulting diagrams are both accurate and formally correct.
The chatbot allows users to create complete and valid models within the predefined BPMN elements (start and end events, tasks, exclusive and parallel gateways). All permissible combinations and nested structures of these elements are supported, ensuring that users can model processes comprehensively without needing expert-level BPMN knowledge. To achieve this, the chatbot's interaction design incorporates dynamic validation mechanisms, which prevent users from selecting incompatible or incomplete options. This ensures that even complex workflows, including nested gateways or parallel sequences, can be accurately represented without requiring users to have an in-depth understanding of BPMN syntax or rules. By facilitating completeness within a clearly defined scope, the chatbot ensures that all possible business processes can be modelled accurately within its scope.
The chatbot employs an iterative approach to modelling, breaking the process into smaller, manageable steps. This design is rooted in the divide-and-conquer principle, which simplifies complex tasks by addressing one component at a time. By focusing on individual elements or decisions at each step, the chatbot ensures that users can progress methodically without feeling overwhelmed by the complexity of the overall process. This incremental approach also allows users to correct or refine their inputs at each stage, reducing the likelihood of cumulative errors. As a result, the iterative method not only improves user confidence but also enhances the accuracy and consistency of the generated BPMN diagrams.

Recognizing the varying expertise levels among users, the chatbot includes an optional tutorial to introduce BPMN terminology and concepts. This tutorial is designed to ensure that non-expert users can engage effectively with the system. It covers foundational elements such as tasks, events, flows and gateways, providing examples and explanations that demystify BPMN's technical aspects. For users who are already familiar with BPMN, the tutorial can be skipped entirely, ensuring that experienced users can proceed without unnecessary interruptions. This dual approach not only promotes inclusivity but also respects the varying knowledge levels within the user base, a key principle in conversational design (Følstad and Brandtzæg, 2017). By offering an adaptable onboarding process, the chatbot balances accessibility for novices with efficiency for advanced users, ensuring a seamless experience for all.

5.6.1 Evolution of the Conversation Flow: From Version 1.0 to 2.0

The conversation flow of the chatbot has undergone a significant change to enhance user experience and ensure adherence to BPMN conventions. In its initial design, version 1.0 allowed users to define what happens next in a process without requiring them to specify the corresponding BPMN element upfront. While this approach offered a degree of flexibility, it introduced notable inefficiencies and complexities that necessitated a redesign.

In version 1.0, as can bee seen in figure 5.1, every process step involved two separate questions: first, the user described what occurs next in the process and then they specified the type of BPMN element it represented (e.g. task, event or gateway). This two-step interaction not only made the process longer, but also created additional cognitive load for users. Moreover, this design posed challenges in enforcing BPMN conventions. For instance, if a gateway was opened, there was no direct mechanism to ensure that it would be closed later in the flow, increasing the likelihood of errors in the generated diagrams. Technically, implementing these checks dynamically within Botpress required additional logic, further complicating the conversation flow.

To address these limitations, version 2.0 was developed with a more straightforward approach. This version can be seen in figure 5.2. The revised flow prompts users to first select the type of BPMN element they wish to add before detailing the process step associated with it. This pre-definition streamlines the interaction by consolidating decisions into a single question for each step. By narrowing the options presented to users, version 2.0 reduces redundancy and ensures that each choice aligns with BPMN standards. For example, when a gateway is introduced, the chatbot proactively reminds users to define conditions or close the gateway as needed. This design not only simplifies the user experience but also enhances compliance with BPMN conventions. This version is overall more efficient and easier to model in Botpress.
Additionally, the modular nature of this approach leverages Botpress's capability to reuse loops for common patterns, such as exclusive gateway logic or a main flow. These reusable components not only reduce the complexity of the chatbot's logic but also improve its maintainability and scalability. The transition to version 2.0 highlights the importance of iterative refinement in chatbot design. This approach is more effective and user-friendly, as it reduces the number of questions required for each step while ensuring thorough adherence to BPMN conventions.

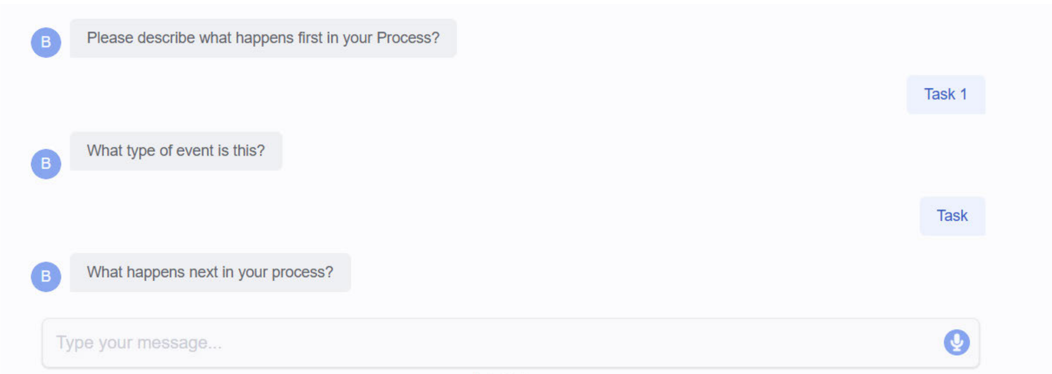

Figure 5.1: Conversation Flow Design - Version 1.0

5.6.2 Conversation Flow Architecture

The current conversation flow of the chatbot follows the structured principles established in Flow 2.0 and is composed of a main module, three sub-modules and an integrated API/http

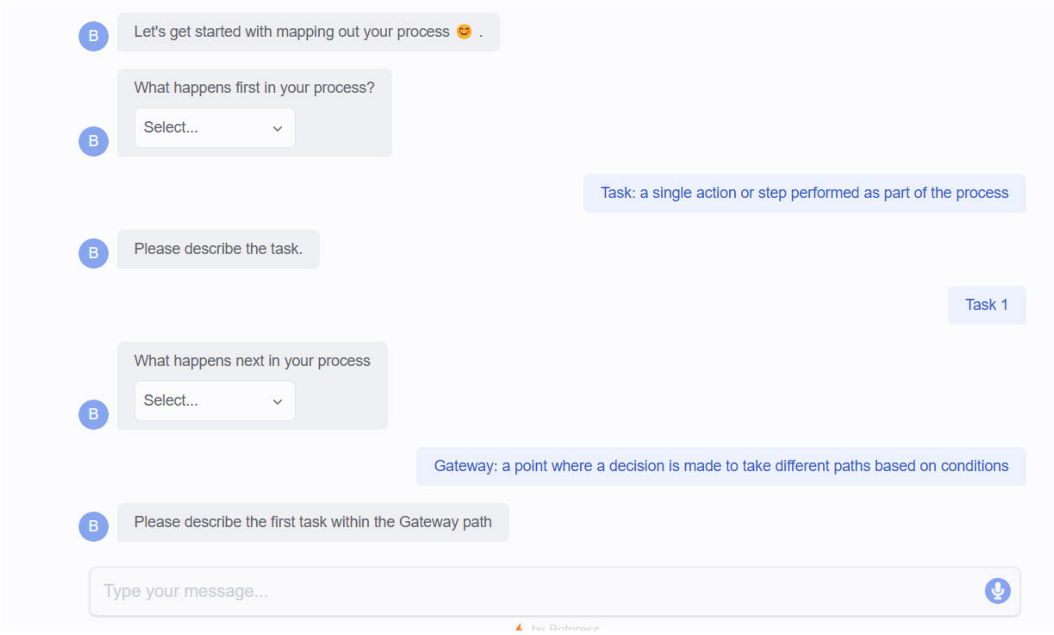

Figure 5.2: Conversation Flow Design - Version 2.0

request flow. Together, these components ensure a structured flow logic, where components have clearly defined tasks. Each module serves a distinct purpose within the chatbot's architecture, enabling both flexibility and adherence to BPMN conventions.

Main Flow: The Main Flow is the central hub of the chatbot's operation. It governs the primary user interaction, initiates the conversation and concludes the process by generating and displaying the BPMN diagram. This flow also incorporates the onboarding tutorial, ensuring that users unfamiliar with BPMN modelling are equipped with the foundational knowledge to proceed. A critical element within the Main Flow is the "what_happens_central" element, which acts as a control centre for guiding users through BPMN element selection. This element is accessed whenever there are no open gateways, providing users with full freedom to choose from all BPMN elements (e.g., tasks, parallel or exclusive gateways). The "what_happens_central" element simplifies navigation by allowing direct transitions to other subflows, such as those for exclusive or parallel gateways, based on user inputs. Once a subflow is completed, the user is redirected to the main flow, ensuring a consistent and intuitive user journey. The Main Flow also handles the communication with the Ngrok server. Upon process completion, it sends an HTTP request to generate the BPMN diagram and subsequently retrieves the finalized diagram for display. Additionally, this flow initiates the conversation regarding potential modifications to the generated diagram, enabling users to refine their models as needed.
Figure 5.3 shows the main flow in the Botpress modeller.

Exclusive and Parallel Gateway Subflows: The Exclusive and Parallel Gateway subflows are designed to handle the creation and management of gateways, ensuring that user inputs align with BPMN conventions. Although the purpose of these subflows is similar, their internal logic varies depending on the type of gateway being modelled. These subflows guide users through the process of defining gateways, including nested or chained gateways. For instance, if a user creates a nested structure, the subflow employs a counter to track the level of nesting and ensures that all gateways are appropriately closed before returning to the Main

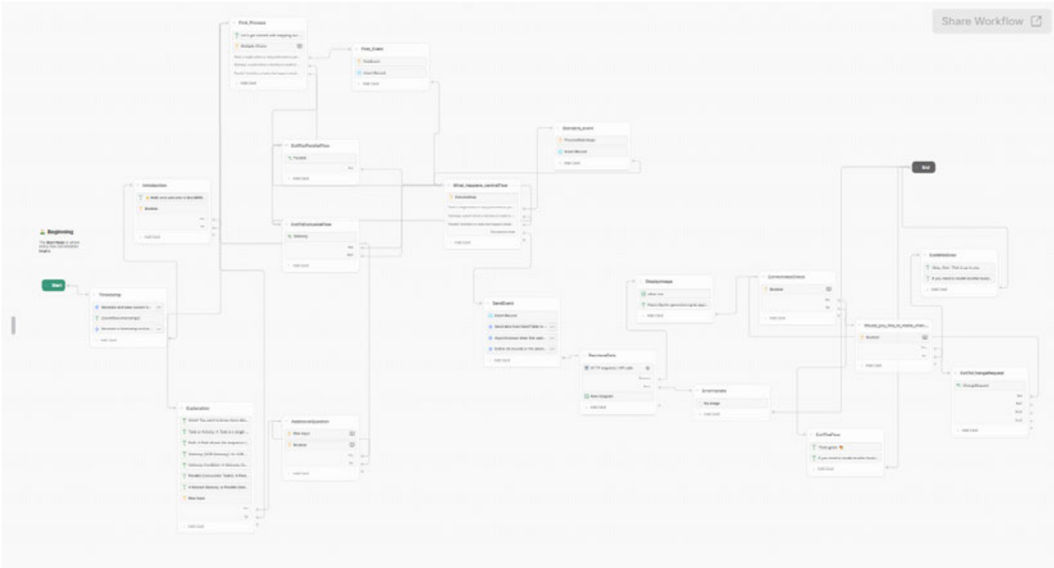

Figure 5.3: Main Flow of the Chatbot

Flow. This design reduces the likelihood of modelling errors and maintains compliance with BPMN standards. Figure 5.4 shows the exclusive Gateway subflow in the Botpress modeller.

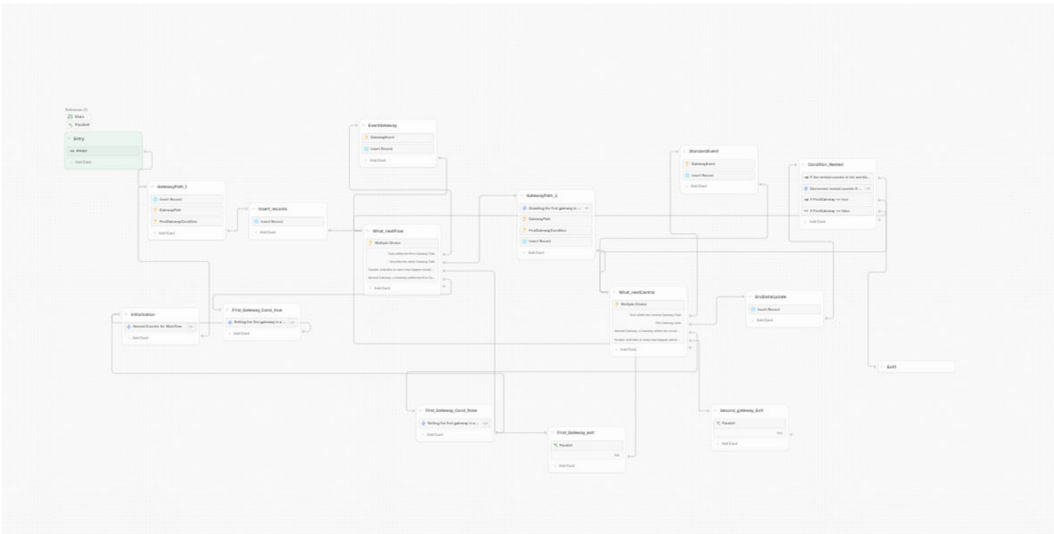

Figure 5.4: Exclusive Gateway subflow of the Chatbot

The parallel gateway flow is similar in it's architecture to the exclusive gateway subflow.

Modification Subflow: The Modification Flow allows users to adjust elements of their BPMN diagrams after they have been generated. Upon entering this flow, users are presented with their diagram annotated with numbers corresponding to individual elements. This annotation simplifies identification, enabling users to pinpoint specific elements for modification. The modification process only allows users to make one change at a time. Users can rename elements, add new BPMN elements or remove existing ones. The user must specify what they wish to change, as well as which element. The modification process can be accessed after the initial diagram rendering is complete. Figure 5.5 shows the modification subflow flow in the in the Botpress modeller.

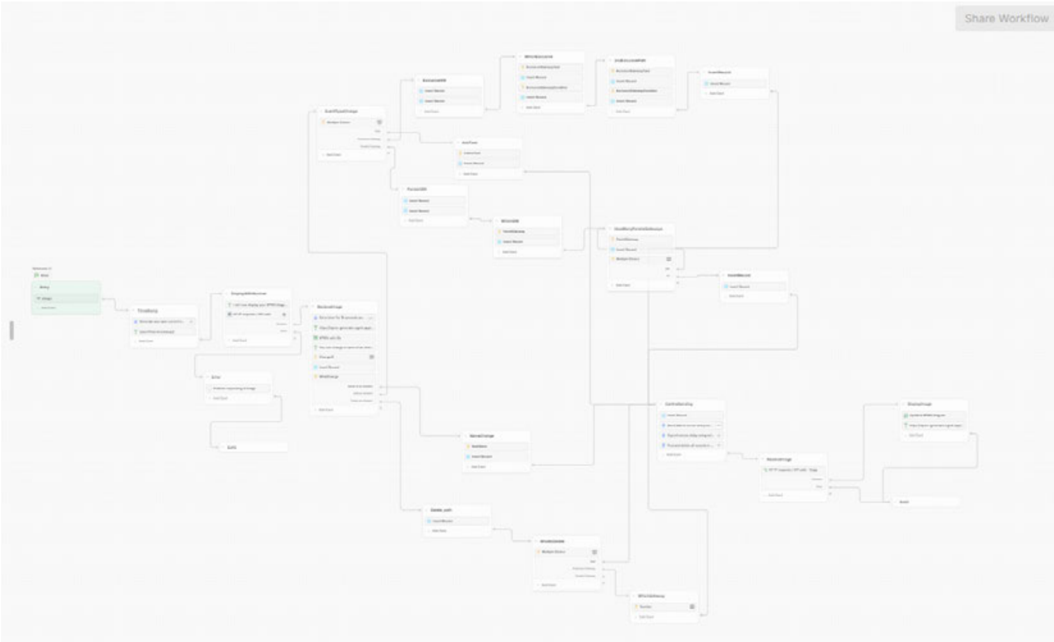

Figure 5.5: Modification subflow of the Chatbot

API/HTTP Request Flow:
The API/HTTP Request Flow is an auxiliary module integrated into the chatbot. Provided by Botpress as a pre-built feature, this flow manages HTTP requests, including the ‘POST‘ and ‘GET‘ requests that are used, to communicate with the Ngrok server. In this implementation, the API/HTTP Request Flow handles the submission of user inputs for diagram generation and retrieves the completed BPMN diagram for further processing. This flow also has error handling integrated.

Together, these flows form the backbone of the chatbot’s conversation architecture, enabling a structured and logical conversation design. The users input is captured either with a raw input or a multiple choice input and written into a database that is inherent within Botpress. This data then gets send when an HTTP request is made to build the BPMN. The modular design allows each flow to function independently while maintaining interoperability, ensuring that user interactions remain efficient, intuitive and compliant with BPMN standards. The Main Flow serves as the central orchestrator, while the subflows for gateways and modifications address specific aspects of the modelling process. The inclusion of the pre-built API/HTTP Request Flow further enhances functionality, providing a reliable mechanism for communication with external servers.

5.7 System Architecture

The architecture of the system integrates multiple components to enable the input from the chatbot to be processed into a BPMN diagram and then once again displayed to the user. At its core, the chatbot is built on the Botpress platform, which handles both the frontend and backend functionalities of the chatbot. Botpress provides the infrastructure for designing the chatbot’s conversation flow and managing user interactions while ensuring an intuitive and efficient experience. In addition to Botpress, a Python backend with a Flask server is used to receive the data from the chatbot and process it. To allow the locally running Python script to receive the data, an Ngrok tunnel is employed.

The conversation flow, created within Botpress Studio, defines how the chatbot guides users through the BPMN modelling process. Botpress studio acts as the main orchestration point that communicates with the frontend, the Ngrok server as well as the database. The Studio leverages Botpress's drag-and-drop editor, which allows developers to create structured interactions by linking nodes and defining tasks. Within this flow, the *Capture Information Cards* play a crucial role in collecting user inputs, such as BPMN element types and all other process steps. These inputs are stored in variables, which are then written to a relational database integrated into Botpress. This database serves as a temporary repository, maintaining data consistency throughout the interaction.

On the frontend, Botpress's Webchat feature enables users to interact with the chatbot directly through a browser interface. Users initiate the modelling process by typing into the Webchat and the chatbot responds dynamically according to the predefined conversation flow. The Webchat interface not only facilitates input capture but also provides feedback to users in the form of messages, the generated BPMN diagram or prompts, ensuring clarity and engagement throughout the modelling process.

Once the user completes their BPMN process modelling, the chatbot prepares the captured inputs for further processing. The data, stored as JSON, is sent via an HTTP request to Ngrok, which serves as a secure tunnel, forwarding requests from the chatbot to a locally hosted Flask server. Ngrok maps the external address https://bpmn-generator.ngrok.app to the Flask server running on http://localhost:5000.

The Flask application, described in the app file, serves as the backbone of the diagram generation process, facilitating communication between the chatbot and the BPMN generation modules. It includes several endpoints, such as /retrieve for receiving and storing user input in JSON format, /upload for handling the storage of generated BPMN diagrams and /clear for resetting stored data and clearing processed entries.

To ensure that the BPMN diagrams are correctly generated and made accessible, the /files/<filename> endpoint allows retrieval of uploaded files, while /files/<filename>_timestamp> ensures files can be accessed without interference from browser caching. Additionally, the application includes a background process, implemented as a separate thread, which periodically fetches new user input from the /retrieve endpoint and passes it to the BPMN generation logic in create_bpmn. This asynchronous approach ensures that new process models are generated without requiring manual intervention, optimizing workflow efficiency and minimizing delays.

Furthermore, a cache-clearing mechanism is implemented via the /clear-cache endpoint, which removes outdated data from memory and optionally deletes old files from the uploads directory to maintain storage efficiency.

The main Python architecture is organized around three main modules: create_bpmn, create_bpmn_car and modify_bpmn. These modules collectively handle the creation, visualization and modification of BPMN diagrams.

5.7.1 Module: `create_bpmn`

The process begins with the `create_bpmn` module, which functions as the central controller. This module receives user inputs in JSON format from the Flask server. These inputs, captured during user interactions with the chatbot, are processed and filtered to remove empty or invalid values. Additionally, the module ensures that each BPMN process is properly formatted by appending mandatory elements, such as "start" and "end" nodes, to the data structure. This preparation step ensures that the data is suitable for diagram generation. Once the input data is preprocessed, it is passed to the `create_bpmn_canvas` module for further handling.

5.7.2 Module: `create_bpmn_canvas`

The function `create_bpmn_canvas` is the primary method in this module, responsible for constructing the BPMN diagram. It initializes a new Graphviz `Digraph` instance and sets basic attributes such as node shapes and styles. The function iterates through the list of input elements, identifying their types and creating corresponding nodes. For example, start events are created as circular nodes labelled "Start Event," while tasks are represented as rectangular nodes with rounded edges. Connections between nodes are established using directed edges, ensuring that the sequence flow is maintained.
Parallel gateways are managed with additional complexity. The module tracks open parallel gateways and the associated paths using lists to ensure that all branches are properly closed and merged before proceeding. The function `create_bpmn_canvas` also handles exclusive gateways by introducing conditional branches with labelled edges to represent decision points. Nested gateways, which require additional attention to maintain hierarchical integrity, are tracked with counters to ensure proper closure and alignment within the diagram.
The module includes the `create_numbered_bpmn_canvas` function, which extends the capabilities of the basic diagram generation by assigning unique identifiers to each element. This numbering simplifies debugging and facilitates subsequent modifications by allowing users to refer to elements by their unique IDs. For example, a user modifying a gateway can directly specify its ID, reducing ambiguity and errors in the update process.
Additionally, the module saves the file in the uploads folder where it can be accessed for an upload to the specified endpoint. The module also includes utility functions such as `load_json_file`, which loads input data from a JSON file for processing. This feature supports scenarios where input data needs to be tested or reused without being directly supplied by the chatbot, which is needed in the modification feature of the chatbot, as the generation process should not needed to be started anew by the user.
Throughout the diagram creation process, the module incorporates validation checks to ensure that all BPMN elements are correctly represented and connected. For instance, it prevents the creation of orphaned nodes by ensuring that every node is linked to a preceding or succeeding element. By maintaining strict adherence to BPMN conventions and providing flexible tools for diagram customization, the `create_bpmn_canvas` module plays a critical role in ensuring the accuracy and usability of the generated diagrams.

5.7.3 Module: `modify_bpmn`

The `modify_bpmn` module enables post-generation modification of BPMN diagrams based on structured input received from the chatbot. Its core function, `create_updated_bpmn_diagram`, processes user-initiated changes such as element renaming, deletion and the addition of new elements. The module interprets and merges incoming events from Botpress thorugh the `create_bpmn` to ensure that all modifications are consistently applied to the existing BPMN element structure. Name changes are handled via the `name_change` function, while new elements are inserted using `element_add`. Deletion of elements is managed through a dedicated function that removes specified nodes while preserving the structural integrity and connectivity of the remaining diagram. Once all changes are processed, the updated BPMN diagram is regenerated using the `create_bpmn_canvas` function. This module allows for refinement of diagrams and supports a flexible modelling experience by eliminating the need to restart the diagram creation process from scratch. Currently, due to limitations in Botpress only one change can be applied.

5.7.4 Unit Testing for BPMN Diagram Generation

To ensure the correctness and reliability of BPMN diagram generation, unit tests were implemented using the `unittest` framework to evaluate the `create_bpmn_canvas` module. The test suite consists of 14 cases covering various BPMN structures, including sequential tasks, parallel and exclusive gateways and nested workflows. The 14 tests were selected as they cover the basic functionalities the BPMN diagram is expected to perform, as well as edge cases, such as an empty diagram with just a start and a finish.
Each test follows the same approach: a predefined JSON input representing a process model is processed by the `create_bpmn_canvas` function and the generated BPMN diagram is compared against an expected output. Any discrepancies in the output can therefore be analysed. The test cases assess key aspects such as basic diagram creation, correct sequencing of tasks, accurate handling of parallel and exclusive gateways as, well as correct processing of nested workflows. The nested workflows are nested three levels down, as further nesting is not significantly different than a thrice nested structure. So if the thrice nested structure performs well adding additional layers will not disrupt the generation process. More complex cases evaluate hybrid models that combine multiple control flows to ensure the chatbot handles interactions between different gateway types and including tasks at various stages in the gateways, correctly. The diagrams can be found in the appendix.
While unit testing validates the structural accuracy of BPMN outputs, it does not assess user experience or interaction efficiency. To provide a more comprehensive evaluation, unit tests are complemented by KPI analysis and user testing, ensuring that both technical correctness and usability are systematically assessed.

5.7.5 Summary of the architecture

The system's architecture employs a modular design to manage the creation, visualization and modification of BPMN diagrams. Botpress serves as the platform for user interaction, capturing inputs through the Webchat interface and storing them in a relational database. The chatbot's conversation flow, designed within Botpress Studio, ensures structured collection of inputs, which are then processed by the Python backend.
The backend is organized into three main modules. The `create_bpmn` module coordinates the data flow, preparing and validating inputs for diagram generation. The `create_bpmn_canvas` module uses the Graphviz library to generate visual representations of BPMN diagrams, ensuring compliance with BPMN standards. It also supports the generation of numbered diagrams to facilitate modifications. The `modify_bpmn` module processes user requests for updates to existing diagrams, such as renaming elements, modifying types or removing components.

A Flask server mediates communication between the chatbot and backend modules, providing endpoints for data retrieval, diagram uploads and cache management. Ngrok enables the locally hosted Flask server to securely receive and send data to the Botpress platform.
This architecture ensures that BPMN diagrams are created and modified in a systematic and methodical manner. By dividing responsibilities across specialized modules, the system maintains compliance with BPMN standards while supporting user-driven diagram generation and iterative refinement.

The system architecture illustrated in Figure 5.6 provides an overview of the key com-

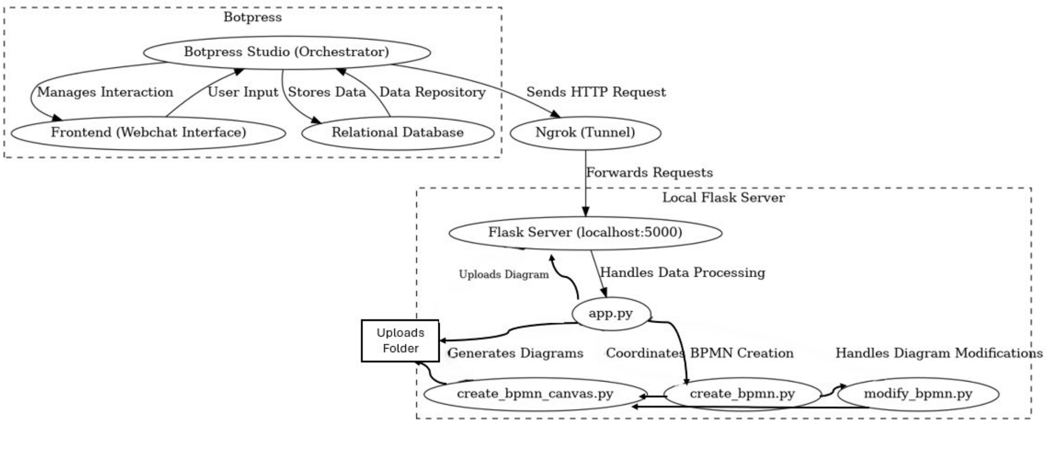

Figure 5.6: System architecture of the BPMN Chatbot

ponents and their interactions within the BPMN chatbot. At the top of the architecture, Botpress Studio serves as the orchestrator, coordinating the communication between the Frontend (Webchat Interface) and the Relational Database. User interactions originate in the Frontend, which communicates user inputs back to Botpress Studio for processing. The relational database temporarily stores these inputs, maintaining data consistency during interactions.
Botpress Studio forwards the processed data to a locally hosted Flask Server via Ngrok, which provides a secure tunnel for HTTP requests. The Flask Server, running on localhost, processes incoming data and directs it to the appropriate Python modules: create_bpmn for data preparation, create_bpmn_canvas for diagram generation and modify_bpmn for user-driven modifications. The generated BPMN diagrams are subsequently uploaded back to the Flask server for review and further refinement. This modular design ensures compliance with BPMN standards while enabling iterative diagram creation and user-driven customization.

Additionally, I have included two sample output of the chatbot in Figure 5.7 and figure 5.8, which demonstrates a BPMN diagram featuring an exclusive gateway and a task. This example highlights the chatbot's ability to generate visually accurate and logically structured BPMN diagrams based on user inputs.

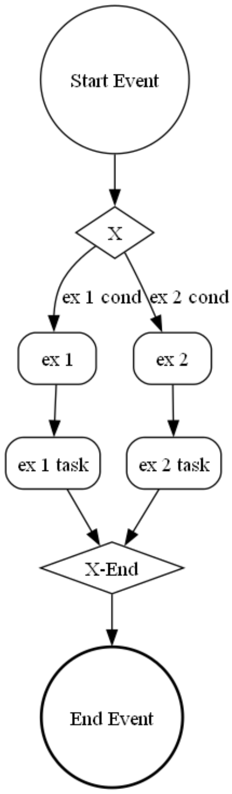

Figure 5.7: Example of the Chatbot output as a BPMN Diagram

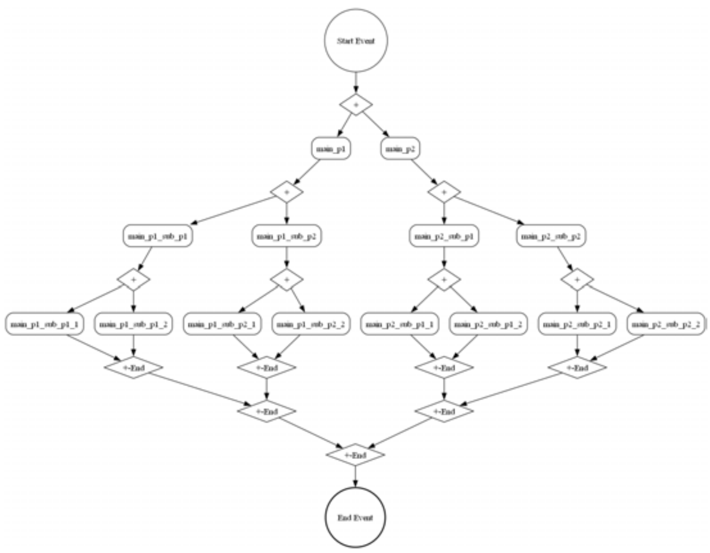

Figure 5.8: Example of the Chatbot output as a BPMN Diagram

6 Testing the BPMN Bot

A robust and balanced testing strategy is essential to ensure the BPMN chatbot fulfils both its technical performance requirements and user-centric goals. Designed to generate accurate and complete BPMN diagrams provided by inputs by the user, the chatbot's evaluation framework must address both its technical accuracy and the quality of the user experience. To achieve this, the testing strategy combines Key Performance Indicators with qualitative methods, including think-aloud protocols and user satisfaction surveys, to provide a comprehensive and nuanced evaluation.

The KPIs, task accuracy, control flow coverage and conversation length, were selected for their direct relevance to the chatbot's ability to produce correct and coherent process models. These metrics offer objective, quantifiable data on the chatbot's performance, focusing on the technical precision of its outputs and the efficiency of its interaction flow. However, quantitative metrics alone cannot capture the full spectrum of user experience, which is why qualitative methods are integrated into the strategy.

Think-aloud protocols allow real-time observation of user interactions, uncovering usability issues such as unclear prompts or confusing outputs, while user satisfaction surveys provide broad, scalable feedback on the chatbot's usability and overall effectiveness. Together, these methods create a balance between technical rigour and user-centric evaluation, ensuring that the chatbot is both functionally robust and intuitive to use.

By combining these approaches, the testing strategy captures both measurable performance indicators and qualitative insights, providing a balanced evaluation of the chatbot's functionality and usability. While this framework effectively targets key aspects of technical accuracy and user experience, it also involves trade-offs. For instance, the use of think-aloud protocols can be resource-intensive and may not fully represent natural user behaviour, while user satisfaction surveys risk oversimplifying complex feedback into subjective metrics. Nevertheless, this combination allows for a practical balance between the effort required and the depth of insights gained, even though limitations in scalability and comprehensiveness remain.

6.1 Selection of KPIs

The BPMN chatbot's core function is to convert provided process descriptions by the user into legitimate BPMN diagrams while maintaining technical accuracy and a smooth user experience. To achieve this, the selection of Key Performance Indicators and testing methods was driven by their relevance to the chatbot's core functionalities. The following KPIs were deemed most critical: Task Accuracy, Control Flow Coverage and Conversation Length. These metrics specifically address the chatbot's ability to generate technically sound BPMN diagrams.

Correctness KPIs are central to evaluating the BPMN chatbot's ability to generate accurate and logically coherent process models. These KPIs ensure that the chatbot's output aligns with user intent while adhering to process modelling standards, making them critical for assessing its technical performance. **Task Accuracy** is a particularly significant metric, as it evaluates whether the tasks in the generated BPMN diagrams are correctly named and aligned with the user's input. Ensuring this KPIs is met, is essential for maintaining the integrity and usability of the diagrams, as inaccuracies in task naming or alignment could compromise the clarity and purpose of the models. Accurate task representation directly reflects the chatbot's capacity to interpret user input correctly, reinforcing its reliability in translating complex descriptions into precise process elements. It is also important as it directly influences the user's satisfaction with the chatbot, because if the tasks of the business process are not correctly named the output is not usable for the user, leading to dissatisfaction with the chatbot. **Control Flow Coverage** is another crucial KPI, as it assesses whether the chatbot includes all necessary control constructs, such as sequences, gateways and nested gateways, in the generated diagrams. These constructs are foundational to the structural integrity of BPMN models, ensuring that the process flows represented are complete and also adhere to BPMN logic. Without adequate coverage, the diagrams risk being incomplete or inconsistent, undermining their usability and the chatbot's core purpose. This KPI is particularly relevant because it evaluates the chatbot's ability to handle process complexities, a critical aspect of BPMN modelling. This KPI also makes it possible to measure whether all diagrams that can be generated using the BPMN elements that were included, are actually able to be generated by the Chatbot. This metric therefore can help approximate the completeness of the generatable BPMN models. **Conversation Length**, while indirectly tied to technical performance, this KPI provides valuable insights into user satisfaction and the chatbot's efficiency in resolving queries. This KPI helps identify whether interactions are optimally balanced: excessively long conversations might indicate inefficiencies or misunderstandings, while overly short ones may suggest that the chatbot missed critical details of the process description. This balance is essential, as the chatbot must not only generate accurate diagrams but also do so in a manner that respects users' time and ensures clarity during the interaction. By analysing interaction length, this KPI offers a nuanced understanding of the chatbot's effectiveness in managing the conversational dynamics necessary for detailed process modelling. This will also measure the steps it takes a user to complete the task and can thus reveal if the chatbot only gives out each prompt once, or if there are errors in the conversation flow.

These KPIs were selected because they align directly with the chatbot's primary purpose: to transform user input into accurate and usable BPMN diagrams while maintaining an efficient and intuitive interaction flow. Together, they address the precision of individual tasks, the completeness of process structures and the overall interaction experience, creating one part of a framework for evaluating the chatbot's functionality and its alignment with user expectations. In contrast, some KPIs commonly used in chatbot evaluations were excluded due to their limited relevance to this specific application. For example, User Engagement Rate, which measures the percentage of users who initiate interactions with the chatbot, was not deemed relevant because the BPMN chatbot is a task-specific tool designed for users with a clear and deliberate intent to model processes. Unlike general-purpose or consumer-focused chatbots that must capture and maintain user interest to ensure interaction, the BPMN chatbot serves a more functional and utilitarian role. Its primary value lies in its ability to generate accurate BPMN diagrams, not in attracting or retaining casual users. Measuring engagement rates would provide little insight into the quality or effectiveness of the chatbot's technical outputs, as users typically engage only when they have a specific process modelling task to complete. Similarly, Fallback Rate, while important for chatbots handling broad and unpredictable inputs, is less applicable here because the BPMN chatbot operates within a constrained domain with relatively standardized inputs. Lastly, Retention Rate, which tracks whether users return after their initial interaction, is not applicable here as this is a strictly controlled experiment and the chatbot is not available to users to use at their pleasure at this time.

Therefore, this specific KPI is not measurable. While these metrics could provide additional insights in a broader context, their exclusion ensures that the evaluation remains focused on aspects that directly impact the chatbot's ability to generate precise and user-friendly BPMN diagrams. This prioritization avoids resource dilution and allows for a sharper, more effective assessment of the chatbot's performance.

6.2 Selection of User Testing Methods

The selection of user testing methods for the BPMN chatbot was guided by the need to evaluate its usability and user experience in a manner that complements the technical assessments provided by KPIs. After consideration, think-aloud protocols and user satisfaction surveys were chosen as the primary methods. These approaches were selected for their ability to address different aspects of user interaction. With one method offering detailed, real-time insights into user behaviour and the other providing broader, scalable feedback on overall user satisfaction. These methodologies were also chosen as they are easy to conduct by a non-expert in user testing such as myself.

Think-aloud protocols were selected due to their effectiveness in uncovering specific usability issues and cognitive challenges that users may face during interaction with the chatbot. By observing users as they verbalize their thoughts while completing tasks, this method allows researchers to identify problems such as unclear prompts, difficulties in interpreting generated BPMN diagrams or uncovering errors during the usage of the chatbot. This level of insight is particularly relevant for the BPMN chatbot, where precise user input and clear outputs are critical to its success, as misunderstood command prompts can lead to an unsatisfactory outcome, as the depicted BPMN diagram may not reflect the intended business process. However, think-aloud testing is not without limitations. The artificial nature of verbalizing thoughts during interaction can disrupt natural user behaviour, potentially leading to skewed results. Additionally, this method is resource intensive, as the researcher needs to be present with each think aloud tests and needs to record the specific responses.
User satisfaction surveys were chosen to complement think-aloud protocols because they enable the collection of structured feedback on users' perceptions of the chatbot's clarity and usability. Surveys allow for the design of detailed questions tailored to capture the specific feedback needed to evaluate the chatbot's performance, ensuring that the collected data is directly relevant to its purpose. Additionally, open-ended questions provide qualitative insights, while multiple-choice questions allow for quantitative analysis, offering a balance between measurable trends and deeper user reflections. An important advantage of surveys is that they provide users with time to reflect on their experience with the chatbot, potentially leading to more thoughtful and accurate feedback than what might be obtained during real-time testing. However, the limited scope of this project means the surveys will not reach a very large number of participants, which could reduce the diversity of feedback and make it more difficult to identify trends across a broader user base. Additionally, surveys are inherently subjective and responses may be influenced by individual biases or recent interactions, potentially skewing results. Despite these limitations, surveys remain an effective method for this context, as they allow for the inclusion of detailed and targeted questions that focus on the chatbot's key functionalities and user experience. This approach ensures that the feedback is both meaningful and actionable, while also complementing the real-time insights gained from think-aloud protocols.

Other user satisfaction methods, such as Wizard-of-Oz testing, A/B testing and longitudinal studies, were excluded due to their limited alignment with the specific needs and scope of this project. Wizard-of-Oz testing, while valuable during early development stages when chatbot functionality is incomplete, is less relevant here because the BPMN chatbot is already operational. Simulating responses manually would not provide additional insights into its automated performance and could create unrealistic user expectations about its capabilities. A/B testing, which evaluates variations in chatbot design or flow, was excluded due to its resource and time demands, which are not feasible within the scope of this project. Proper A/B testing requires multiple rounds, each isolating a single hypothesis, such as changes in phrasing or conversation flow, to ensure reliable results. This process also depends on a large and diverse user base to achieve statistical significance. Given the narrow focus on technical accuracy in BPMN diagram generation and the limited timeline and user pool, conducting such iterative tests would not provide sufficient value. While A/B testing could be useful for broader refinements in larger-scale deployments, it is not practical or relevant at this stage. Finally, longitudinal studies were excluded because they require tracking user interactions over an extended period, which is resource-intensive and challenging to implement without a large, sustained user base. While such studies could provide valuable insights into the chatbot's long-term usability and adoption, the project's timeline and scope do not allow for the extended data collection and analysis required. By focusing on think-aloud protocols and user satisfaction surveys, the testing strategy addresses the chatbot's immediate usability and performance needs while avoiding methods that demand significant time, resources, or expertise. These excluded methods, though valuable in other contexts, are not the most efficient or relevant for evaluating the BPMN chatbot's current development stage and objectives. For this purpose, the introductory text for the user test is also always the same.

6.2.1 Methodology for User Testing and KPI Evaluation

To evaluate the BPMN chatbot's usability and technical performance, a combined methodology involving user testing and KPIs was implemented. The methodology was designed to capture both quantitative data, KPIs and qualitative insights, through user testing methods, ensuring a comprehensive assessment of the chatbot's ability to generate accurate and complete BPMN diagrams while maintaining a user-friendly interaction flow.

Study Design and Participants: The study involved five participants, selected to represent a range of familiarity with process modelling, ensuring that the chatbot's usability could be evaluated across varying levels of expertise. This sample size was feasible within the scope of this project, but could of course be thought of as too narrow for a complete user evaluation. The evaluation combined think-aloud protocols and a user satisfaction survey which included the testing for predefined KPIs. Each participant interacted with the chatbot individually in a controlled setting, with sessions lasting approximately 10-20 minutes. Participants were provided with a brief introduction to the chatbot's purpose and functionality, those completely unfamiliar with BPMN Process modelling were shown a different process model shortly before beginning the modelling task. This was followed by a series of predefined tasks designed to evaluate the chatbot's performance. Tasks included modelling a pre-defined business process, making an addition to the process and reviewing the corresponding BPMN diagram generated by the chatbot for accuracy and completeness. A predefined process was chosen because test users have different levels of experience with business process modelling and so the results are comparable. If one chose a significantly easier or harder process to model, this might skew the results in terms of comparability. A researcher, which was myself, was present during the sessions to guide the participants and ensure the protocol was followed, while avoiding interference with their natural interaction with the chatbot. The researcher also recorded the sessions and tracked the KIPs. As the participants were all German with varying degrees of fluency in English, some user tests were conducted in German. While this does skew comparability a bit, it was deemed as more important that the users each knew exactly what they were supposed to do and could easily verbalize their thought process. The chatbot, which is written in English, was auto translated into German for two interviews, as this was deemed more user friendly for those two participants. Two of the participants had little to no knowelegde of BPMN modelling prior to the experiment. Another two participants were somewhat familiar with BPMN modelling and had used it in the past either during their work or came across the concept during their studies. One participant was quite familiar with BPMN modelling and uses the framework on a regular basis.

KPI Evaluation: Three Key Performance Indicators were used to quantitatively assess the chatbot's technical performance:

- **Task Accuracy**: This KPI assessed whether the tasks in the BPMN diagrams were correctly named and aligned with their descriptions in the user's input. During the sessions, I compared the generated diagrams to predefined reference models to evaluate the accuracy of task naming and alignment. Discrepancies were observed and categorized to identify patterns or recurring errors.

- **Control Flow Coverage**: This measured whether all necessary control constructs, such as sequences, branches and loops, were present in the generated diagrams. To track this KPI, I reviewed the structure of each BPMN diagram against a checklist of required control constructs for the predefined tasks. Missing or incorrectly modelled elements were documented to evaluate the chatbot's ability to handle the complexity of process flows.

- **Conversation Length**: This KPI tracked the average length of interactions between participants and the chatbot, measured by the number of messages exchanged during each session. Excessively long interactions were flagged as potential indicators of inefficiency, such as repetitive clarifications or unclear prompts, while overly short conversations were analysed to determine whether critical details were omitted.

Each KPI was documented for every task completed by the participants and the results were aggregated to identify trends and areas for improvement.

Think-Aloud Protocols: The think-aloud protocol method was employed to gain real-time insights into participants' thought processes, uncovering usability issues that could not be captured through KPIs alone. Participants were instructed to verbalize their thoughts, including what they found clear, confusing or unexpected during their interaction with the chatbot. This method provided granular feedback on specific aspects of the user experience, such as prompt clarity, ease of navigation and comprehension of the generated BPMN diagrams. The researcher observed and recorded participant behaviour, noting instances where users hesitated, expressed confusion, or misunderstood the chatbot's prompts or outputs. These observations were later analysed to identify recurring usability issues and inform recommendations for improvement.

User Satisfaction Surveys: After completing the tasks, participants were asked to fill out a user satisfaction survey designed to capture their perceptions of the chatbot's clarity, usability and effectiveness in generating BPMN diagrams. The survey consisted of both Likert-scale questions and open-ended questions to ensure a balance between quantitative and qualitative feedback. Likert-scale questions allowed participants to rate specific aspects of the chatbot on a scale from 1 (strongly disagree) to 5 (strongly agree), while open-ended questions provided an opportunity for detailed suggestions and insights. To ensure focus and relevance, the survey was designed with five short but critical questions that addressed key aspects of the user experience:

1. **Clarity of Prompts**: "The chatbot's prompts were clear and easy to understand."
2. **Accuracy of Outputs**: "The BPMN diagram generated by the chatbot accurately reflected the process I described."
3. **Ease of Interaction**: "It was easy to interact with the chatbot and complete the tasks."
4. **Overall Satisfaction**: "Overall, I am satisfied with my experience using the chatbot."
5. **Improvements**: "What improvements would you suggest for the chatbot to make it more user-friendly?"

Survey Analysis Methodology: Responses from the surveys and the KPIs will be analysed to identify common themes, focusing on recurring pain points and areas where the chatbot either performs well or requires improvement. The combination of quantitative Likert-scale responses and qualitative open-ended feedback is expected to provide both structured comparisons and deeper insights into user experiences. For instance, if multiple participants rate the clarity of prompts poorly, this may be flagged as a usability issue requiring refinement in future iterations. Conversely, if ratings for ease of interaction or task completion efficiency are consistently high, this could indicate strengths in the chatbot's conversational design and process handling.
A key advantage of this survey-based approach is its ability to systematically capture user perceptions while allowing for nuanced feedback through open-ended questions. This structure enables the identification of trends that may not be evident in real-time observations alone. However, the methodology also presents certain limitations. Given the relatively small sample size, survey results may not fully represent a broader user base, potentially leading to skewed findings if individual responses carry disproportionate weight. Additionally, while Likert-scale responses allow for quantifiable trends, they may not always provide sufficient depth to fully understand the reasons behind user frustrations or satisfaction. To mitigate this, the open ended questions and think aloud protocols were added.
Despite these challenges, it is expected that the survey analysis will yield valuable insights into both the chatbot's technical performance and user experience.

6.3 Analysis of the User Tests

The analysis of the user interviews was conducted only after all the interviews were finished and transcribed. The transcription was automatically done by a free software called TurboScribe (TurboScribe, 2023). The interviews as well as the generated BPMN diagrams from those interviews can be found in the appendix. For ease of reading and clarity vocal disfluencies or hesitations are removed, while pauses are still indicated by timestamps, as pauses can give an indication when a user might be hesitating. I have also translated the quotes by the German users into English, as to have a more fluid reading experience and keep the thesis available for reading for a larger audience. Due to issues with the chatbot interface the user tets were conducted directly in the Botpress emulator.

6.3.1 Technical Performance

The chatbot's performance in interpreting natural language descriptions into BPMN elements was consistently strong across all five user sessions, therefore task accuracy was uniformly high, each participant's described tasks were correctly identified and labelled in the generated diagrams. Users overwhelmingly agreed that the resulting process models matched their intent. As one participant observed, *"Everything looked accurate. I didn't spot any missing or misplaced elements"* (User 1), highlighting that the bot captured all the described steps without misinterpretation.
In terms of control flow coverage, the chatbot reliably constructed the proper gateways and sequence flows to represent branching logic. All required decision points (e.g. the stock availability check in the test scenario) were modelled and each user's diagram contained the necessary start, end and branching elements. Notably, every session achieved a complete BPMN diagram and the control flow was accurately depicted, as described by the user, meaning the chatbot did not omit any paths or tasks. There were, however, variations in the generated diagrams, as they were down to the user's interpretation of the process. For example, one intermediate user inadvertently introduced an extra nested gateway when a single exclusive gateway would have sufficed. This user struggled briefly to close the unnecessary gateway, causing a duplicated condition on one branch. Such an issue reflects a momentary slip in using the tool rather than a failure of the bot's understanding, highlighting a potential usability issue or unfamiliarity with the tool on the user's side. Likewise, a novice user confused the order of inputs when defining an exclusive decision: the chatbot prompted for a task name followed by the condition, but the user provided the condition first and then the task. This led to a slight mix-up in that branch's modelling, which is reflected in the diagram. Despite these hiccups, the outcomes were technically sound for all users – each final diagram fully represented the described process logic. Conversation length, interpreted as steps interacted with the chatbot, varied between users depending on the users input and flow through the bot. The longest interaction was with user 1 who added a nested gateway and the chatbot added more steps in order to correctly close the gateway. There was no session where the chatbot added unnecessary prompts and thus delayed the session.

6.3.2 Usability and Interaction Flow

The usability and conversational flow of the chatbot received generally positive feedback, especially regarding the clarity of its step-by-step guidance. Participants from varied BPMN familiarity levels felt the interface guided them through the modelling task in a logical manner. A common sentiment was that the bot's prompts were easy to follow during the initial process creation. For instance, one user noted being *"very surprised how easy it [was]"* (User 1) to articulate the process and see it reflected in the diagram, indicating that even without extensive BPMN knowledge, the sequential prompts (*"What happens first?"*, *"What happens next?"*, etc.) were intuitive. The think-aloud protocols show that users comfortably narrated their process (e.g. *"First, a customer visits an online store... then they place an order..."*) (User 3) and the chatbot responded with appropriate modelling actions. This clear mapping between user phrases and diagram updates helped build trust that the system was correctly understanding the user. Experienced users breezed through the basic steps, often declining introductory help (one expert said, *"I already know my way around BPMN"* (User 2)) and proceeded directly to the task prompts, while the novice users wanted the initial guidance. A user remarked, however, that it was too much text and information at once. This is probably due to the fact that basic concepts of BPMN modelling are all introduced to the user at the same time. This can be seen in figure 6.1.

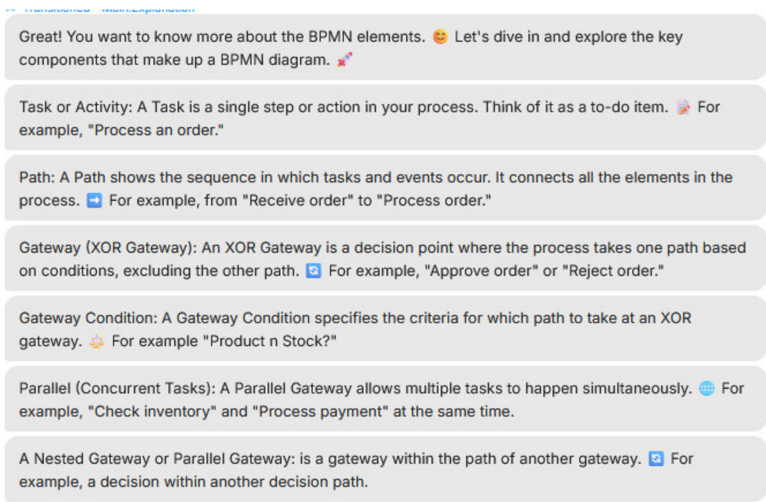

Figure 6.1: Compact Tutorial

Where the interaction flow showed limitations was in editing and more complex dialog sequences. When users needed to modify the diagram or handle the branching logic, the conversation became more repetitive. Multiple participants pointed out that the chatbot tended to break down actions into too many small confirmation steps, which sometimes tested the users' patience. One participant explained that *"when editing, it starts to feel a bit repetitive. It asks too many follow-up questions for small tasks — like, you have to confirm every little thing in separate steps"* (User 1). This comment reflects a pattern seen in three out of the five sessions: during tasks such as inserting a missed step or adjusting a branch, the bot would ask a series of clarification questions (e.g. confirming the ID of an insertion point, then confirming the content of the new task, then confirming completion. While this granular approach can prevent errors, it also made the conversation length longer for those users. Think-aloud comments revealed mild frustration with this rigidity.
Such instances suggest that error handling and prompt phrasing during edits could be improved. The chatbot did ensure no critical mistakes, such as an unclosed gateway, went through, but it also did not proactively guide the user back on track without human intervention. Despite these issues, it is worth noting that no user became completely stuck or failed to complete the initial modelling task, the repetitive confirmations were an annoyance rather than a barrier, resulting in moderate efficiency ratings for those sessions.
The user satisfaction surveys and qualitative feedback further highlight the difference in perspective between different users with the conversation flow. Users were satisfied with the chatbot overall, but their suggestions for improvement differed. Two users highlighted the clarity and guidance as positive. One of those users said that the small, guided tasks made it easy to follow along *"I think it was a 5 because it was a direct response with the chatbot and I got small tasks to do and was guided through the BPMM from beginning"* (User 1). This implies that not all users had the same preference and some valued the more rigid guided style of the chatbot. Other users found the repetitiveness tedious and wanted more flexibility. It is

important to note that this pro vs. contra rigidity group did not correlate to experience levels. It is also important to note that, while User 1 rated the repetitiveness in the feedback section as positive that sentiment was not necessarily reflected by the user during the process, as can be seen in the previous quote by User 1. In addition, one experienced user desired shortcuts and more advanced interaction options once the basic flow was mastered. For instance, an expert user proposed introducing an *"advanced option for task types, lanes or subprocesses"* (User 2) for those already familiar with BPMN, indicating a willingness to trade guided dialogue for more direct control when appropriate. Another common request, raised by both an intermediate and an expert user, was to streamline the editing process. They suggested the chatbot *"could group related questions together instead of asking them one by one,"* (User 2) and even enable direct manipulation of the diagram, such as *"being able to select an element visually — like just clicking on it in the diagram — would really help"* (User 3) during editing. These insights show that while the current design is user-friendly for users who prefer a more guided approach, its step-by-step clarity can feel constraining or inefficient to more advanced users or during iterative refinements. The balance between guidance and efficiency is a key theme, the chatbot succeeded in guiding all users to a BPMN that reflected their described process (though not necessarily the underlying process). However, this formal correctness comes at the cost of longer conversations and a sense of redundancy in the interaction.

6.3.3 Behavioural Insights and Misunderstandings during Think-Aloud Sessions

During the think-aloud sessions, several behavioural patterns and misunderstandings emerged as participants worked through the process of creating the BPMN diagram. All users were able to navigate the basic sequence of tasks (e.g. *"Customer visits online store"*, *"Customer places an order"*, *"System checks if product is in stock"*) without issue. Notably, no major task mislabelling occurred and participants described steps in their own words and the chatbot correctly translated them into BPMN matching the intended process flow.

Even participants with no BPMN experience managed to identify process steps in a way the bot could understand. One novice (Participant 4) with no knowledge of BPMN prior to the interview still managed to model the process correctly, demonstrating the approachability of the system for beginners. This approachability of the system for beginners was a major driving factor for the motivation of the thesis. Despite this overall success, misunderstandings arose around more complex BPMN constructs, especially the decision gateway. The process scenario included a branching logic (*"if the product is in stock, then... else..."*), which required using an exclusive gateway (XOR) in the diagram. Several participants showed confusion in constructing the gateway and its branches. For example, one intermediate user (User 1) inadvertently introduced a nested gateway instead of a single XOR gateway, then struggled to close that extra gateway. The chatbot had prompted the user step-by-step to add the gateway and subsequent tasks, but a misunderstanding led to an extra decision point being added inside the first branch. This confusion was evident in the think-aloud transcript: the participant appeared to enter a condition when the system expected a task, triggering an unintended gateway. The user's running commentary shows them attempting to correct course: *"I click the gateway ends. The nested gateway path is closed... I go to describe the other gateway path... If the product is not in stock, the order is cancelled"* (User 1). The linear prompt sequence (*"add task... then gateway... then task... then condition..."*) may not have been sufficiently clear for some users, leading to steps being done out of order.

Another common misunderstanding involved the order of specifying conditions vs. tasks in the branches of the gateway. One novice user (Participant 3) switched the order and input the branch's condition before describing the task that occurs under that condition. This happened for both branches of the XOR gateway, indicating the user's mental model differed from the chatbot's expected input order, probably due to the phrasing of the task for the user interviews. In the given task, the *"product stock check"* is described before the task as well due to the more natural sentence structure.

Process editing operations were another area where behavioural issues surfaced. After constructing the initial diagram, participants had the opportunity to make edits or additions. The chatbot handles edits by asking users to refer to diagram elements by an ID number. This proved unintuitive for some. One participant became stuck when trying to insert a missing task because they did not input the element ID in the expected format. The transcript reflects this mistake and the system's failure to recognize the input. The user initially included the prefix *"ID"* with the number (e.g. *"ID 466"*) and the chatbot gave an error and the user was not able to complete the change request portion of the task set in the user interview. This ID-based editing mechanism slowed down interactions: even another experienced participant remarked that *"manually typing IDs for edits might be a bit much for more complex diagrams"* (User 2).

6.3.4 Emotional Responses and User Confidence

Throughout the sessions, participants exhibited a range of emotional responses. This includes hesitation, frustration, surprise and confidence – which offer insight into the user experience with the chatbot. Early in the interaction, hesitation was common as users familiarized themselves with the chatbot's workflow. For instance, one user paused at the very start, saying *"Alright… so… let me take a moment to go through this scenario slowly"* (Participant 2) or User 4 who had quite a long pause of around 60 seconds reading the introductory text that was provided by the BPMN. This initial caution reflects users orienting themselves to both the given process scenario and the chatbot interface.

However, once they began inputting steps, most users showed growing confidence in describing the process. The three participants with prior BPMN experience were especially self-assured, because they immediately opted out of the optional tutorial at the beginning. This suggests that domain knowledge, in that case BPMN familiarity, contributed to higher initial confidence and a more relaxed demeanour during the interaction. In contrast, the novice users both opted to look at the tutorial. After having completed it, these users could also model the initial process steps with confidence and only expressed first hesitations at the gateway logic. It is important to note that both novice users selected the correct gateway type and were able to open and close the gateway, with one novice user managing to model the process completely correctly (User 4).

All users expressed mainly pleasant surprise when they saw the final outcome. After finishing the conversation and receiving the generated diagram in the chat, participants took a moment to examine the BPMN diagram. One user remarked with satisfaction that they *"didn't spot any missing or misplaced elements"* in the diagram (User 1). This expresses confidence that the chatbot accurately modelled their described process. A couple of participants became impatient during the editing phase or when the bot's responses were slow. Participant 5, for example, noted that the bot essentially *"takes a bit long to compute, which I find kind of annoying."*

Another participant expressed annoyance at themselves when they encountered a problem. After inputting an incorrect ID format and realizing the mistake, the user said, *"I made the mistake, that annoys me"* (User 3), rating their satisfaction slightly lower as a result. This comment reveals how user error, even if partly due to system design, translated into self-frustration. On the positive side, whenever participants successfully overcame an issue (for example figuring out the gateway or correcting an input), their tone shifted to relief or contentment. By the end of each session, most users conveyed a sense of satisfaction. In the post-test questionnaires, three out of five participants rated their overall experience as very positive (5 out of 5), often citing that they could complete the diagram successfully. Even those who gave slightly lower ratings (4 out of 5) still acknowledged the system's usefulness, docking points only for the minor frustrations mentioned. These emotional insights underscore that user satisfaction was generally high, boosted by the chatbot's reliable outcomes, but could be undermined by interaction hiccups that caused short-lived confusion or irritation.

6.3.5 Influence of Language and Translation on Interaction

The user testing included sessions in different languages, which shed light on how language and translation can influence interaction fluency and perceived ease of use. Two sessions were conducted in English and three in German (with the chatbot's output auto-translated by the firefox auto translate plugin into German for two of those users). The bilingual aspect introduced an extra layer of complexity for the German-speaking participants. On the one hand, allowing users to interact in their native language is beneficial for comfort and understanding; on the other hand, the translation of technical terms and the potential mix of languages in the interface can cause minor confusion. Some of the German auto-translated features were not as well translated by the plugin as one would expect. An example of these translation difficulties can be seen in figure 6.2 which shows the original English version compared to the German translation. Despite these challenges in translation, the users who used the auto-

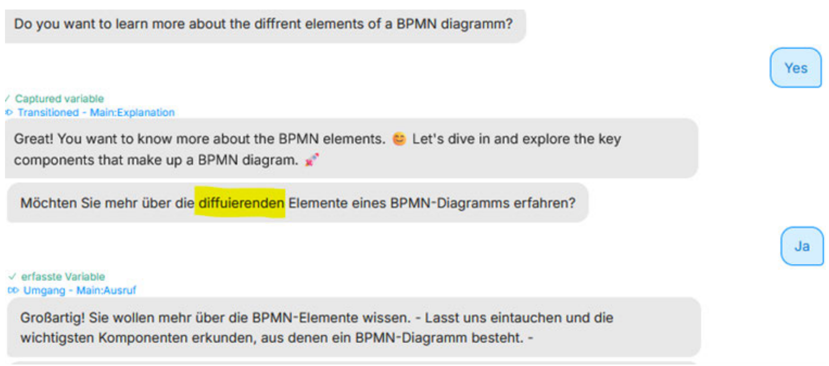

Figure 6.2: Comparison of original English Version and German Translation

translated version of the chatbot were still able to complete the tasks given and understood the meaning of the prompts. This auto-translation could have possibly caused an extra cognitive load for those users and for user-friendliness, a human-translated German version of the chatbot would have probably been preferable. In terms of satisfaction, the German users gave ratings in the same range (4 and 5 out of 5) as the English users, indicating they felt the chatbot was usable in their language. In summary, the use of auto-translation enabled non-English speakers to use the chatbot with success.

6.3.6 Participant Suggestions for Improvement

After completing the tasks, participants were asked for open-ended suggestions to improve the chatbot's user-friendliness.

One aspect that was mentioned most frequently, whether in the open-ended question or throughout the process, was the repetitiveness of the chatbot. Several participants recommended making the conversation less repetitive. One experienced user explicitly said *"reduce the number of repetitive prompts during editing"* (User 2), proposing that the bot consolidate some questions. A second user similarly noted the bot *"repeats itself quite often"* (User 5), indicating this was a generally felt issue. This suggestion ties to the earlier discussion on prompt pacing and was a direct reaction to feeling that certain sequences, like confirming gateway branches or finishing modifications, were overly drawn-out. For a new version of the chatbot, the number of questions may be reduced by, for example, combining the gateway condition and task input into a single question, as suggested by a user and letting the backend handle the additional logic.

Another suggestion was the handling of IDs by the chatbot. As this was brought up in multiple interviews, it becomes clear that the editing mechanism is not as user-friendly as it should be. One novice remarked: *"not entirely clear what I had to input there, whether it's just the number or an ID or something"* (User 3). This feedback directly references the confusion about the ID number. In this case the user also gave a concrete suggestion about what would have been preferable: *"Being able to select an element visually... by just clicking on it... would really help,"* the user said. This suggestion is a good one but, regarding the current tool choice (Botpress), might not be trivial to accomplish. Nevertheless, the interviews showed that the current edit mechanism could be improved, whether by better guidance, clearer prompts or possibly making a tool change.

Another piece of feedback was to adjust the order of input for XOR Gateways. This suggestion, which came specifically from a less experienced user, was to make the input order for decisions more natural. One participant proposed that it would be easier if *"a gateway is created first and then the two possible task parts follow and not first the tasks and then the either-or gateway decision"* (User 4). As mentioned earlier, this could be due to the way the natural language task for the BPMN modelling task was given, where first the condition was mentioned and then the task. It may also be that the logic of *"condition first, then task"* is simply more intuitive. Implementing this might require a larger overhaul of the current Botpress logic.

Another suggestion, mentioned by an experienced user, was to include more BPMN-specific elements such as service tasks. This is an understandable request and would improve the usability and accuracy of the tool. However, the scope of this master's thesis was intentionally narrowed to only include basic BPMN modelling elements (tasks, gateways, nested gateways), since the goal was to provide a functional MVP rather than a complete modelling solution.

Finally, regarding the tutorial, one user noted that the current tutorial is set up to give a complete overview of the basic BPMN elements in one single formatted prompt. This is a lot of information condensed into one interaction and may leave some users feeling overwhelmed. *"God, so much text"* (User 4). This indicates that a more granular approach, for example, splitting the tutorial into multiple prompts or offering tutorial guidance contextually during the interaction process, may be more beneficial for users.

6.3.7 Conclusion

The user testing of the BPMN chatbot provided a comprehensive look at its strengths and pain points, yielding valuable insights for refinement. In summary, the chatbot was able to provide the users with formally correct BPMN diagrams, that described their underlying processes.

Participants, regardless of background, were able to model a non-trivial e-commerce process from scratch, as evidenced by task accuracy and control-flow completeness. This reliability is a major strength, building user trust that the system works. Participants voiced confidence in the final diagrams, noting no steps were missing and the logic was correctly captured.
Alongside these strengths, the evaluation revealed key usability challenges. The most notable issues were related to the interaction flow and some chatbot prompts, rather than the solution's technical correctness. Users encountered friction with the dialog's pacing and mechanics, especially the step-by-step questioning, which sometimes felt tedious and repetitive. Editing via ID input was also not intuitive and led to occasional confusion. These issues caused moments of frustration, as reflected in think-aloud protocols, e.g., *"God, so much text"* (User 4) or *"repeats itself quite often"* (User 5). These instances highlight the need for usability improvements to streamline the user experience.
From these findings, several future improvement opportunities emerge, some of which are easier to implement than others. A more iterative tutorial could break down the current single prompt into multiple smaller chunks of BPMN guidance. Certain prompts, such as those for XOR Gateways, could be merged to reduce repetition. Enhancing the editing experience is also a concrete improvement with high impact; allowing direct interaction with the diagram (clicking to select where to insert or dragging-and-dropping steps) would eliminate the awkward ID reference step. It would make the chatbot feel more like a seamless modelling tool and less like a command-line system. Additionally, the logic of the XOR gateway could be refined to prompt the condition before the task. Finally, to ensure consistent and clear translations, especially for multilingual usage, verifying or supplementing auto-translations with human-translated content would help reduce confusion.
In conclusion, the user testing phase has demonstrated that the BPMN chatbot is a promising tool that can democratize process modelling by guiding users through BPMN diagram creation conversationally. Its core functionality is sound and users responded positively to being able to produce a correct diagram with relative ease. At the same time, addressing the aforementioned pain points will be crucial for improving the overall user experience. For the MVP version of this master's thesis, the suggestions of a more iterative tutorial and improved branching logic for the XOR Gateway were implemented, as these could be integrated without fundamentally changing the underlying chatbot framework.

7 Discussion

In this chapter, the design decisions,positive outcomes and limitations of the chatbot-assisted BPMN modelling approach are critically examined. The discussion covers the rationale behind the chosen rule-based architecture and platform, reflects on the system's successes in lowering barriers to process modelling and addresses the shortcomings of the chatbot.

7.1 Rationale Behind the Chosen Rule-Based Chatbot Architecture

Early in the development process, various chatbot architectures were assessed, in specific rule-based architecture, generative-based chatbots, transactional and hybrid chatbots were evaluated. The rule based architecture was ultimately chosen due to several key reasons. First and foremost, formal correctness was paramount. In BPMN diagram creation, even small inaccuracies, such as unclosed gateways or misplaced sequence flows, can lead to significant confusion and errors in subsequent stages of process analysis. A rule-based system, built on predefined decision trees and scripting logic, provides determinism: each user input is mapped to a known, validated action, ensuring the resulting BPMN diagram remains syntactically correct. This deterministic nature also helps novice users, who benefit from being guided step-by-step with unambiguous prompts rather than navigating open-ended questions. Secondly, the user demographic further justified the choice. Many target users were expected to be BPMN novices, meaning the system had to reduce ambiguity, enforce best practices, and serve as a built-in tutor. A rule-based chatbot naturally caters to these needs by breaking down the modelling procedure into clear, logical steps. This structure alleviates some of the cognitive load on beginners. Rather than worrying about BPMN syntax and semantics, they can focus on describing their process, confident the system will enforce correctness.

Another key factor for choosing the rule based architecture, was the risk of unpredictable or contextually inappropriate outputs that generative models sometimes produce. Especially at the time of the project's inception, large language models were not proven to consistently generate complex process structures (like BPMN diagrams) accurately. While generative systems excel at producing natural-sounding text, they can also introduce 'hallucinations' or guesswork. For a process as dependent on formal correctness, this could have led to potentially placing gateways, tasks or events in unintended ways. By contrast, a rule-based design operates within strict constraints tailored to BPMN, making it much easier to prevent or intercept formal errors before they appear in the final output.
Additionally, practical considerations also played a role in the decision. Training a custom model to reliably handle BPMN modelling would require a domain-specific dataset featuring thousands of dialogues and example diagram. This undertaking that was not feasible within the given scope and timeframe. Off-the-shelf generative models also lacked robust, fine-grained BPMN knowledge, meaning any conversational logic they provided could still yield invalid or incomplete diagrams. In addition to that pre-trained models can be costly if multiple requests are put towards the selected model over a larger time frame and this was not feasible either. Thus, building a rule-based system leveraging explicit BPMN rules and design patterns was deemed the most direct way to ensure correct, on-spec process diagrams from day one.
Beyond these core factors, one additional benefit that emerged was the architecture would be

easily amendable and extendable for future change. By leaning on deterministic rules, subsequent feature expansions, such as advanced BPMN constructs, become more straightforward to integrate and can be integrated into the existing logical framework with relative ease. This modular design underscores the chatbot's potential for long-term adaptability, ensuring it can grow alongside organizational or industry changes.
Transactional chatbots, though highly effective for automating brief, well-defined tasks, like as booking appointments or processing simple queries, offer limited scope when it comes to the iterative complexity of BPMN modelling. BPMN diagrams typically require users to input multiple steps and process descriptions can become quite complex. Transactional bots, by contrast, are designed to complete a single, bounded exchange, providing minimal support or allowing only a very limited number of outcomes. They excel in scenarios where the conversation can be terminated once the transaction is completed (e.g., a purchase or a reservation), but lack mechanisms for guiding users through nuanced workflows. Consequently, transactional chatbots were not a suitable choice for this project, where creating an accurate, multifaceted BPMN diagram demanded a step-by-step, detail-oriented interaction flow rather than a short transactional exchange. Transactional chatbots are also defined by their close interrelatedness to a database or internal business application. This is also not the case for the proposed chatbot.

While the rule-based architecture provided control and precision, it inherently sacrificed flexibility and natural language richness. In retrospect, a hybrid architecture combining rule-based logic with generative, AI-driven components could offer a better balance between determinism and user experience. A hybrid chatbot leverages the strengths of both approaches. The rule-based component can handle the structured, critical parts of the dialogue ensuring, for example, that all mandatory BPMN elements are collected and syntactically valid, while a generative component could manage more open-ended inputs or unforeseen user queries and could offer variability when interacting with the chatbot. This kind of architecture might have improved the chatbot's ability to handle phrasing variability and follow-up questions that were not explicitly anticipated in the rule set and could have maybe given users help when they needed it. For instance, if a user phrased a process step in an unexpected way or asked a clarification question, an AI module could interpret the intent or provide a helpful answer rather than forcing the conversation down a fixed path. The project's current rule-based bot does not allow deviations outside the happy path, as it only gives users multiple choice options what they can do next. When a user strays from the happy path and regardless of input selection puts an unfitting question or answer, the bot does not know how to handle that unexpected input. In a hybrid scenario, the generative NLP could step in to interpret user's inputs that fall through the rule-based cracks, making the interaction more forgiving and conversational. Scenarios where a hybrid approach would be especially beneficial include handling questions by the user and guiding the user through the change process of the BPMN diagram. This could also help with the user suggestion of grouping multiple steps into one step, so the chatbot does not feel as repetitive. In essence, a hybrid design could maintain the formal correctness afforded by rules while injecting the flexibility of an AI model to improve user comfort. Prior research and industry solutions have noted that such hybrid chatbots can cover routine structured interactions with scripted flows and simultaneously tackle complex or context-based queries using AI (Adamopoulou and Moussiades, 2020).

7.2 Assessment of the Implementation Platform Botpress

The implementation platform for the chatbot was Botpress, a framework specifically geared towards building chatbots (Botpress, 2024). The decision to use Botpress was influenced by several of its strengths in terms of user interface and architecture. First, Botpress provides an intuitive visual conversation flow editor that allowed the rapid development of a complex dialogue without extensive low-level coding. This graphical interface was critical and user friendly for designing the rule-based flows of the chatbot. It made it easier to map out the conversation as a series of nodes and transitions corresponding to BPMN modelling steps. The platform's modular design was another key factor. Botpress comes with a suite of pre-built cards for common chatbot functions (sending messages, capturing user input, branching logic, calling APIs, etc.), which could be assembled to implement the desired functionality, without the user having to manually set up that logic.

This modularity aligned well with the needs of the project, as the BPMN chatbot required integrating multiple functions such as guiding the dialogue, validating inputs, storing intermediate data and making http requests. Additionally, it includes a built-in database and supports custom code execution through action cards. The user interface provided by Botpress (a web chat widget) meant that a basic front-end for user interaction was readily available and easily deployable in a browser. This saved development effort on building a chat UI from scratch. Another practical reason for choosing Botpress was its cost and community. The platform is available as a free open-source offering and it has extensive documentation and community support.

Despite the advantages of Botpress, using the platform also introduced several challenges that became apparent during implementation and testing. One significant disadvantage was that Botpress exhibited significant UI issues that were not apparent when using the built-in emulator. For instance, questions sometimes were asked multiple times by the chatbot, images failed to render properly in the browser and various minor errors occurred unpredictably in the live interface, contrasting sharply with the smoother emulator experience. This disconnect made the system feel unreliable and led me to the decision to only use the emulator for user testing, as the UI was such a significant drawback and would have left users very dissatisfied. As the goal of this thesis was not to test the reliability of the Botpress interface, but rather to get a sense if users of varying experience levels could create a chatbot with the created Bot2BPMN, I decided to not disrupt the user experience by using the actual UI but focused the interviews on the logic of the bot itself and used the emulator for that reason.

In addition to these UI complications, Botpress imposes a rigid arrangement of main flows and subflows. Specifically, subflows can only rejoin the main flow at its starting point, rather than resuming at arbitrary moments in the conversation. This restriction becomes problematic in scenarios where the user might need to revisit or refine a BPMN diagram after it has been generated. Under the current structure, such re-entry is limited to the main flow's initial nodes, preventing a seamless or more context-aware conversation that could adjust mid-stream. For a system intended to iteratively guide users through complex process modelling, this inflexibility hampers the chatbot's ability to accommodate non-linear user journeys.

The identified UI and flow-related challenges underscore that platform choice plays a crucial role in the success of this project. Although Botpress offers a fast setup and a straightforward interface for rule-based development, the unforeseen issues with the UI and the rigid subflow structure highlight limitations that can significantly affect the user experience. In retrospect a more comprehensive research into alternative chatbot frameworks or a paid version of Botpress, might have led to a more balanced solution, combining ease of development with higher usability. Additionally, I should have tested the UI more thoroughly before developing the chatbot on the platform, as to spot UI issues earlier. During development I mainly used the emulator as it was easier and I assumed the UI provided by the platform would be robust.

Going forward, developers looking to build a similar chatbot-driven BPMN system should keep these constraints in mind, particularly around supporting non-linear dialogues and ensuring a robust UI to handle iterative, user-driven process modelling.

7.3 Focus on core BPMN elements as a limitation

Beyond platform and architecture, another strategic limitation was the narrow scope of BPMN elements supported by the chatbot. The implementation of the Bot2BPMN was restricted to the core BPMN components. This scoping was an intentional constraint, introduced to manage complexity and ensure that development and testing could focus on the structural integrity and user interaction with the essential building blocks of BPMN. BPMN is a very complex framework and offers a wide range of modelling constructs, including various types of events, message flows and exception handling mechanisms. Incorporating the full breadth of the BPMN standard would have significantly increased the system's complexity and extended development beyond the available timeframe and would have shifted the focus more onto modelling complexity, rather than a functional MVP. Therefore, the scope was intentionally limited to a minimal, stable set of features, with the clear objective of building a functioning MVP that could validate the core idea, that a chatbot can guide users through the creation of semantically correct BPMN diagrams using structured conversation.
While this limited scope made it possible to validate the concept efficiently, it also imposed limits on the expressiveness and real-world applicability of the tool. More advanced modelling constructs were not included and the advanced user noted this as an improvement in the User interviews. In practical settings, business processes often require a richer modelling vocabulary to represent exception handling, asynchronous communication, subprocess hierarchies or conditional flows. The current implementation, while sufficient for simple or linear workflows, does not yet demonstrate whether the chatbot can support the full range of modelling needs encountered in real organisational contexts. To capture the business process as close to the how it is implemented and practiced in a business is very important at is ensures clarity of how the process workdays and improves communication about the process, as all stakeholders have the same level of detailed knowledge, that the BPMN provides. These additional BPMN elements help convey the nuances of a process in a way that is understandable to both technical and non-technical stakeholders, which is particularly important when BPMN diagrams serve as shared documentation or as the basis for automation.
As a result, although the system proved successful in guiding users through core modelling tasks, its effectiveness in supporting more complex scenarios remains untested. This limits the conclusions that can be drawn about the tool's readiness for deployment in enterprise environments where BPMN is used not only for documentation but also for communication between stakeholders. Bridging this gap will require extending the chatbot to accommodate a broader set of BPMN features and verifying that the conversational flow remains intuitive and manageable even as modelling complexity increases. Until such expansions are implemented and evaluated, the current system should be seen primarily as a prototype for educational or low-complexity use cases, rather than a fully general-purpose BPMN modelling solution.

7.4 Positive Outcomes and Successes of the Bot2BPMN Chatbot

Despite the challenges, the project produced several positive outcomes, demonstrating the value and feasibility of a chatbot-assisted approach to BPMN modelling. These successes validate the core idea and provide a foundation to build upon.

One of the most significant achievements is that the chatbot made BPMN process modelling accessible to users with little to no prior knowledge of the notation. By walking users through the modelling task step-by-step in natural language, the system guided novices in constructing a process diagram without requiring them to understand the formal diagramming language upfront. During user testing, even participants with no BPMN experience were able to model a non-trivial process (an e-commerce scenario) from scratch with the chatbot's help. This suggests that the interactive dialogue, combined with embedded explanations of BPMN concepts, successfully democratized process modelling for beginners. In traditional BPMN tools, novices often struggle with where to start and how to apply the notation correctly. This barrier to entry seems to be lowered by the conversational interface of the chatbot, the modelling by breaking down the whole business process into small steps and by the users non having to worry about the notation, but that they can just focus on accurately describing the process. Lowering the barrier of entry for novice or inexperienced users was a core motivation for this thesis and the user tests proved, preliminary, that inexperienced user can successfully generate semantically correct and complete BPMN diagrams that describe the underlying business process well.

Another positive element was the sense of user empowerment reported in the user tests. Several testers found it rewarding that they could drive the generation of a formal process diagram with minimal external assistance. This perceived autonomy can enhance adoption in real-world organizational settings, as users are more likely to integrate the tool in daily workflows if they feel capable of creating valid outputs themselves (Huang and Rust, 2018). In turn, such acceptance can accelerate BPM initiatives, as the bottleneck of scarce modelling expertise becomes less limiting.

In addition to lowering the barriers of entry, the project demonstrated that it is technically feasible to generate valid BPMN diagrams from a structured chatbot dialogue. In all testing sessions, the bot was able to successfully produce a BPMN diagram that corresponded to the user's described process by the end of the conversation, even though the business process was not always reliably recounted by the users. This is not a direct issue with the chatbot itself but stems from a user error, that may be corrected in future versions of the chatbot, by for example asking clarification questions or providing more support during the modelling process. The fact that participants voiced confidence in the final diagrams, noting that no steps were missing and the logic was correctly represented, attests to the viability of this approach in ensuring completeness and correctness. In essence, the core hypothesis of the thesis was confirmed: a chatbot can indeed assist users in creating a BPMN diagram with core BPMN elements that is both accurate and useful.

Finally, this thesis contributes to a broader academic conversation on how chatbots can serve more specialized, discipline-specific tasks as opposed to merely handling generic Q&A or simple transactional requests (Klievtsova, Benzin, Kampik, et al., 2023). Specifically, by integrating rule-based logic with explicit BPMN rules, the thesis demonstrates that even novice users, with little or no prior knowledge of BPMN process modelling, can be guided through the creation of BPMN diagrams step-by-step. In doing so, this work not only aligns with existing research on conversational BPMN assistants such as BPMN-GPT (YesChat.AI, 2024) or DeCLO (Alman et al., 2020) but also contributes new evidence on the effectiveness of structured, deterministic chatbot flows in lowering entry barriers for beginners. Compared to other chatbot-driven modelling tools such as SOCIO (Sara Pérez-Soler, 2018) or Coral (Barón-Espitia, Dumas, and González-Rojas, 2022), which emphasize diagram creation or process simulation, this thesis specifically focuses on ensuring strict BPMN syntax control while simultaneously guiding novices through step-by-step tutorials. This combination of formal correctness and structured instruction differentiates the present work by addressing both the technical demands of BPMN and the educational needs of new users.

7.5 Error Handling and Robustness Challenges

While the chatbot successfully guides users through BPMN modelling, it also exhibits notable technical limitations that impact its robustness. One major issue is the rigidity of its scripted conversation flow. Each dialogue step and response is predefined, which means the system handles expected inputs well but struggles with any deviation or unexpected user behaviour. For instance, if a user provides information out of the expected order or uses phrasing not anticipated in the flow, the bot has limited ability to adjust or seek clarification. In the user tests, a novice user inadvertently gave an exclusive gateway condition before specifying the task name, confusing the bot's logic. The system did not intelligently recognize this mix-up and processed the inputs in the fixed order it was programmed for, leading to a mix up between the condition and task in the final diagram (as can be seen in the appendix). This illustrates the lack of context-aware input validation, as the chatbot does not understand the intent behind user inputs and simply processes the text it is given. It cannot yet detect when a user's response, though syntactically valid, is contextually misaligned and then recover from the error or ask additional questions. As a result, the burden falls on the user to notice and correct any mistakes in input, rather than the system handling errors interactively.
Another limitation is in the bot's error handling for invalid inputs. The current implementation does not use any checks beyond the control flow logic of the bot itself. When an error by the user is caught, for example if a user enters something when the bot does not currently expect input, the bot responds by simply repeating the question, without more nuanced guidance. It does not offer suggestions like *"Did you mean step X?"* nor does it adapt by showing the list of valid steps in that moment. This lack of proactive support in error handling can frustrate users who are unsure how to recover from their mistake. In the tests, no user became completely stuck due to these missing error handling responses but think-aloud comments showed mild frustration when the system did not act as they would have expected and did not offer help. The system's inability to dynamically interpret and respond to user confusion underscores its limited robustness. Should a user go off-script or enter information that violates assumptions, the conversation cannot easily self-correct or guide the user back on track beyond reiterating the predefined questions. This in turn leads to frustration on the part of the user and an invalid or incorrect BPMN document, such as this was the case with User 3, who could not make the required change.

Compounding these user-facing issues is the lack of error handling for Botpress-internal errors, that is problems that arise from flaws within the bot's own execution logic or the underlying platform. This includes situations where the bot enters an undefined state, fails to store variables correctly or misroutes the dialogue due to faulty transitions or data handling within Botpress. For example, in multiple instances Botpress produced erroneous responses, such as failing to acknowledge a confirmed input or sending an incomplete HTTP request to the backend via Ngrok. In such cases, the bot does not issue any system-level error messages

or have fallback behaviours. Currently it merely loops back to the last correct previous state, repeats the question or silently fails to produce a correct result.
This type of failure is particularly concerning because it is invisible to the user. Unlike user-triggered errors, which at least provide some hint that something went wrong, Botpress-internal errors may cause the bot to behave incorrectly while still appearing to function normally. The logs in the backend show these errors but do not provide any opportunity for the end user on how to handle them. This can lead to flawed outputs without any indication to the user that an internal issue occurred. Notably, no global error boundary or exception-handling mechanism has been implemented to catch these platform-side issues and guide the user out of them. Thus, even though the dialogue seems to progress, the BPMN logic may deviate from the intended model or users have to repeat their inputs.

The implications of these limitations for the chatbot's overall robustness and trustworthiness are significant. In its current rule-based implementation, the system is only as resilient as the scenarios explicitly anticipated and encoded by the developer. When users remain within the predefined bounds of interaction, the bot performs reliably and produces formally correct BPMN outputs. However, any deviation can lead to silent failure or unusable results. As modelling tasks increase in complexity, these issues are likely to become more frequent and more impactful.
For practical deployment in real-world scenarios, a truly robust assistant must be capable of handling both user-related and internal system errors in a transparent manner. This includes recognizing unexpected or ambiguous inputs and responding with follow-up clarification questions, as well as offering contextual suggestions and help. When a user refers to a task or step using an incorrect or unclear ID, the system should ideally provide a user-friendly list of valid references or confirm potential matches. In addition, the assistant should be equipped with global fallback mechanisms to handle internal errors, such as those caused by faults within the Botpress platform itself. The bot could display diagnostic messages like *"I did not quite catch that. Please repeat your last input"* allowing the user to re-enter their last input while making this process transparent and user-friendly. Furthermore, transparent logging of internal failures, coupled with optional user notifications, would help maintain user trust and allow for better debugging and iterative improvement of the system.
The fundamental challenge facing this rule-based chatbot design is balancing formal correctness with natural interaction flexibility, particularly in structured domains like BPMN modelling. To move beyond the constraints of the current prototype and towards a more robust, production-ready tool, future iterations of the chatbot must incorporate both architectural and conversational enhancements. These may include the integration of more context-aware natural language understanding components or an expanded set of rule-based error recognition and recovery strategies tailored to frequent user mistakes. Additionally, the implementation of global error handling mechanisms for internal failures would significantly improve the chatbot's resilience and transparency. Without these improvements, the chatbot may remain a useful proof of concept or educational prototype, but its application as a dependable assistant for professional BPMN modelling remains limited.

7.6 Limitations of the User Testing Approach

The evaluation of the chatbot was conducted with a small sample that was partly chosen due to their different levels of expertise with BPMN and party for availability reasons. This limited number and the choice of participants introduces several limitations to the findings. Only

five user sessions were carried out, which is a modest number sufficient for exploratory usability insights but too small to generalize performance results. Although the participants represented a relatively diverse range of ages (28 to 72 years) and educational backgrounds, including two individuals without a university degree, one with a bachelor's degree working in a therapeutic and artistic field and two with related academic backgrounds, the sample was nonetheless convenience-based. A unifying factor was that all participants were personally acquainted with the researcher, which may have introduced unintentional bias in their feedback, potentially leading them to view the system more favourably or to hesitate in expressing negative impressions candidly. This potential bias, when combined with the small sample size, constrains the extent to which the results can be considered representative of the experiences and evaluations of a broader, less personally connected group of users.
To strengthen the robustness of future assessments, a larger and more diverse participant pool should be sought, ideally including users from various professional domains, digital literacy levels and with no prior connection to the researcher. This would help ensure that both the observed strengths, such as task completion and the encountered weaknesses, like the rigidity of the chatbot, reflect the broader usability and performance of the tool.

Another notable limitation of the evaluation lies in the design and scope of the modelling task itself. Participants were asked to model a single, relatively short process: an e-commerce order fulfilment flow that involved several tasks and one exclusive gateway. While this task was sufficient to demonstrate the basic functionalities of the chatbot and a structurally sound BPMN diagram could be generated from it, the task was intentionally kept short and straightforward to avoid overburdening participants and to ensure the session remained within a manageable timeframe. However, this decision also likely influenced the overall assessment of the user experience. Specifically, the rigid and repetitive nature of the chatbot's interactions, which was noted by some participants. This issue may have been exacerbated if the participants would have needed to model a longer process with more steps. As the chatbot asks users to proceed step by step with very specific responses, even minor inefficiencies or restrictions in the dialogue flow could feel increasingly burdensome over time. One repetition of a specific step may not be perceived as a burden by users but should that repetition compound, the user might be much more frustrated by it. Therefore, the brevity of the modelling task may have masked potential usability issues that would emerge in more realistic, complex modelling scenarios. Future evaluations should therefore include more complex and variable process tasks to uncover how well the chatbot scales in terms of interaction design, error tolerance and user satisfaction.

Another important limitation concerns the timing of the user feedback. In this study, user testing was conducted only at the end of the development process, once a functional prototype had already been implemented. While this allowed participants to interact with a working system, it also meant that their feedback could no longer influence the foundational design of the chatbot. As a result, insights regarding interaction flow, feature expectations or usability pain points could only be documented retrospectively rather than actively addressed during development. A more iterative approach, incorporating earlier stages of user involvement through low-fidelity prototypes such as wireframes or mock dialogues, would likely have yielded more actionable insights and allowed for course corrections throughout the development process. In future iterations, early-stage usability testing with rough prototypes should be conducted to ensure that the chatbot design better reflects the needs and expectations of its intended users. This would enhance not only the final usability of the system but also its relevance and acceptance across different user groups.

Another key limitation lies in the KPIs selected for evaluating the chatbot's success. The KPIs were initially intended to offer quantitative insight into the chatbot's performance. However, in practice, they proved to be of limited value. While it was expected that these measures would help evaluate both the technical correctness of the output and the efficiency of the interaction, they ultimately did not provide meaningful differentiation between participants or deeper insights into the quality of the user experience.
Task accuracy, defined as the correct representation of the described tasks in the BPMN diagram, turned out to be uninformative. All five participants achieved a perfect score in this regard, which rendered the metric ineffective for distinguishing between sessions or highlighting usability challenges. The absence of variation was largely due to the simplicity of the modelling task and the chatbot's structured guidance, which left little room for task-level inaccuracies and effectively eliminated naming errors by structurally aligning BPMN tasks closely with the user's input. Instead, the user question that was posed during the interviews, *"The chatbot's prompts were clear and easy to understand"* provided more meaningful insight into how well users could comprehend and work with the system, revealing satisfaction differences that the task accuracy metric could not capture.
Conversation length, measured in steps initiated by the chatbot, was similarly unhelpful. Due to the chatbot's rigid and predefined structure, the number of steps was largely determined by the bot's internal logic rather than by user behaviour. In cases where a conversation was longer, this was typically due either to a deviation from the expected input pattern by the user or to necessary confirmation steps built into the dialogue flow. As a result, a longer conversation did not necessarily reflect confusion or inefficiency, nor did a shorter one guarantee a better user experience. The metric lacked the sensitivity needed to reflect qualitative aspects such as clarity, satisfaction or cognitive effort.
The only KPI that provided genuine insight was control flow coverage. This metric assessed whether all required BPMN elements that were input by the user, were displayed in the final generated BPMN diagram. It was the most useful of the three, as it confirmed that all participants succeeded in producing formally correct and complete BPMN models. This finding validated the chatbot's core functionality and its ability to guide users through a process that results in syntactically valid outputs.
In summary, while the selected KPIs appeared appropriate at the outset, only control flow coverage delivered meaningful evaluative value. The experience highlights the need for a more thoughtful combination of quantitative and qualitative metrics in future evaluations—ones that can capture not just the correctness of the result but also the quality and usability of the interaction from the user's perspective.
An additional limitation of the user testing arises from the way the chatbot's editing functionality was evaluated. Due to a technical constraint within the control flow logic of Botpress, it was not possible to support multiple iterative modifications to the process model during a single conversation. As a result, participants were only asked to make one change, which was to insert one task into the existing BPMN structure, after completing the initial modelling phase. This decision was made to align the evaluation with the chatbot's current capabilities, but it introduced a degree of artificial simplicity into the test scenario. By restricting users to a single change, the evaluation did not reflect the reality of process modelling as it typically occurs in practice, where diagrams are refined iteratively and often undergo several rounds of modification. The editing experience, evaluated in isolation and in a simplified form, was therefore likely perceived more positively than it might have been in a more realistic, iterative

use case. While it allowed the system's basic functionality to be demonstrated and evaluated within technical constraints, it also prevented users from realising this constraint with the chatbot and its limited use in real world scenarios. Future versions of the chatbot should fully support iterative editing to better reflect the dynamic and exploratory nature of real-world process modelling. The current limitation of allowing only a single change per session falls short of professional users' expectations and typical workflows. Supporting multiple edits would not only improve usability and robustness but also enable more realistic user testing, where participants are asked to make a series of changes rather than just one.

In summary, the BPMN chatbot evaluation had significant methodological limitations. The study used a small, convenience-based sample of five participants who were personally acquainted with the researcher, potentially introducing bias. The test task was intentionally simplified, which may have masked usability issues that would emerge in more complex, real-world scenarios. Most of the selected KPIs proved uninformative, with only control flow coverage providing meaningful insight. Finally, the testing only allowed for a single model modification rather than the iterative refinement process typical in actual BPMN modelling, creating an artificially simplified evaluation that likely overestimated the system's practical usability.

7.7 Ethical and Human Factors Considerations

Beyond its technical performance, the chatbot introduces several ethical and human-centric considerations that are important for its long-term usability and acceptance. One critical aspect is the step-by-step conversational style, may influence how users think about process modelling. While this guided interaction lowers the immediate cognitive load by breaking tasks into manageable pieces, it can also narrow the user's focus to local elements (Vogel-Walcutt et al., 2022). While this can help novice modelers avoid feeling overwhelmed, it might inadvertently encourage a sequential "tunnel vision" approach. As a result, users may neglect holistic process design or overlook elements that span across different teams, which would be more apparent in a traditional, full-diagram view. In a conventional modelling scenario, an experienced human modeler might pause to consider these broader aspects independently and thus create a full-scale view of the process rather than a narrower scope. By contrast, a chatbot-led session could omit such context unless explicitly programmed to address it. This raises the concern that the convenience of stepwise guidance comes at the expense of comprehensive process thinking. Designers of such systems must strive to counteract this by periodically encouraging users to review the entire model or by building in prompts that elicit global considerations, ensuring that the forest is not lost for the trees.

The user test could not reveal whether this narrow scope led to the omission of certain aspects of a holistic and cross-department business process, as the users were given a limited task, with all the process steps already defined for them. Therefore, they did not need to adopt an eagle-eye perspective or think of the various possibilities and interdependencies that occur in open-ended modelling tasks.

User trust and reliance constitute another crucial consideration. A well-performing conversational assistant can engender high trust, meaning that users may begin to rely on the system's recommendations without independent verification. However, an over-trusting user is vulnerable to automation bias, accepting the assistant's BPMN outputs even if subtle errors or suboptimal modelling choices are present. If trust is too high relative to performance, over-reliance and user complacency can result, wherein the user might blindly accept a generated diagram as correct. The chatbot's friendly, dialog-based interaction style might even enhance the sense of an infallible guide, further tempting users to cede their critical oversight. On the other hand, if the assistant makes mistakes or fails to understand inputs, user trust can quickly erode. A user who encounters repeated errors or confusing behaviour might lose confidence in the tool and disengage from using it altogether. This miscalibration of trust—either too much or too little—is problematic. Over-trust can lead to erroneous models going unchecked, while under-trust means the benefits of the assistant, such as guidance and efficiency, are not fully realised. From a system design perspective, ensuring that users maintain an appropriate level of trust in the chatbot is important. This can be fostered by the system communicating its uncertainties or limitations candidly, helping users to adjust expectations and by offering clear explanations for its suggestions. Achieving this balance aligns with the HCI principle of trust calibration, which calls for the user's trust to match the system's actual capabilities (Carragher, Sturman, and Hancock, 2024).

It is important to note, however, that these aforementioned human factors were not systematically explored or measured within the scope of this thesis. The focus of the evaluation was on technical correctness and basic usability, rather than on psychological or behavioural aspects of user interaction. While some of these concerns surfaced anecdotally in user comments and observations, no dedicated user study was conducted to examine how the chatbot influences trust or modelling mindset. This represents a clear limitation of the current work. Future studies should incorporate targeted methods, such as trust rating scales, think-aloud protocols focused on mental effort, or comparative studies against traditional modelling tools, to better understand the broader human implications of working with a conversational assistant. A comprehensive future evaluation that includes human factors would be essential to ensure the chatbot supports a reliable, user-aligned and trustworthy modelling experience.

In summary, design factors such as influencing the user's perspective or establishing appropriate levels of trust are central to evaluating the effectiveness of chatbot-assisted process modelling. While the conversational style offers structure and ease of use, it must be designed thoughtfully to avoid encouraging overly narrow thinking or uncritical reliance on the system. Insights from human-computer interaction, such as the importance of calibrated trust and manageable cognitive load, provide a valuable lens for addressing these concerns. A reflective design approach should therefore include safeguards against automation bias, encourage users to remain actively involved in validating the model and provide support for maintaining an overview of the entire process. In doing so, the chatbot can not only guide users through modelling tasks but also empower them to stay in control and think critically throughout.

7.8 Conclusion of the Discussion

Overall, the discussion highlights the dual nature of the chatbot-assisted BPMN modelling approach. While technically promising and conceptually validated, the current prototype also reveals important limitations that must be addressed before broader deployment. The choice of a rule-based architecture enabled high reliability and formal correctness, which is especially useful for novice users unfamiliar with BPMN. The chatbot successfully lowered entry barriers by offering structured guidance, allowing even non-experts to create syntactically valid and semantically complete process diagrams. This represents a meaningful contribution, particularly in educational or low-complexity business settings. At the same time, the evaluation uncovered several areas requiring improvement. Platform-related issues, especially those tied to the rigidity of Botpress and its flawed UI, hampered usability and constrained the system's capacity for non-linear or iterative modelling. Error handling remained basic, both at the user input level and within the bot's internal execution, which occasionally resulted in silent failures or confusing behaviour. Moreover, the chatbot's limited support for advanced BPMN constructs restricted its applicability to simple scenarios, limiting its usefulness in complex real-world contexts. The user testing, while supportive of the system's core functionality, was limited in scale and depth. A small, convenience-based sample, combined with a fixed and simplified modelling task, means the findings cannot be generalized without caution. In addition, many important human factors, such as how the conversational structure shapes user thinking or how trust and responsibility are managed, were not systematically studied. Taken together, the current implementation provides a strong foundation and proof of concept but also points clearly toward the need for more adaptive, resilient and user-aware system design.

7.9 Outlook

Building on the results of this thesis, there are a number of encouraging directions to broaden and improve the applicability of the chatbot-assisted BPMN modelling approach. The current rule-based prototype showed that a chatbot can help non-experts create valid BPMN diagrams, but it also showed some inherent limitations, especially with regard to flexibility, robustness and the range of BPMN elements supported. This last section discusses real-world application areas, suggests enhancements that could further transform the current system into a complete modelling assistant and outlines possible avenues for future research.

Future Improvements: Adding more BPMN components is an obvious next step. In order to preserve clarity and guarantee functional correctness, this thesis purposefully concentrated on a core set of modelling constructs. Adding more sophisticated components that are used in many real-world business processes would greatly enhance the chatbots functionality and this would enable it to support more intricate and enterprise-scale modelling use cases.
Future versions of the chatbot could expand its adaptive dialogue strategies to better match the user's level of expertise. This fundamental adaptability could be improved, even though the current prototype already provides a skippable tutorial to assist less experienced users in getting started and let advanced users continue straight to the modelling. The chatbot could offer more specialised assistance by dynamically modifying the amount of direction and explanation given during the interaction, based on the user's responses or self-reported experience. For example, it could identify when a user requires more explanation or provide more complex shortcuts for more seasoned users. These improvements would further solidify the chatbot's position as an adaptable and inclusive modelling assistant while also making it easier to use across a range of user groups.

Furthermore, switching from a chatbot that is solely rule-based to one that is hybrid or generative may improve the interaction's adaptability and naturalness. Rule-based components can continue to enforce the structural correctness of BPMN, while generative AI models such as GPT-3.5 or GPT-4 can interpret free-form user input and react adaptively to unexpected inputs. This component can also help guide users who need help with the chatbot and act as a tutorial during process modelling. The viability of employing large language models to produce BPMN diagrams via intermediary schemas has already been shown by early-stage concepts like GPT-Codegen (RobJenks, 2022). Without sacrificing the syntactic and semantic integrity of the final diagrams, a hybrid design could leverage the advantages of both paradigms and allow for more natural, error-tolerant conversations. Multilingual support is

another crucial area that needs improvement. The current system was only partially tested in German and was primarily designed for English input. But during user testing, depending solely on automated translations proved problematic, as the automatic translation was not very reliable. Future iterations ought to incorporate more complex multilingual NLP pipelines or native-language conversation flows, starting with German. This would lower barriers for non-English speaking organisations, especially in DACH and other BPMN-adopting regions and make the chatbot more accessible to a larger range of users.
The chatbot should also have more sophisticated error-handling features to increase the robustness of interactions. The current prototype makes the assumption that inputs will be given in the expected formats and in the predetermined order. When users make mistakes or stray from the optimal flow, this rigidity presents difficulties. To mitigate this, future versions of the chatbot should implement more advanced dialog management techniques. Features such as context-aware slot filling, which allows the system to capture and store information across different turns in the conversation regardless of sequence, can improve the handling of out-of-order or fragmented input. In the same way, intent recognition would allow the chatbot to identify the user's goal, even if the input doesn't follow the standard format, enabling the system to modify the conversation accordingly. Instead of leading to dead ends, causing missing or faulty outputs, fallback techniques like suggestions or clarification prompts can help users get back on track. These methods would help create a more responsive and dynamic system that can accommodate a wider variety of user input methods and behaviours. These issues may already be party addressed if the architecture would be a hybrid and not a solely rule based chatbot.
Furthermore, the chatbot should enable users to go back and edit earlier steps, such changing a task that has already been defined or removing the most recent addition during the initial modelling phase. This would be more in line with how people naturally behave, as concepts or the users understanding of a process frequently changes while being modelled. The bot would be better suited for dynamic and cooperative modelling scenarios if it could manage such backwards navigation or real-time editing commands.

Along with these functional restrictions, the unreliability of the Botpress user interface was another significant problem found during testing. Duplicate questions, dropped inputs and failure to display the generated BPMN diagram were among the frequent errors and inconsistencies that plagued the frontend chat component. These usability errors should be avoided in future iterations of the chatbot. Because of its flow-based architecture and restricted support for conditional navigation or reactivity, Botpress itself also added to the system's vulnerability. Due to the way Botpress manages context persistence and session state it was not possible to implement iterability for example for the modification flow. For future development another platform that builds chatbots should be selected or maybe Botpress can improve in these areas and may still be a viable option.

Improving the usability, dependability and practicality of chatbot-assisted BPMN modelling will require tackling both the internal logic of the chatbot and the external system constraints imposed by Botpress. In addition to those improvements one could also look to extend the functionality of the chatbot, by providing a deeper integration with AI-powered BPM tools such as process mining and automatic verification. This could elevate the chatbot's capabilities beyond modelling support. For instance, once a BPMN diagram is generated, the chatbot could provide suggestions for optimization, detect potential inefficiencies or validate model compliance against domain-specific standards. Incorporating best-practice templates or drawing on industry-specific knowledge bases would transform the chatbot into a context-aware process advisor—capable not only of translating ideas into formal models, but also of improving their quality and alignment with operational goals.

Bibliography

Abd-Alrazaq, A. et al. (2020). “Technical Metrics Used to Evaluate Health Care Chatbots: Scoping Review”. In: *Journal of Medical Internet Research* 22.6, e18301. DOI: 10.2196/18301.

Adam, M., M. Wessel, and A. Benlian (2020). “AI-based chatbots in customer service and their effects on user compliance”. In: *Electronic Markets* 31.2, pp. 427–445. DOI: 10.1007/s12525-020-00414-7.

Adamopoulou, E. and L. Moussiades (2020). “Chatbots: History, Technology, and Applications”. In: *Machine Learning with Applications* 2, p. 100006. DOI: 10.1016/j.mlwa.2020.100006.

Albert, William and Thomas Tullis (2013). *Measuring the User Experience*. Morgan Kaufmann.

Alkhatib, A., R. Jo, and R. Albustanji (2024). *A Proposed Technique for Business Process Modeling Diagram using Natural Language Processing*. Retrieved from ResearchGate. URL: https://www.researchgate.net/publication/379270223_A_Proposed_Technique_for_Business_Process_Modeling_Diagram_using_Natural_Language_Processing_1_st_Shorouq_Elmanaseer.

Alman, A. et al. (2020). “DeCLO: A Chatbot for User-Friendly Specification of Declarative Process Models”. In: *CEUR Workshop Proceedings*. Vol. 2673, pp. 116–129. URL: https://ceur-ws.org/Vol-2673/paperDR12.pdf.

Asllani, A. (2024). “Artificial Intelligence for Generation and Verification of BPMN Diagrams”. Master’s thesis. Polytechnic University of Turin. URL: https://webthesis.biblio.polito.it/secure/31175/1/tesi.pdf.

Bangor, Aaron, Philip T Kortum, and James T Miller (2009). “Determining what individual SUS questions are important”. In: *Journal of Usability Studies* 4.3, pp. 114–129.

Barón-Espitia, D., M. Dumas, and O. González-Rojas (2022). “Coral: Conversational What-If Process Analysis”. In: *CEUR Workshop Proceedings*. Vol. 3299, pp. 317–330. URL: https://ceur-ws.org/Vol-3299/Paper25.pdf.

Bender, E. M. et al. (2021). “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623. DOI: 10.1145/3442188.3445922.

Botpress (2024). *The Complete AI Agent Platform*. URL: https://botpress.com/de.

Brocke, J. V. and M. Rosemann (2010). *Handbook on Business Process Management 2*. Springer eBooks. DOI: 10.1007/978-3-642-01982-1.

Caldarini, G., S. Jaf, and K. Mcgarry (2022). “A Literature Survey of Recent Advances in Chatbots”. In: *Information* 13.1, p. 41. DOI: 10.3390/info13010041.

Camargo, M. et al. (2023). “Learning Business Process Simulation Models: A Hybrid Process Mining and Deep Learning Approach”. In: *Information Systems* 117, p. 102248. DOI: 10.1016/j.is.2023.102248.

Carragher, D. J., D. Sturman, and P. J. B. Hancock (2024). “Trust in automation and the accuracy of human–algorithm teams performing one-to-one face matching tasks”. In: *Cognitive Research: Principles and Implications* 9.1. DOI: 10.1186/s41235-024-00564-8.

Colby, Kenneth Mark, Sylvia Weber, and Franklin Dennis Hilf (1971). “Artificial Paranoia”. In: *Artificial Intelligence* 2.1, pp. 1–25.

Compagnucci, I. et al. (2023). “Trends on the Usage of BPMN 2.0 from Publicly Available Repositories”. In: *Lecture Notes in Business Information Processing*. Springer, pp. 84–99. DOI: 10.1007/978-3-030-87205-2_6.

Dahlbäck, Nils, Arne Jönsson, and Lars Ahrenberg (1993). "Wizard of Oz studies: Why and how". In: *Knowledge-Based Systems* 6.4, pp. 258–266.

De Almeida Bordignon, A. C. (2016). *A Systematic Literature Review on Natural Language Processing in Business Process Identification and Modeling*. URL: `https://www.semanticscholar.org/paper/A-systematic-literature-review-on-natural-language-Bordignon/bb8f27dae0a520ab1ac68717b1a0fbd936b5873d`.

De Weerdt, J., S. K. L. M. Vanden Broucke, and F. Caron (2015). "Bidimensional Process Discovery for Mining BPMN Models". In: *Lecture Notes in Business Information Processing*, pp. 529–540. DOI: `10.1007/978-3-319-15895-2_45`.

Decker, G. and J. Mendling (2008). "Instantiation Semantics for Process Models". In: *Lecture Notes in Computer Science*. Springer, pp. 164–179. DOI: `10.1007/978-3-540-85758-7_14`.

Deng, Alex et al. (2017). "Trustworthy analysis of online A/B tests: Pitfalls, challenges and solutions". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 641–649.

Dilmegani, C. (2025). *Top 25 Metrics for Chatbot Analytics in 2025*. Accessed: 2025-04-14. URL: `https://research.aimultiple.com/chatbot-analytics/`.

Dohrn, F. (2022). *Speech-to-Process: Von Sprache zum Geschäftsprozess in Python*. Retrieved from GitHub. URL: `https://github.com/bitnulleins/speech2process/raw/main/paper_speech2process.pdf`.

Dumas, M. et al. (2018). *Fundamentals of Business Process Management*. Springer. DOI: `10.1007/978-3-662-56509-4`.

Ericsson, K Anders and Herbert A Simon (1984). *Protocol analysis: Verbal reports as data*. MIT Press.

Ferreira, R. C. B., L. H. Thom, and M. Fantinato (2017). "A Semi-Automatic Approach to Identify Business Process Elements in Natural Language Texts". In: *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS)*. Vol. 2. Scitepress, pp. 250–261. DOI: `10.5220/0006305902500261`.

Figl, K., A. Koschmider, and S. Kriglstein (2013). "Visualising process model hierarchies". In: *European Conference on Information Systems*, p. 180. URL: `http://nm.wu-wien.ac.at/research/publications/b1024.pdf`.

Figl, K., J. Mendling, and M. Strembeck (2013). "The Influence of Notational Deficiencies on Process Model Comprehension". In: *Journal of the Association for Information Systems* 14.6, pp. 312–338. DOI: `10.17705/1jais.00335`.

Følstad, A. and P. B. Brandtzæg (2017). "Chatbots and the New World of HCI". In: *Interactions* 24.4, pp. 38–42. DOI: `10.1145/3085558`.

Følstad, Asbjørn and Petter Bae Brandtzaeg (2018). "Chatbots for customer service: user experience and motivation". In: *Proceedings of the 1st International Conference on Conversational User Interfaces*, pp. 1–9.

Friedrich, F., J. Mendling, and F. Puhlmann (2011). "Process Model Generation from Natural Language Text". In: *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pp. 482–496. DOI: `10.1007/978-3-642-21640-4_36`.

Gartner (2022). *Gartner Predicts Conversational AI Will Reduce Contact Center Agent Labor Costs by $80 Billion By 2026*. URL: `https://www.gartner.com/en/newsroom/press-releases/2022-08-31-gartner-predicts-conversational-ai-will-reduce-contac`.

Graphviz (n.d.). *Graphviz*. `https://graphviz.org/`. Accessed January 14, 2025.

Grassi, L., C. T. Recchiuto, and A. Sgorbissa (2022). "Knowledge-Grounded Dialogue Flow Management for Social Robots and Conversational Agents". In: *International Journal of Social Robotics* 14.5, pp. 1273–1293. DOI: 10.1007/s12369-022-00868-z.

Gururangan, S. et al. (2020). "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8354. DOI: 10.18653/v1/2020.acl-main.740.

Hammer, M. (2014). "What is Business Process Management?" In: *Springer eBooks*, pp. 3–16. DOI: 10.1007/978-3-642-45100-3_1.

He, L. et al. (2024). "Exploring User Engagement Through an Interaction Lens: What Textual Cues Can Tell Us about Human-Chatbot Interactions". In: *Proceedings of the 6th Conference on ACM Conversational User Interfaces, CUI 2024*, Article 9. DOI: 10.1145/3640794.3665536. URL: https://doi.org/10.1145/3640794.3665536.

Holmqvist, Kenneth et al. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press.

Howard, Sarah and Carl Gutwin (2022). "Longitudinal studies in human-computer interaction: Challenges and opportunities". In: *ACM Transactions on Computer-Human Interaction* 29.4, pp. 1–35.

Huang, M. and R. T. Rust (2018). "Artificial intelligence in service". In: *Journal of Service Research* 21.2, pp. 155–172. DOI: 10.1177/1094670517752459.

ISO (2018). *ISO 9241-11:2018*. https://www.iso.org/standard/63500.html. Accessed January 14, 2025.

Jain, M. et al. (2018). "Evaluating and Informing the Design of Chatbots". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1, pp. 1–24. DOI: 10.1145/3196709.3196735.

Jurafsky, D. and J. H. Martin (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd. Prentice Hall. URL: https://www.researchgate.net/publication/200111340_Speech_and_Language_Processing_An_Introduction_to_Natural_Language_Processing_Computational_Linguistics_and_Speech_Recognition.

Kalenkova, A. A. et al. (2015). "Process Mining Using BPMN: Relating Event Logs and Process Models". In: *Software and Systems Modeling* 16.4, pp. 1019–1048. DOI: 10.1007/s10270-015-0502-0.

Karagiannis, D., S. Junginger, and R. Strobl (1996). "Introduction to Business Process Management Systems Concepts". In: *Springer eBooks*, pp. 81–106. DOI: 10.1007/978-3-642-80317-8_5.

Kelley, John F (1999). "The Wizard of Oz technique: A method for evaluating conversational interfaces". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 43. 5, pp. 482–486.

Klievtsova, N., J. Benzin, and T. Kampik (2023). *Conversational Process Modeling: Can Generative AI Empower Domain Experts in Creating and Redesigning Process Models?* URL: https://arxiv.org/html/2304.11065v2.

Klievtsova, N., J. Benzin, T. Kampik, et al. (2023). "Conversational Process Modelling: State of the Art, Applications, and Implications in Practice". In: *Lecture Notes in Business Information Processing*. Vol. 463. Springer, pp. 319–336. DOI: 10.1007/978-3-031-41623-1_19.

Kohavi, Ron, Alex Deng, et al. (2013). "Online controlled experiments at large scale". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1168–1176.

Kohavi, Ron, Roger Longbotham, et al. (2009). “Controlled experiments on the web: survey and practical guide”. In: *Data mining and knowledge discovery* 18.1, pp. 140–181.

Kohlbacher, M. (2010). “The effects of process orientation: a literature review”. In: *Business Process Management Journal* 16.1, pp. 135–152. DOI: 10.1108/14637151011017985.

Kourani, H., A. Berti, and D. Schuster (2024). *Process modeling with large language models*. URL: https://arxiv.org/html/2403.07541v1.

Kourani1, Humam, Alessandro Berti1, and Wil M. P. van der Aalst1 (2024). *Process Modeling with Generative AI*. URL: https://arxiv.org/pdf/2403.04327.

Krogstie, J. (2012). *Model-Based Development and Evolution of Information Systems*. Springer eBooks. DOI: 10.1007/978-1-4471-2936-3.

Leemans, S. J. J., D. Fahland, and W. M. P. Van der Aalst (2013). “Discovering Block-Structured Process Models from Event Logs - A Constructive Approach”. In: *Lecture Notes in Computer Science*, pp. 311–329. DOI: 10.1007/978-3-642-38697-8_17.

Leopold, H., J. Mendling, and O. Gunther (2016). “Learning from Quality Issues of BPMN Models from Industry”. In: *IEEE Software* 33.4, pp. 26–33. DOI: 10.1109/ms.2015.81.

Lewis, James R (2018). *Practical Usability Assessment: A Primer*. Elsevier.

López, A. et al. (2019). “From Process Models to Chatbots”. In: *Lecture Notes in Computer Science*. Vol. 11756. Springer, pp. 383–398. DOI: 10.1007/978-3-030-21290-2_24.

Melnyk, L. et al. (2024). “Prospects of Business Process Management Based on Chatbots”. In: *Problems and Perspectives in Management* 22.2, pp. 197–212. DOI: 10.21511/ppm.22(2).2024.16.

Mendling, J., G. Neumann, and M. Nüttgens (2005). *Yet another Event-Driven Process Chain (Extended version)*. Tech. rep. Vienna University of Economics, Business Administration, and University of Hamburg. URL: https://www.bwl.uni-hamburg.de/harcis/01-zentrum/01-team/chair/markus-nuettgens/publikationen/tr05-yepc.pdf.

Mendling, J., H. Reijers, and W. Van der Aalst (2010). “Seven process modeling guidelines (7PMG)”. In: *Information and Software Technology* 52.2, pp. 127–136. DOI: 10.1016/j.infsof.2009.08.004.

Microsoft (2024). *Overview of Process Mining in Power Automate*. Retrieved from Microsoft Learn. URL: https://learn.microsoft.com/en-us/power-automate/process-mining-overview.

Miner, BPMN (2016). “BPMN Miner: Automated Discovery of BPMN Process Models with Hierarchical Structure”. In: *Information Systems* 56, pp. 284–303. URL: https://eprints.qut.edu.au/83646/1/bpmnminer.pdf.

Molich, Rolf (2018). *Handbook of Usability Testing*. Wiley.

Moody, D. et al. (2022). “Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework”. In.

Moore, R. J. and R. Arar (2019). *Conversational UX Design: A Practitioner’s Guide to the Natural Conversation Framework*. DOI: 10.1145/3304087.

Mustansir, A., K. Shahzad, and M. K. Malik (2022). “Towards Automatic Business Process Redesign: An NLP Based Approach to Extract Redesign Suggestions”. In: *Automated Software Engineering* 29.1, pp. 1–25. DOI: 10.1007/s10515-021-00316-8.

Nasiri, S., A. Adadi, and M. Lahmer (2023). “Automatic Generation of Business Process Models from User Stories”. In: *International Journal of Electrical and Computer Engineering* 13.1, pp. 809–822. DOI: 10.11591/ijece.v13i1.pp809-822.

ngrok (n.d.). *ngrok | API Gateway, IoT Device Gateway, Secure Tunnels for Containers, Apps & APIs*. https://ngrok.com/. Accessed January 14, 2025.

Nielsen, Jakob (2010). *Usability engineering*. Morgan Kaufmann.

Object Management Group (2011a). *About the Case Management Model and Notation Specification Version 1.1*. URL: https://www.omg.org/spec/CMMN/1.1/.

— (2011b). *Business Process Model and Notation (BPMN) Version 2.0*. URL: https://www.omg.org/spec/BPMN/.

— (2015). *About the Decision Model and Notation Specification Version 1.5 beta*. URL: https://www.omg.org/spec/DMN.

Okoye, K. et al. (2017). "Process Models Discovery and Traces Classification: A Fuzzy-BPMN Mining Approach". In: *Journal of International Technology and Information Management* 26.4, pp. 2–50. DOI: 10.58729/1941-6679.1337.

P, Ray. (2023). "Benchmarking, ethical alignment, and evaluation framework for conversational AI: Advancing responsible development of ChatGPT". In: *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*.

Polančič, G. (2022). *Popularity-BPMN-rising*. Retrieved September 19, 2022. URL: https://goodelearning.com/popularity-bpmn-rising.

Pollock, L. (2024). *What Is an NLP Chatbot — And How Do NLP-Powered Bots Work?* URL: https://www.ultimate.ai/blog/ai-automation/how-nlp-text-based-chatbots-work.

Radford, A. et al. (2019). *Language Models are Unsupervised Multitask Learners*. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Radziwill, Nicole M and Morgan C Benton (2019). "Evaluating quality of chatbots and intelligent conversational agents". In: *arXiv preprint arXiv:1704.04579*.

Rebuge, Á. and D. R. Ferreira (2012). "Business Process Analysis in Healthcare Environments: A Methodology Based on Process Mining". In: *Information Systems* 37.2, pp. 99–116. DOI: 10.1016/j.is.2011.01.003.

Reijers, H. A., J. Mendling, and J. Recker (2014). "Business Process Quality Management". In: *Springer eBooks*. Springer, pp. 167–185. DOI: 10.1007/978-3-642-45100-3_8.

Ren, R. et al. (2019). "Usability of Chatbots: A Systematic Mapping Study". In: *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*. DOI: 10.18293/seke2019-029.

RobJenks (2022). *GPT-CodeGen: BPMN model generation using GPT AI language models*. Retrieved from GitHub. URL: https://github.com/RobJenks/gpt-codegen.

Roller, S. et al. (2020). "Recipes for Building an Open-Domain Chatbot". In: *arXiv preprint arXiv:2004.13637*. DOI: 10.48550/arXiv.2004.13637.

Rosenfeld, A. and N. Haimovich (2022). "Designing Rule-Based Conversational Agents with Behavioral Programming: A Study of Human Subjects". In: *EuroMed Journal of Business*. DOI: 10.1108/emjb-09-2021-0144.

Rubin, Jeffrey and Dana Chisnell (2008). *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.

Sadiq, S., G. Governatori, and K. Namiri (2007). "Modeling Control Objectives for Business Process Compliance". In: *Springer eBooks*, pp. 149–164. DOI: 10.1007/978-3-540-75183-0_12.

Santos, E. (2025). *14 must-have metrics for measuring user interaction with AI chatbots*. Accessed: 2025-02-14. URL: https://www.mo2o.com/en/blog/14-must-have-metrics-for-measuring-user-interaction-with-ai-chatbots/.

Sara Pérez-Soler Esther Guerra, Juan de Lara (2018). "Collaborative Modeling and Group Decision Making Using Chatbots in Social Networks". In: *IEEE Xplore Conference Proceedings*. URL: https://ieeexplore.ieee.org/document/8409918.

Sauro, Jeff (2016). *Measuring Usability with the System Usability Scale (SUS)*. Measuring Usability LLC.

Scuotto, I. (2022). *Natural Language Processing Chatbots: The Beginner's Guide*. URL: https://landbot.io/blog/natural-language-processing-chatbot.

Shawar, B. A. and E. Atwell (2007). "Chatbots: Are They Really Useful?" In: *Journal for Language Technology and Computational Linguistics* 22.1, pp. 29–49. DOI: 10.21248/jlcl.22.2007.88.

Shewale, R. (2024a). *65 Chatbot Statistics & Trends 2024*. URL: https://www.demandsage.com/chatbot-statistics/.

— (2024b). *65 Chatbot Statistics and Trends 2024*. URL: https://www.demandsage.com/chatbot-statistics/.

Shneiderman, Ben et al. (2018). *Designing the User Interface: Strategies for Effective Human-Computer Interaction, Global Edition*. https://books.google.de/books/about/Designing_the_User_Interface_Strategies.html?id=lPKGEAAAQBAJ&redir_esc=y. Accessed January 14, 2025.

Silva, G. R. S. and E. D. Canedo (2022). "Towards User-Centric Guidelines for Chatbot Conversational Design". In: *International Journal of Human-Computer Interaction* 40.2, pp. 98–120. DOI: 10.1080/10447318.2022.2118244.

Silver, Bruce (2011). *BPMN Method and Style*. URL: https://methodandstyle.com/books/bpmn-method-and-style/.

Sintoris, K. and K. Vergidis (2017). *Extracting Business Process Models Using Natural Language Processing (NLP) Techniques*. URL: https://www.semanticscholar.org/paper/Extracting-Business-Process-Models-Using-Natural-Sintoris-Vergidis/6c94ec89603ffafcdba4600bc7506a2df35cc246.

Sonbol, R., G. Rebdawi, and N. Ghneim (2022). *A Machine Translation Like Approach to Generate Business Process Model From Textual Description*. Research Square Preprint. URL: https://doi.org/10.21203/rs.3.rs-1242866/v1.

Tam, A., L. Chu, and D. Sculli (2001). "Business process modelling in small- to medium-sized enterprises". In: *Industrial Management + Data Systems* 101.4, pp. 144–152. DOI: 10.1108/02635570110390107.

TurboScribe (Oct. 2023). *TurboScribe: Transcribe audio and video to text*. Accessed: 2025-04-01. URL: https://turboscribe.ai/.

Van der Aalst, W. (2016). *Process Mining: Data Science in Action*. Springer. URL: https://vipa.wiwi.uni-saarland.de/wordpress/wp-content/uploads/2014/11/14WS_BWinfo_Schlueko_ProcessMining.pdf.

Van der Aalst, W., A. Adriansyah, and B. Van Dongen (2012). "Replaying History on Process Models for Conformance Checking and Performance Analysis". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.2, pp. 182–192. DOI: 10.1002/widm.1045.

Van der Aalst, W. M. P. (n.d.). "Business Process Management: A Comprehensive Survey". In: *ISRN Software Engineering* 2013 (), pp. 1–37. DOI: 10.1155/2013/507984.

Van Someren, Maarten W, Yvonne F Barnard, and Jacobijn AC Sandberg (1994). *The think aloud method: a practical approach to modelling cognitive processes*. Academic Press.

Vaswani, A. et al. (2017). "Attention is All You Need". In: *Advances in Neural Information Processing Systems* 30. DOI: 10.48550/arXiv.1706.03762.

Vogel-Walcutt, Jennifer et al. (2022). *Cognitive Load and the Whole-Person Learner*. https://thinkmagellan.com/wp-content/uploads/2022/10/Cognitive-Load-and-the-Whole-Person-Learner-White-Paper.pdf. Accessed: 2024-04-08.

Wang, Y. and W. et al Zhong (2023). “Aligning Large Language Models with Human: A Survey”. In.

Weizenbaum, Joseph (1966). “ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine”. In: *Communications of the ACM* 9.1, pp. 36–45.

Wesenberg, H. (2011). “Enterprise Modeling in an Agile World”. In: *Lecture Notes in Business Information Processing*. Springer eBooks, pp. 126–130. DOI: 10.1007/978-3-642-24849-8_10.

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer. DOI: 10.1007/978-3-540-73522-9.

Xu, Anbang et al. (2017). “Evaluating user satisfaction with end-to-end dialogue systems: The usefulness of subjective metrics”. In: *2017 IEEE International Conference on Cognitive Computing*. IEEE, pp. 65–71.

YAWL Foundation (2022). *YAWL BPM*. URL: https://yawlfoundation.github.io/.

YesChat.AI (2024). *BPMN-GPT: Kostenlose BPMN-Codegenerierung und -analyse*. URL: https://www.yeschat.ai/de/gpts-2OToEiyNiE-BPMN-GPT.

Zamora, Jennifer (2017). “A framework for designing and evaluating user experience in chat bots”. In: *International Conference on Universal Access in Human-Computer Interaction*. Springer, pp. 705–713.

8 Appendix

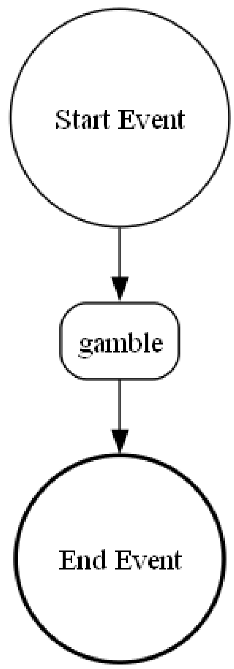

Figure 8.1: Test Case: simple bpmn_diagram

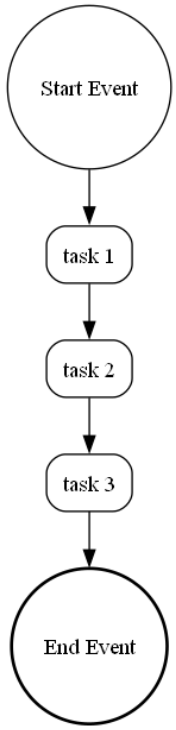

Figure 8.2: Test Case: bpmn_diagram_event_multiple_tasks

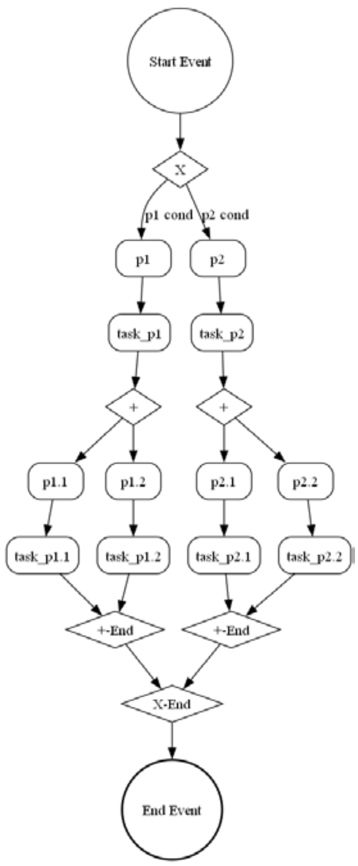

Figure 8.3: Test Case: bpmn_diagram_exclusive_with_parallel_nested

Figure 8.4: Test Case: bpmn_diagram_exclusive_with_parallel_nested_and_third_level

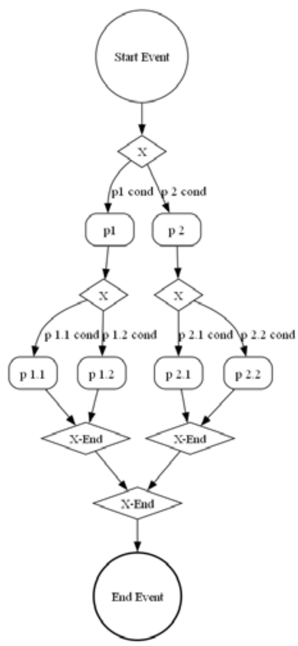

Figure 8.5: Test Case: bpmn_diagram_level_2_exclusive_gateway_nested_new

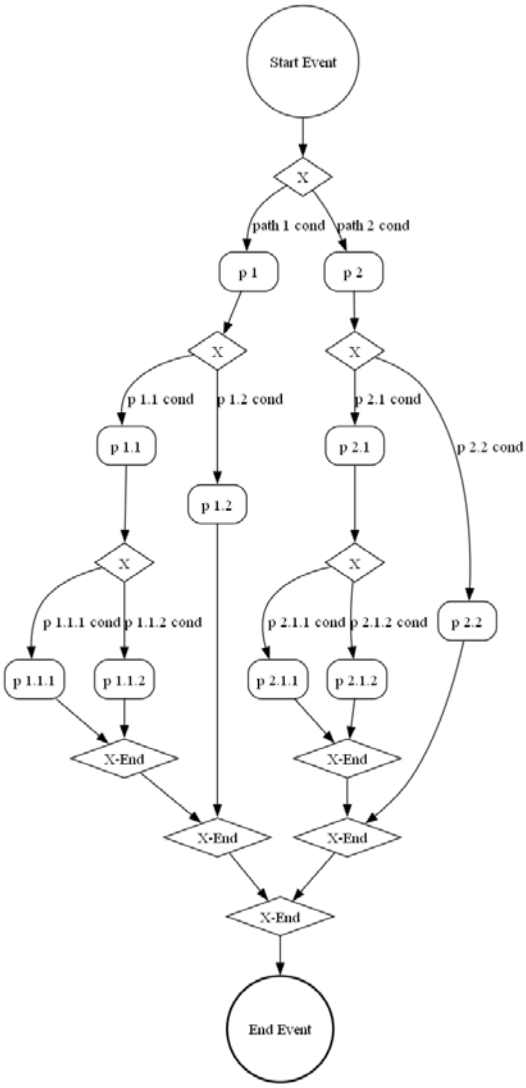

Figure 8.6: Test Case: bpmn_diagram_level_3_exclusive_gateway_nested

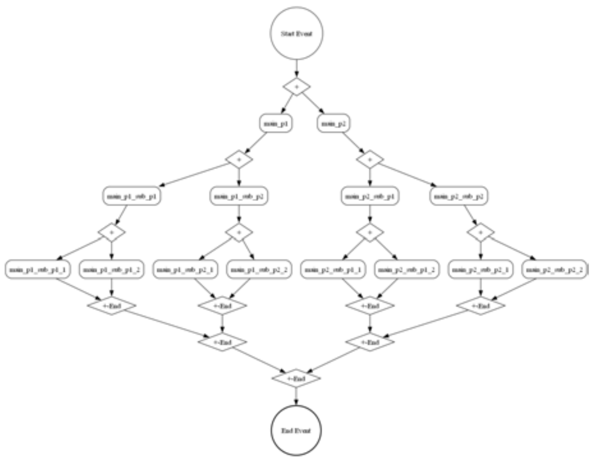

Figure 8.7: Test Case: bpmn_diagram_level_3_nested_parallel

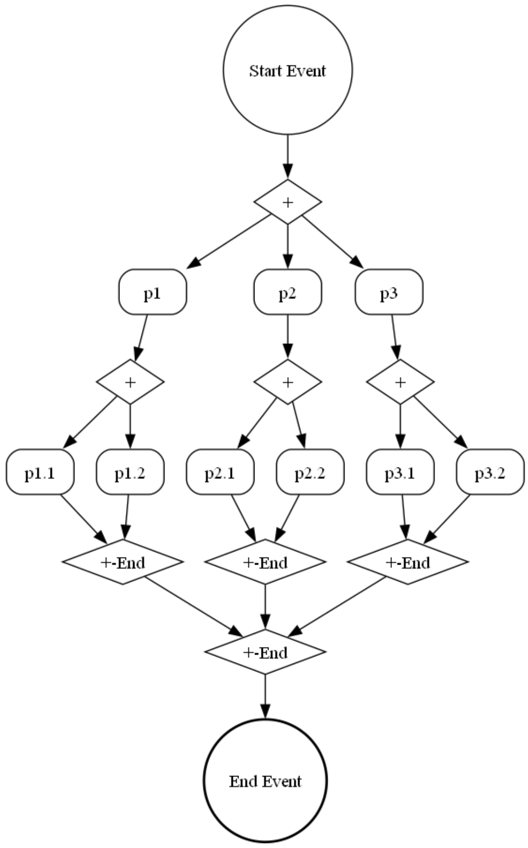

Figure 8.8: Test Case: bpmn_diagram_nested_parallel_with_nested_parallel

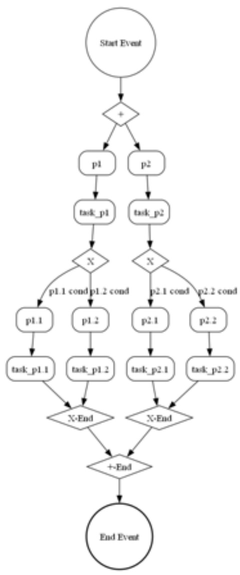

Figure 8.9: Test Case: bpmn_diagram_parallel_exclusive_parallel_nested

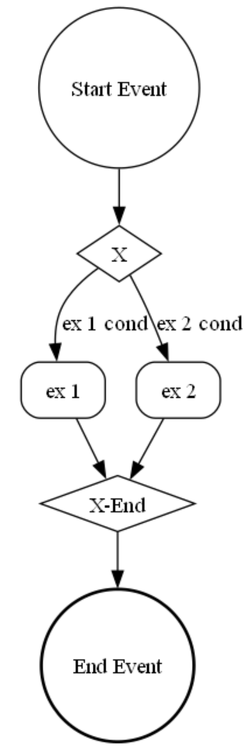

Figure 8.10: Test Case: bpmn_diagram_simple_exclusive_gateway

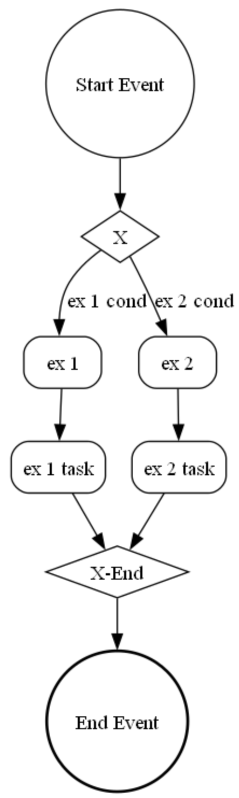

Figure 8.11: Test Case: bpmn_diagram_simple_exclusive_gateway_with_tasks

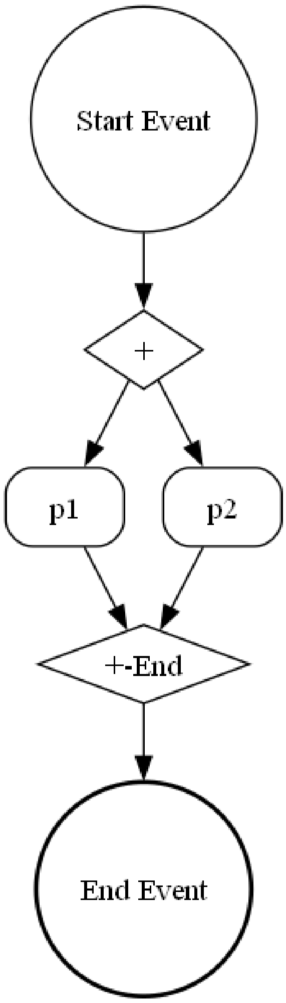

Figure 8.12: Test Case: bpmn_diagram_simple_parallel_paths

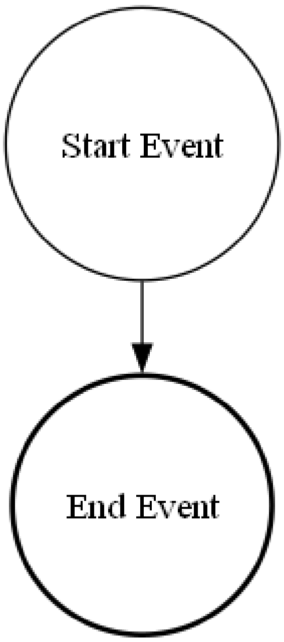

Figure 8.13: Test Case: bpmn_diagram_simple_start_end

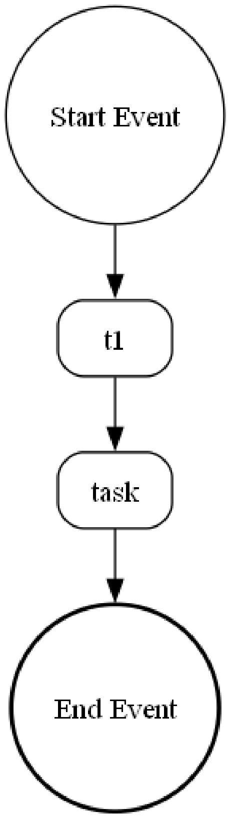

Figure 8.14: Test Case: bpmn_diagram_test_modify_list_with_added_task

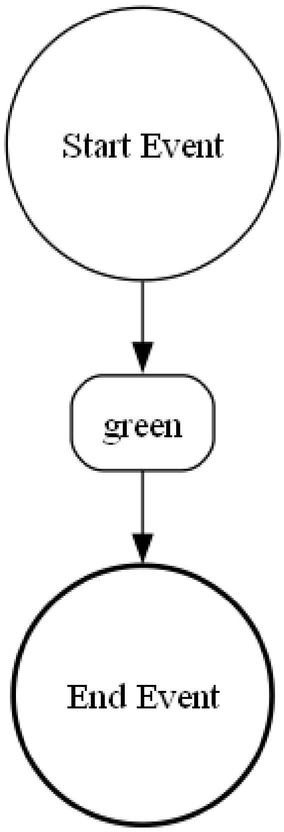

Figure 8.15: Test Case: bpmn_diagram_test_modify_list_with_name_change

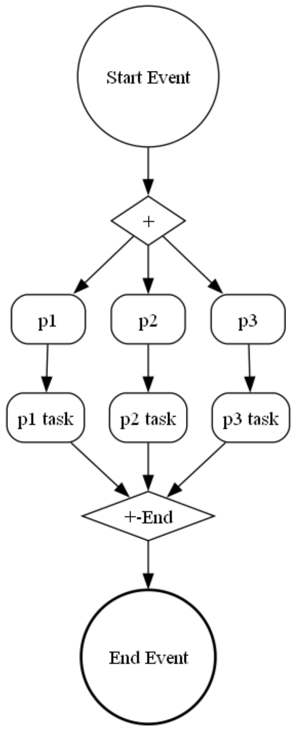

Figure 8.16: Test Case: bpmn_diagram_three_parallel_paths_with_tasks

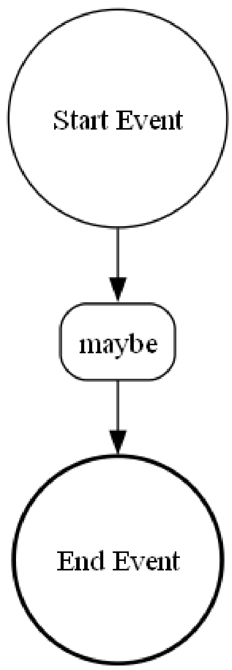

Figure 8.17: Test Case: bpmn_diagram_updated

8.1 User Testing

8.1.1 Introduction Text

Welcome, and thank you for participating in this user test.
The purpose of this session is to evaluate the usability and technical accuracy of a chatbot that automatically generates BPMN diagrams from natural language descriptions of business processes.
You will be given a short task in which you'll interact with the chatbot to model a business process. While doing so, I kindly ask you to think aloud—that is, verbalize what you find easy, unclear, or surprising as you interact. This will help me understand the clarity and guidance of the chatbot's interface and responses.
After completing the task, you'll be asked to fill out a short questionnaire. The entire session will take approximately 15 to 30 minutes.
If you're ready, let's begin with the task.

Business Process: Scenario A customer visits an online store. A customer places an order through an online shop. Once the order is received, the system checks whether the product is in stock.

- If the product is in stock, it is packaged and shipped to the customer. If the order is in stock the parcel is shipped and the process is ended.

- If the product is not in stock, the order is cancelled and the customer is notified by email. This notification is the end of the process.

fTask: Please make a modification to the business process and add an element after "A customer visits an online store."
Please add "The customer adds an item to the shopping basket."

User Satisfaction Survey Please rate the following four questions on a scale of 1 to 5.
While 1 = strongly disagree and 5 = strongly agree.

1. The chatbot's prompts were clear and easy to understand.

2. The BPMN diagram generated by the chatbot accurately reflected the process I described.

3. It was easy to interact with the chatbot and complete the tasks.

4. Overall, I am satisfied with my experience using the chatbot.

The last question is an open-ended question. Please answer whatever comes to mind:

1. What improvements would you suggest for the chatbot to make it more user-friendly?

8.2 User Testing (Deutsch)

Einleitungstext Willkommen und danke, dass du an diesem User-Test teilnimmst.
Ziel dieser Sitzung ist es, die Benutzerfreundlichkeit und technische Genauigkeit eines Chatbots zu evaluieren, der automatisch BPMN-Diagramme aus natürlichsprachlichen Beschreibungen von Geschäftsprozessen erstellt.
Du wirst eine kurze Aufgabe erhalten, in der du mit dem Chatbot interagierst, um einen Geschäftsprozess zu modellieren. Dabei bitte ich dich, laut zu denken – also auszusprechen, was du als einfach, unklar oder überraschend empfindest, während du interagierst. Das hilft mir dabei zu verstehen, wie klar und hilfreich die Benutzeroberfläche und die Antworten des Chatbots sind.
Nachdem du die Aufgabe abgeschlossen hast, wirst du gebeten, einen kurzen Fragebogen auszufüllen. Die gesamte Sitzung dauert etwa 15 bis 30 Minuten.
Wenn du bereit bist, beginnen wir mit der Aufgabe.

Geschäftsprozess: Szenario Ein Kunde besucht einen Onlineshop. Ein Kunde gibt eine Bestellung über einen Onlineshop auf. Sobald die Bestellung eingegangen ist, prüft das System, ob das Produkt auf Lager ist.

- Wenn das Produkt auf Lager ist, wird es verpackt und an den Kunden versendet. Wenn das Produkt auf Lager ist, wird das Paket versendet und der Prozess ist beendet.

- Wenn das Produkt nicht auf Lager ist, wird die Bestellung storniert und der Kunde per E-Mail benachrichtigt. Diese Benachrichtigung ist das Ende des Prozesses.

Aufgabe: Bitte nimm eine Änderung am Geschäftsprozess vor und füge ein Element nach „Ein Kunde besucht einen Onlineshop.“ hinzu.
Bitte füge „Der Kunde legt einen Artikel in den Warenkorb“ hinzu.

Zufriedenheitsumfrage Bitte bewerte die folgenden vier Fragen auf einer Skala von 1 bis 5. Wobei 1 = stimme überhaupt nicht zu und 5 = stimme voll und ganz zu.

1. Die Hinweise des Chatbots waren klar und leicht verständlich.

2. Das vom Chatbot generierte BPMN-Diagramm hat den von mir beschriebenen Prozess korrekt abgebildet.

3. Es war einfach, mit dem Chatbot zu interagieren und die Aufgaben abzuschließen.

4. Insgesamt bin ich mit meiner Erfahrung mit dem Chatbot zufrieden.

Die letzte Frage ist eine offene Frage. Bitte antworte ganz frei:

1. Welche Verbesserungen würdest du vorschlagen, um den Chatbot benutzerfreundlicher zu machen?

8.3 Interview Transcripts

8.3.1 Interview 1

Speaker 2 (0:00) Welcome and thank you for participating in this user test.
(0:05) The purpose of this session is to evaluate the usability and technical accuracy of the chatbot that automatically generates BPMN diagrams from natural language descriptions of business processes.
(0:17) You will be given a short task in which you will interact with the chatbot to model a business process.
(0:23) While doing so, I kindly ask you to think aloud, that is, verbalize what you find easy, unclear or surprising as you interact.
(0:33) This will help me understand the clarity and guidance of the chatbot's interface and responses.
(0:39) After completing the task, you will be asked to fill out a short questionnaire.
(0:44) The entire session will approximately take 15 to maximum 30 minutes.
(0:48) If you are ready, let's begin with the task.
(0:52) This is your scenario and here you can see the chatbot.

Speaker 1 (0:57) Okay, the scenario is a customer visits an online store.
(1:03) A customer places an order through an online shop.
(1:07) Once the order is received, the system checks whether the product is in stock.
(1:15) If the product is in stock, it is packaged and shipped to the customer.
(1:21) If the order is in stock, the parcel is shipped and the process is ended.
(1:25) If the product is not in stock, the order is cancelled and the customer is notified by e-mail.
(1:33) This notification is the end of the process.
(1:39) I will analyze the new check.
(1:47) I will go through the task sentence by sentence.
(1:51) I have to start the chatbot with Hi.
(2:12) Hello, welcome to Bot2 BPMN.
(2:16) Do you want to learn more about the different elements of a BPMN diagram?
(2:23) I am quite confident with the elements, so I press No.
(2:32) Let's get started with mapping out your process.
(2:36) What happens first in your process?
(2:39) The first sentence is the customer visits an online store.
(2:44) I say a task and describe it with visits online store.
(3:07) The next step is customer places an order.
(3:11) I select task and place an order.
(3:37) The next sentence is the system checks whether the order is received.
(3:43) I think once the order is received, it is not so important for the BPMN.
(3:50) I go on with a new task.
(3:56) I type a message.
(4:03) System checks product in stock.
(4:23) Now I have the two bullet points.
(4:27) The order is in stock or the product is not in stock.
(4:31) I choose gateway.
(4:36) The first task within the gateway path is in the first bullet point.
(5:09) The product is in stock.

(5:12) The order is packaged and shipped to the customer.
(5:17) Please describe the condition under which this gateway path gets executed.
(5:54) Product in stock.
(6:01) What happens next?
(6:06) The next thing is a new task to check if the order is in stock.
(6:25) Task within the first gateway path to describe the other gateway path.
(6:31) Nested gateway.
(6:34) First task within the first gateway path to check if order is in stock.
(7:06) What happens next is a nested gateway.
(7:10) Please describe the first task within the gateway path.
(7:27) Order is in stock.
(7:31) The parcel is shipped.
(7:47) Under what condition?
(7:50) It is shipped if the order is in stock.
(8:06) What happens next within your parcel path?
(8:18) I would go to describe the other gateway path.
(8:25) What happens if the order is not in stock?
(8:38) I guess the customer has to be notified about missing pieces.
(9:08) If the order is not in stock.
(9:20) What happens next within your process?
(9:24) I can choose task within the second gateway path.
(9:32) Gateway ends in the nested gateway parallel path.
(9:41) I'm in my nested gateway.
(9:45) I click the gateway ends.
(9:52) The nested gateway path is closed.
(9:59) What happens next within your process?
(10:02) I can choose task within the first gateway path.
(10:05) Or describe the other gateway path.
(10:07) I go to describe the other gateway path.
(10:12) To clarify what happens if the product is not in stock.
(10:43) If the product is not in stock, the order is cancelled.
(10:50) I say cancel order.
(11:03) Under what condition?
(11:05) Product is not in stock.
(11:14) What happens next within your process?
(11:17) New task in the second gateway path.
(11:23) To notify the customer.
(11:37) We have both paths at an end.
(11:40) We can close the gateway.
(11:43) Click the gateway ends.
(11:50) The process ends.
(12:01) I'm waiting for the response after the process ends.
(12:10) I got a link.
(12:14) The question is your diagram accurately depicting your business process?
(12:32) I'm looking at the BPMN.
(12:36) Start event.
(12:41) Visits online store.
(12:44) Places order.

(12:47) System checks product in stock.
(12:53) XOR gateway.
(12:58) Left lane product in stock.
(13:01) Product packed and shipped.
(13:03) Check if order in stock.
(13:04) New XGateway.
(13:09) Parcels shipped.
(13:11) Customers notified.
(13:14) Order in stock.
(13:16) Order not in stock.
(13:18) XEnter gateway.
(13:20) Everything looks good.
(13:29) Right.
(13:29) On the right side.
(13:32) Product is not in stock.
(13:33) Cancel order.
(13:38) XEnd gateway.
(13:40) End event.
(13:45) Yes.
(13:49) Matching your BPMN process.
(13:54) I would say yes.

Speaker 2 (13:55) Please now click no instead of yes.
(14:05) So you can make a modification to the BPMN.

Speaker 1 (14:11) Okay.
(14:15) Would you like to make changes to the generated BPMN diagram?
(14:19) Yes.
(14:20) So it's going on.

Speaker 2 (14:24) Please now make a modification to the business process.
(14:28) And add an element after the customer visits an online store.
(14:33) And please add the customer adds an item to the shopping basket.

Speaker 1 (14:46) I have to look at the diagram.
(14:49) I see every BPMN element with an ID.
(15:00) And I type.
(15:03) Please tell me after what ID the change needs to be made.
(15:09) And the ID is for that 425.
(15:23) And I'm waiting for the response.
(15:28) What do you want to change?
(15:30) According to the task I would say add an element.
(15:34) I have to choose whether from name of my element, add an element or delete an element.
(15:40) I click add an element.
(15:43) And what type of element do you want to add?
(15:47) I have to look at the task.
(15:52) Yes.

(15:52) Awesome task.
(15:59) The customer adds an item to the shopping basket.
(16:05) Please describe the task you want to add.
(16:09) So the customer adds an item to the basket.
(16:21) Press enter.
(16:33) I'm waiting for the response and I hope I get a new image.
(16:52) And there.
(16:59) I see a new task.
(17:03) Directly behind the ID.
(17:05) I choose from 425.

Speaker 2 (17:11) Thank you. This is all for the tasks.
(17:13) Now we have a small user satisfaction survey.
(17:21) Please rate the following questions on a scale of 1 to 5, where 1 is strongly disagree and 5 is strongly agree.
(17:31) First, the chatbots prompts were clear and easy to understand.

Speaker 1 (17:40) I would say it is a 4.
(17:45) Just because it was my first time using the chatbot...

Speaker 2 (18:15) The BPMN diagram generated by the chatbot accurately reflected the process I described.

Speaker 1 (18:24) I would say everything is as I expected.
(18:32) Everything is good and fully there, so I would go with 5.

Speaker 2 (18:43) It was easy to interact with the chatbot and complete the task.

Speaker 1 (18:51) I think it was a 5...

Speaker 2 (19:19) Overall, I am satisfied with my experience using the chatbot.

Speaker 1 (19:24) Yes, I would totally agree.
(19:25) I am very surprised how easy it goes.
(19:43) I would say to have more examples so other users have used it and got an idea what is possible and where are the limits maybe.
(20:10) I didn't come along with some limits in this task, but maybe there are some.

Speaker 2 (20:19) Thank you very much.

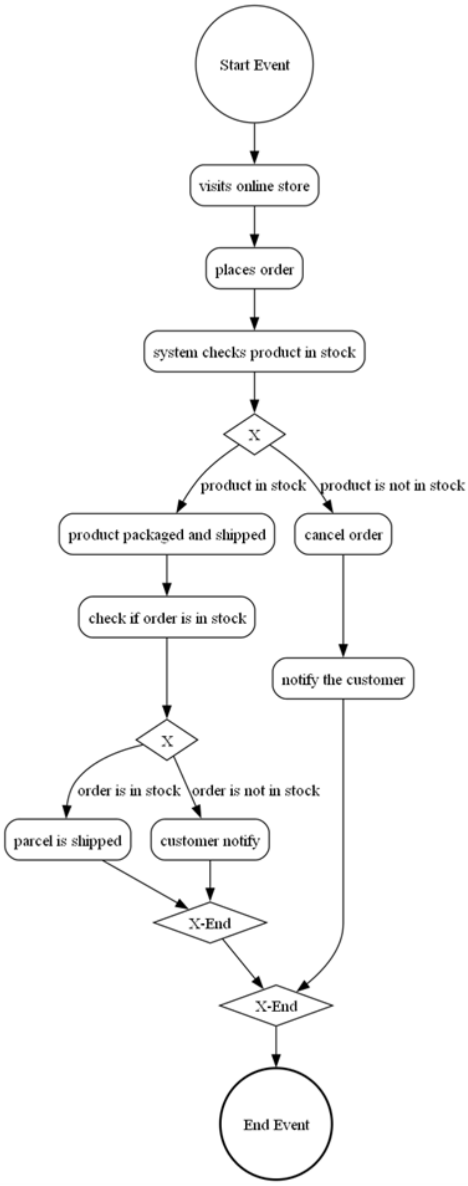

Figure 8.18: Generated output User 1

8.3.2 Interview 2

Speaker 2 (0:00) Welcome and thank you for participating in this user test. The purpose of this session is to evaluate the usability and technical accuracy of the chatbot that automatically generates BPMN diagrams from natural language descriptions of business processes.
(0:16) You will be given a short task in which you will interact with the chatbot to model a business process.
(0:23) While doing so, I kindly ask you to think aloud—that is, verbalize what you find easy, unclear or surprising as you interact.
(0:32) This will help me understand the clarity and guidance of the chatbot's interface and responses.
(0:38) After completing the task, you will be asked to fill out a short questionnaire.
(0:43) The entire session will approximately take 15 to maximum 30 minutes.
(0:48) If you are ready, let's begin with the task. This is your scenario and here you can see the chatbot.

Speaker 1 (0:58) Alright… so… let me take a moment to go through this scenario slowly.
(1:06) Okay, so first: a customer visits an online store.
(1:13) Then… they place an order through the online shop.
(1:20) Once the order is received, the system checks whether the product is in stock.
(1:31) If it is… it gets packaged and shipped to the customer.
(1:39) If it's not in stock… then the order is cancelled and the customer is notified by email.
(1:48) And that notification marks the end of the process.
(1:53) Yeah… alright, pretty standard e-commerce process.
(1:59) Okay, I'll go ahead and start interacting with the bot now.
(2:04) Just typing "Hello" to begin.
(2:09) The bot responds: Hello, welcome to Bot2 BPMN. Do you want to learn more about the different elements of a BPMN diagram?
(2:24) I'll click "No"—I already know my way around BPMN modeling.
(2:30) It now says: Let's get started with mapping out your process. What happens first?
(2:45) First task... I'm entering: customer visits an online store.
(3:05) That's a user-initiated action, so it makes sense as a simple task.
(3:12) Next up: customer places an order.
(3:32) That's another straightforward task—I'll type that in.
(3:39) Then we get: system checks whether the product is in stock.
(3:59) Technically this is a service task, but since task types aren't specified here, I'll go with "check if product is in stock."
(4:10) Okay… now we've hit a decision point.
(4:15) I add a gateway.
(4:18) For the first path, I add: packaged and shipped to customer.
(4:38) On the second path, I enter: cancelled and customer is notified.
(4:58) Now I end both branches and connect them to a common end event.
(5:05) That wraps it up.
(6:00) The bot asks: Is your diagram accurately depicting your business process?
(6:25) Let me scroll through and check…
(6:42) Yeah, it looks right to me.
(6:56) The structure is clear, and the elements are where I'd expect them to be.

Speaker 2 (7:01) Please now click "no" instead of "yes" so you can make a modification to the BPMN.

Speaker 1 (7:16) Alright.
(7:18) It's now asking: Would you like to make changes to the generated BPMN diagram?
(7:33) I'll select "Yes."

Speaker 2 (7:37) Please now make a modification to the business process and add an element after "the customer visits an online store."
(7:44) And please add "The customer adds an item to the shopping basket."

Speaker 1 (7:59) Got it.
(8:01) So first I check the diagram.
(8:05) The ID for "customer visits an online store" is… 449.
(8:15) When it asks me: Please tell me after what ID the change needs to be made—I'll enter 449.
(8:35) Now it asks: What do you want to change?
(8:45) I'll choose "Add an element."
(8:49) Then it asks: What type of element do you want to add?
(9:09) I'll go with "Task."
(9:13) And now it prompts me: Please describe the task you want to add.
(9:33) I'll type: adds an item to the basket.
(9:53) Pressing Enter…
(9:58) Alright, just waiting for the diagram to refresh.
(10:33) Okay… there it is.
(10:46) The new task appears right after the store visit.
(10:55) Looks clean. Flow still makes sense.

Speaker 2 (11:15) Thank you. This is all for the tasks. Now we have a small user satisfaction survey.
(11:31) Please rate the following questions on a scale of 1 to 5, where 1 is strongly disagree and 5 is strongly agree.
(11:38) First, the chatbot's prompts were clear and easy to understand.

Speaker 1 (11:53) I'd give that a 3.
(11:56) The prompts are generally okay…
(11:59) but when editing, it starts to feel a bit repetitive.
(12:04) It asks too many follow-up questions for small tasks—like, you have to confirm every little thing in separate steps.

Speaker 2 (12:11) The BPMN diagram generated by the chatbot accurately reflected the process I described.

Speaker 1 (12:26) That's a 5.
(12:28) Everything looked accurate.
(12:31) I didn't spot any missing or misplaced elements.

Speaker 2 (12:36) It was easy to interact with the chatbot and complete the tasks.

Speaker 1 (12:51) I'd say 4.
(12:53) Most of it felt natural, especially the initial process building.
(12:58) But manually typing IDs for edits might be a bit much for more complex diagrams.

Speaker 2 (13:04) Overall, I am satisfied with my experience using the chatbot.

Speaker 1 (13:19) Yep, I'd give it a 4.
(13:22) It does the job well.
(13:24) Just needs a few tweaks to make the interaction smoother, especially for repeat users or longer processes.

Speaker 2 (13:30) What improvements would you suggest for the chatbot to make it more user-friendly?

Speaker 1 (13:45) I'd say… first off, reduce the number of repetitive prompts during editing.
(13:51) It could group related questions together instead of asking them one by one.
(13:57) Also, being able to select an element visually—like just clicking on it in the diagram—would really help.
(14:04) And maybe an advanced option for task types, lanes, or subprocesses could be useful for people who know BPMN already.
(14:12) But yeah… overall, it's good.

Speaker 2 (14:19) Thank you very much.

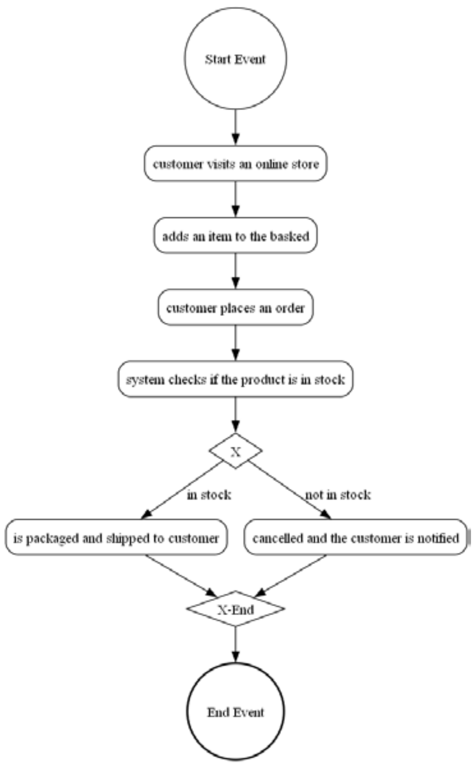

Figure 8.19: Generated output User 2

8.3.3 Interview 3

Speaker 2 (0:05) Willkommen und danke, dass du an diesem User-Test teilnimmst. Ziel dieser Sitzung ist es, die Benutzerfreundlichkeit und technische Genauigkeit des Chatbots zu bewerten, der automatisch BPMN-Diagramme aus natürlichsprachlichen Beschreibungen von Geschäftsprozessen erstellt.
(0:21) Du wirst eine kurze Aufgabe erhalten, indem du mit dem Chatbot interagierst, um einen Geschäftsprozess zu modellieren. Dabei bitte ich dich, laut zu denken, also auszusprechen, was du als einfach, unklar oder überraschend empfindest, während du mit dem Bot agierst. Das hilft mir dabei zu verstehen, wie klar und hilfreich die Benutzeroberfläche und die Antworten des Chatbots sind. Nachdem du diese Aufgabe abgeschlossen hast, wirst du gebeten, einen kurzen Fragebogen auszufüllen. Die gesamte Sitzung dauert etwa 15 bis 30 Minuten. Wenn du bereit bist, beginnen wir mit der Aufgabe.

Speaker 1 (0:54) Ja, ich bin bereit.
(1:01) So, ein Kunde besucht einen Online-Shop, der Kunde gibt eine Bestellung über einen Online-Shop auf. Sobald die Bestellung eingegangen ist, prüft das System, ob das Produkt auf Lager ist. Dann würde ich jetzt als allererstes so prüfen, ob das Produkt auf Lager ist.
(1:26) Also, der Kunde besucht einen Online-Shop, dann würde ich das als erstes eintragen.
(1:56) Und gibt eine Bestellung auf. Das wäre jetzt der erste Schritt.
(2:14) Okay.
(2:24) Aha, hallo und willkommen.
(2:26) Hier bin ich, um das Gespräch durchzuführen und durch jeden Prozess Schritt für Schritt zu beschreiben. Am Ende unseres Gesprächs erhalten Sie ein komplettes BPMN-Diagramm. Möchten Sie mehr über die fundierenden Elemente?
(2:49) Ja.
(2:56) Eintauchen in die wichtigsten Komponenten.
(2:59) Task oder Aktivität, das hatten wir eben.
(3:02) Ja, genau.
(3:05) Task
(3:06) Zeigt die Reihenfolge der Aufgabe an.
(3:10) Getaway.
(3:13) Ja, genau, das ist der Entscheidungspunkt.
(3:17) Okay.
(3:19) Findet die Kondition der Kriterien.
(3:26) Genau.
(3:27) Okay.
(3:28) Haben Sie weitere Fragen?
(3:29) Nein.
(3:31) Beginnen wir mit dem Mapping Ihres Prozesses.
(3:36) Was passiert erst in Deinem Prozess?
(3:41) Das erste.
(3:42) Achso, guck mal, da kann ich ja wählen.
(3:44) Das ist, achso.
(3:46) Ich dachte jetzt, ich könnte wählen.
(3:49) Aufgabe, einzelnen Schritt, der im Rahmen des...
(3:55) Dann wäre das das.
(3:56) Das ist ja die einzige Aktion, weil der Kunde sucht im Grunde...
(4:02) Also, es wird geprüft, ob das Produkt da ist.

(4:13) Bitte beschreiben Sie die Aufgabe.
(4:19) Kunde, die sucht Onlineshop.
(4:38) Was ist das Nächste?
(4:43) Kunde gibt Bestellung auf.
(4:47) Achso, dann geht es ja wieder, dass Sie in Linear kommen.
(5:09) Okay.
(5:15) Ah ja, das ist ja gut.
(5:17) Und dann als Nächstes.
(5:26) Das ist ja auch wieder Linear.
(5:28) Dann, genau, dann...
(5:33) So, Prüfung, ob das Produkt da ist.
(5:48) Auf Lager ist, oder?
(5:50) Nein, machen wir das mal.
(5:53) Auf Lager ist.
(5:56) So.
(5:58) Okay.
(6:01) So.
(6:03) Und als Nächstes muss dann natürlich ja...
(6:06) Gibt es ja zwar unterschiedliche...
(6:09) Wenn das Produkt auf Lager ist, wird es an den Kunden versendet.
(6:16) Okay, also dann hätte ich jetzt als Nächstes...
(6:23) Dann wäre jetzt eine Entscheidung.
(6:31) Produkt auf Lager.
(6:54) Produkt auf Lager.
(6:58) Okay.
(7:02) So, bitte beschreiben den Pfad, oder den...
(7:09) Produkt wird versandt.
(7:31) So.
(7:33) Was passiert als Nächstes in Ihrem Prozess?
(7:36) Jetzt würde ich ja das andere...
(7:40) den anderen...
(7:45) Okay.
(7:48) Produkt...
(7:50) Produkt...
(7:51) Nein.
(7:55) Nicht...
(7:57) Lager.
(7:59) Genau, Produkt nicht auf Lager.
(8:06) Und dann kommt jetzt eine E-Mail.
(8:19) E-Mail an den Kunden.
(8:21) Okay.
(8:27) So.
(8:28) Und danach ist ja der...
(8:36) Okay, dann wären jetzt...
(8:40) Okay.
(8:41) Jo.
(8:44) Das Gateweay endet.
(8:46) Und dann Prozess abgeschlossen.

(8:49) Dann ist ja im Prinzip...
(8:59) Der Prozess endet.
(9:01) So.
(9:02) Okay, der Prozess endet jetzt.
(9:06) Genau.
(9:10) Und...
(9:11) Ja, das war ja eigentlich relativ...
(9:14) Gut, bitte nimm eine Änderung am Geschäftsprozess vor.
(9:24) Ach, ja, guck mal, super.
(9:27) Das ist ja großartig, da wird es mir angezeigt. Also das Diagramm...
(9:29) Dass der staatliche Kunde besucht einen anderen Shop,
(9:31) Kunde gibt Bestellungen auf,
(9:33) Prüfung, ob das Produkt auf Lager ist.
(9:36) Ja, großartig.
(9:40) Das ist ja...
(9:41) Das ist ja gut gehandelt gewesen.

Speaker 1 (9:47) Bitte nimm jetzt eine Änderung am Prozess vor.
(9:52) Und...
(9:53) Füge ein Element...
(9:57) Nach, ein Kunde besucht einen Online-Shop hinzu.

Speaker 2 (10:03) Und ich möchte natürlich ja...
(10:02) Möchten Sie generell etwas ändern?
(10:05) Ja, und ich sollte ja jetzt insgesamt etwas ändern hier, nachdem eigentlich das Produkt auf Lager ist.
(10:16) Genau, dann
(10:38) Fügen ein Element nach dem Kundenbesuch
(10:41) Ach so, ein Element dazu, okay. Kunde gibt Bestellung auf, Kunde besucht den Onlineshop
(10:50) Genau, und hier müsste ich jetzt einfach einfügen.
(10:53) hier will ich jetzt einfügen, Kunde gibt Bestellung, Kunde besucht Onlineshop
(11:02) Kunde besucht Onlineshop
(11:07) Ja, jetzt Kunde legt Produkt.. Das ist hier Kunde gibt Bestellung auf
(11:24) jetzt mal nur gucken, das war jetzt, ach so
(11:51) das musste ich machen, jetzt setze ich mal ein Klammern
(12:04) So …Name des Elements. Ach so, ich möchte noch ein Element dazufügen, das ist gut. Kunde legt ein Artikel in den Warenkorb Kunde legt Artikel in den Warenkorb.
(12:45) So, genau, das ist ja schon mal gut. Ach guck ..

Speaker 2 (12:56) Genau, das hat jetzt nicht geklappt, weil du die ID nicht als Zahl eingegeben hast, sondern als ID in Klammern (466), deswegen hat er das nicht genommen. Genau, nur einmal zur Erklärung des Bildes
(13:22) Okay, genau, dann gehen wir jetzt einmal in die Zufriedenheitsumfrage über.
(13:31) Bitte bewerte die folgenden vier Fragen auf einer Skala von 1 bis 5, wobei 1 Stimme überhaupt nicht zu bedeutet und 5 Stimme voll und ganz zu. Die Hinweise des Chatbots waren klar und leicht verständlich.

Speaker 1 (13:45) Ja, das ist eine 5 auf jeden Fall, also die waren klar.

Speaker 2 (13:50) Das vom Chatbot generierte BPMN-Diagramm hat den von mir beschriebenen Prozess korrekt abgebildet

Speaker 1 (13:59) Also, ja, ich überlege jetzt gerade, weil ich denke
(14:04) okay, aber ich habe ja selbst einen Fehler gemacht, also im Prinzip ist es korrekt abgebildet also auch 5.

Speaker 2 (14:12) Es war einfach, mit dem Chatbot zu interagieren und die Aufgabe abzuschließen

Speaker 1 (14:23) Ja, auch 5.

Speaker 2 (14:29) Insgesamt bin ich mit meiner Erfahrung mit dem Chatbot zufrieden.

Speaker 1 (14:39) Da bin ich bei einer 4, weil ich selbst den Fehler gemacht habe. Das ärgert mich, so, genau.

Speaker 2 (14:48) Fünftens, das ist jetzt eine offene Frage, welche Verbesserung würdest du vorschlagen, um den Chatbot benutzerfreundlicher zu machen?

Speaker 1 (14:45) Also für mich wäre tatsächlich so dieser Punkt wie das abgebildet werden soll, also jetzt hier, wenn ich jetzt irgendwie noch was Neues einfüge, um zu sagen, an welcher Stelle war das für mich nicht so ganz klar ersichtlich, was ich da jetzt eingeben muss, ob es jetzt tatsächlich nur die Zahl ist oder Idee dazu, wie auch immer, also das wäre für mich jetzt, das war noch ein bisschen erklärungsbedürftig, ansonsten fand ich das Handeln damit aber sehr gut.

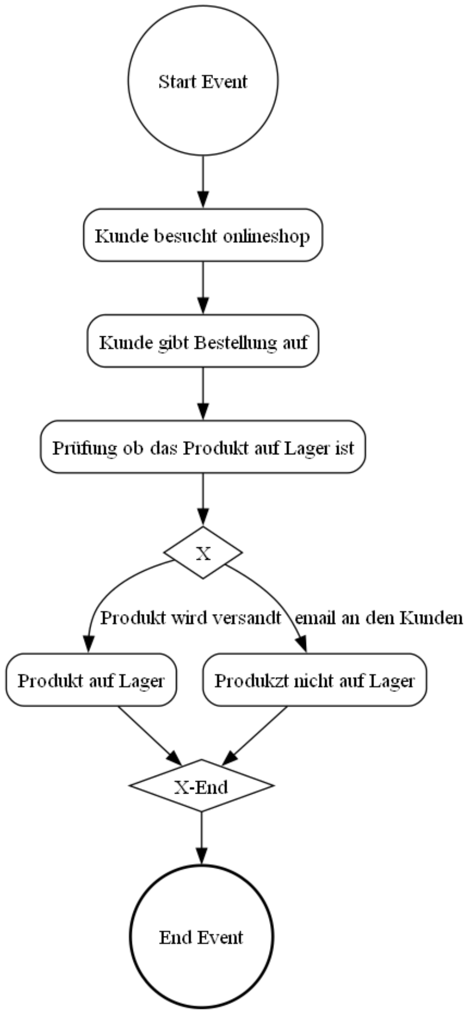

Figure 8.20: Generated output User 3 - no modification achieved

8.3.4 Interview 4

Speaker 2 (0:00) Willkommen und danke, dass du an diesem User-Test teilnimmst. Ziel dieser Sitzung ist es, die Benutzerfreundlichkeit und die technische Genauigkeit des Chatbots zu evaluieren, der automatisch BPMN-Diagramme aus natürlichsprachlichen Beschreibungen von Geschäftsprozessen erstellt.
(0:16) Du wirst eine kurze Aufgabe erhalten, in der du mit dem Chatbot interagierst, um einen Geschäftsprozess zu modellieren.
(0:23) Dabei bitte ich dich, laut zu denken, also auszusprechen, was du denkst, was du einfach oder unklar oder als überraschend empfindest, während du mit dem Bot interagierst. Das hilft mir dabei, zu verstehen, wie klar und hilfreich die Benutzeroberfläche und die Antworten des Chatbots sind. Nachdem du die Aufgabe abgeschlossen hast, wirst du gebeten, einen kurzen Fragebogen auszufüllen.
(0:54) Die gesamte Sitzung dauert etwa 15 bis 30 Minuten. Wenn du bereit bist, beginnen wir mit der Aufgabe.

Speaker 1 (1:02) Ja, ich tipp erstmal „Hallo“.
(1:10) Ah, guck da ist was. Ich les mal.
(2:15) Gott, so viel Text.
(2:31) Okay, jetzt muss man erst mal wissen, was ein Gateway ist.
(3:03) Nein, ich habe erst mal keine weiteren Fragen. Moment, was passiert erst in deinem Prozess? Eine einzige Aktion oder ein Schritt?
(3:12) Erst mal eine einzige Aktion passiert. Ein Kunde besucht einen Online-Shop.
(3:37) Als nächstes. Ein Kunde gibt eine Bestellung über einen Online-Shop auf.
(4:00) Das System prüft, ob das Produkt auf Lager ist. Das ist auch wieder eine einzige Aktion.
(4:17) Jetzt kommen wir zu einem Gateway. Ach so, nein, es ist ein Gateway. Und zwar, wenn das Produkt auf Lager ist, wird es verpackt und an den Kunden versendet. Oh, ist das doppelt? Moment, wenn das Produkt auf Lager ist, wird es verpackt.
(5:11) Wenn das Produkt auf Lager ist, wird das Paket versendet. Ach so, Paket wird versendet.
(5:17) Gut, so sind wir gerade. Okay, dann tippe ich auf Lager. Als nächstes, wenn das Produkt nicht auf Lager ist, aber wahrscheinlich kommt das jetzt später. Ach so, Aufgabe im ersten Gateway. Das ist jetzt der andere Gateway-Pfad.
(6:05) Genau, die Bestellung wird storniert. Beschreiben Sie unter welcher Bedingung die Aufgabe? Unter der Bedingung, dass das Produkt nicht auf Lager ist. Als nächstes bekommt der Kunde eine E-Mail.
(6:40) Achso, und Kunde bekommt E-Mail. Dann ist der Prozess beendet. Das Gateway endet. Prozess endet.
(7:12) Kommt jetzt noch was? Ja. Ah, oh, das ist das neue Diagramm. Start Event. Kunde besucht Online-Shop. Kunde gibt Bestellung auf. Prüfung, ob das Produkt auf Lager ist. Dann ist es entweder auf Lager. Produkt wird versendet.
(7:42) Produkt nicht auf Lager. E-Mail an den Kunden. Aha. Ende. Endevent. Okay, klicken Sie auf den Link, um das Bild anzuzählen.
(8:07) Dann stellt ihr Diagramm genau den Geschäftsprozess dar. Ähm, ja.

Speaker 2 (8:29) Genau, danke. Klicken Sie hier bitte auf Nein. Es gibt nämlich jetzt eine neue Aufgabe.
(8:37) Und zwar bitte nimm Änderungen am Geschäftsprozess vor und füge ein Element ein, nach „Ein Kunde besucht einen Online-Shop“. Und füge bitte das Element „Der Kunde legt

einen Artikel in den Warenkorb“ hinzu.

Speaker 1 (8:57) Okay, ich sage hier Nein. Dann stellt ihr Diagramm. Ja, ich würde gerne Änderungen vornehmen. Moment. Der Kunde legt den Artikel.
(9:51) Also, nach der ID 489 soll eine Änderung vorgenommen werden. Was wollen Sie ändern? Namen des Elements?
(10:09) Ähm, nee. Fügen Sie ein Element hinzu. Und zwar eine Aufgabe. Ein Artikel in den Warenkorb legen.
(11:20) Ja, das sieht doch gut aus.

Speaker 2 (11:31) Okay, vielen Dank. Danke, dann kommen wir jetzt zur Zufriedenheitsumfrage. Bitte bewerte die folgenden vier Fragen auf einer Skala von 1 bis 5, wobei 1 Stimme überhaupt nicht zu bedeutet und 5 Stimme voll und ganz zu.
(11:51) Erstens, die Hinweise des Chatbots waren klar und leicht verständlich.

Speaker 1 (12:01) Da gebe ich eine 4.

Speaker 2 (12:08) Zweitens, das vom Chatbot generierte BPM-Diagramm hat den von mir beschriebenen Prozess korrekt abgebildet.

Speaker 1 (12:11) Das ist korrekt, eine 5.

Speaker 2 (12:14) Es war einfach, mit dem Chatbot zu interagieren und die Aufgaben abzuschließen.

Speaker 1 (12:18) Ja, das war einfach, eine 5.

Speaker 2 (12:20) Insgesamt bin ich mit meiner Erfahrung mit dem Chatbot zufrieden.

Speaker 1 (12:25) Ja, 5. Ich bin zufrieden.

Speaker 2 (12:29) Die letzte Frage ist eine offene Frage. Bitte beantworte ganz frei: Welche Verbesserung würdest du vorschlagen, um den Chatbot benutzerfreundlicher zu machen?

Speaker 1 (12:38) Ich glaube, ich fände es einfacher, wenn die Aufgaben so eingegeben werden könnten, dass sie auch nach der Formulierung Sinn ergeben.
(12:51) Also, dass ein Gateway als erstes erstellt wird und dann die zwei möglichen Aufgabenteile, die dann eben folgen.
(13:05) Und nicht erst die Aufgaben und dann die Entweder-Oder-Gateway-Entscheidung. Ich hoffe, das ist verständlich, so wie ich es meine.

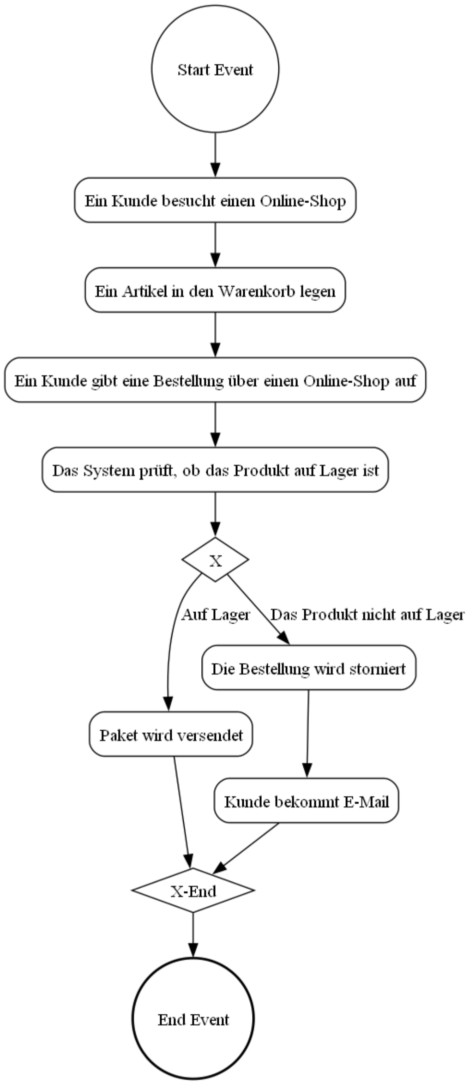

Figure 8.21: Generated output User 4

8.3.5 Interview 5

Speaker 2 (0:02) Willkommen und danke, dass du an diesem User-Test teilnimmst. Ziel dieser Sitzung ist es, die Benutzerfreundlichkeit und technische Genauigkeit eines Chatbots zu evaluieren, der automatische BPMN-Diagramme aus natürlichsprachlichen Beschreibungen von Geschäftsprozessen erstellt.
(0:17) Du wirst eine kurze Aufgabe erhalten, in der du mit dem Chatbot interagierst, um ein Geschäftsprozess zu modellieren.
(0:25) Dabei bitte ich dich laut zu denken, also auszusprechen, was du als einfach, unklar oder überraschend empfindest, während du interagierst. Das hilft mir dabei, zu verstehen, wie klar und hilfreich die Benutzeroberfläche die Antworten des Chatbots sind.
(0:41) Nachdem du die Aufgabe abgeschlossen hast, wirst du gebeten, einen kurzen Fragebogen auszufüllen. Die gesamte Sitzung dauert etwa 15–30 Minuten, wenn du bereit bist, beginnen wir mit der Aufgabe.

Speaker 1 (0:52) Okay, ja, ich bin bereit, okay, type a message, okay, hallo, ah, okay, der schreibt was.
(1:11) Nee, ich will nicht mehr lernen, okay, jetzt kann ich starten, also mein Szenario ist, Kunde
(1:22) besucht einen Onlineshop, Stellung auf, sobald die Bestellung eingegangen ist, prüft das
(1:28) System, okay, wenn das Produkt auf Lage ist, okay, und wenn es nicht auf Lage ist, okay,
(1:36) und dann Ende, okay, jetzt kann ich dann wählen, ich kann noch was wählen, ne, okay, ja, nee.
(1:48) das ist, glaube ich, erstmal eine Aufgabe, okay, beschreiben, Kunde besucht, okay, ja, jetzt fragt er mich wieder, was passiert, okay, auf, okay, okay, nochmal, ja, warte.
(2:24) mal, ist das jetzt schon das Gateway, ist das das, fängt jetzt das Gateway an, oder ist das noch eine Aufgabe, warte, nee, nee, ich glaube, das ist noch eine Aufgabe.
(2:54) System prüft, kann ich jetzt, ah, jetzt kann ich auch ein Gateway auswählen, okay, okay.
(3:04) Erster Schritt in den Gateway fahren, es wird verpackt, wird verpackt, wenn, auf Lager, okay, und es wird versendet, das ist dann ja noch eine Aufgabe, wird versendet, gut, und der Prozess ist
(3:42) beendet, ja, okay, das war jetzt eine Aufgabe, jetzt ist das erste, den anderen.
(4:07) Bestellung storniert, wann, nicht auf Lage, ah, und der Kunde, ja, ja, jetzt ist der Prozess beendet,
(4:41) okay, geht nicht, das geht gerade noch nicht, warum, okay, dann mache ich erstmal Gateway Ende, und jetzt, jetzt, ah, jetzt kann ich den Prozess beenden, okay, ja, das ist es ja, glaube ich.
(5:05) Er macht nichts, ah, okay, es sendet jetzt, ja, macht irgendwas, ah, okay, ah, jetzt, ja, Diagramm start, okay, okay, ja, okay, und jetzt haben wir, siehst du das Diagramm mit deinen Eingaben.

Speaker 2 (5:49) Genau, danke. Klicken Sie hier bitte auf Nein. Es gibt nämlich jetzt eine neue Aufgabe.
(5:55) Und zwar bitte nimm Änderungen am Geschäftsprozess vor und füge ein Element ein, nach „ein Kunde besucht einen Online-Shop“. Und füge bitte das Element „der Kunde legt einen Artikel in den Warenkorb“ hinzu.

Speaker 1 (6:27) Also will ich Änderungen machen, ah, ja, ja, okay, er macht wieder was, ah, okay, ID, ja, und warte mal, wo soll ich das machen.
(6:51) nach „ein Kunde besucht einen Online-Shop“, gut, das habe ich „Kunde besucht Online-Shop“ genannt, ah, okay, also 5, 0, 2, was will ich in meiner hier hinzufügen, was will ich hinzufügen, bitte füge eine Aufgabe, legt Artikel in den Warenkorb, okay, ah, jetzt rechnet er wieder irgendwas.
(7:45) ja, dauert ganz schön, okay, ah, jetzt kriegt er was, ah, ja, er hat das hinzugefügt, okay, ja, gut, okay, danke, sagt das Programm, okay, ja, ist gut.

Speaker 2 (8:11) Genau, jetzt sind wir am Ende des Eingabe-Prozesses, dann würde ich dich noch bitten, eine Bewertung vorzunehmen.
(8:19) Bitte bewerte die folgenden 4 Fragen auf einer Skala von 1 bis 5, wobei 1 ist „Stimme überhaupt nicht zu“ und 5 „Stimme voll und ganz zu“.
(8:34) Die Hinweise des Chatbots waren klar und leicht verständlich.

Speaker 1 (8:41) 4, ja, 4 würde ich sagen.

Speaker 2 (8:45) Das vom Chatbot generierte BPMN-Diagramm hat den von mir beschriebenen Prozess korrekt abgebildet.

Speaker 1 (8:53) Ja, also war ein einfacher Prozess, aber 5, also war richtig.

Speaker 2 (8:59) Es war einfach, mit dem Chatbot zu interagieren und die Aufgaben abzuschließen.

Speaker 1 (9:06) Ich würde auch sagen eine 4, das ist eine 4, vielleicht eine 3, 3 oder 4.

Speaker 2 (9:15) Insgesamt bin ich mit meiner Erfahrung mit dem Chatbot zufrieden.

Speaker 1 (9:19) Ja, auch eine 4 würde ich sagen, genau.

Speaker 2 (9:23) Die letzte Frage ist eine offene Frage, bitte antworte ganz frei. Welche Verbesserung würdest du vorschlagen, um den Chatbot benutzerfreundlicher zu machen?

Speaker 1 (9:43) Also der rechnet mir ein bisschen lange, das finde ich irgendwie nervig.
(9:55) Und das wiederholt sich schon relativ häufig, also ich habe das Gefühl, der fragt immer das Gleiche.
(10:17) Genau, aber sonst kann ich halt mit dem ganz normal chatten, das ist gut.

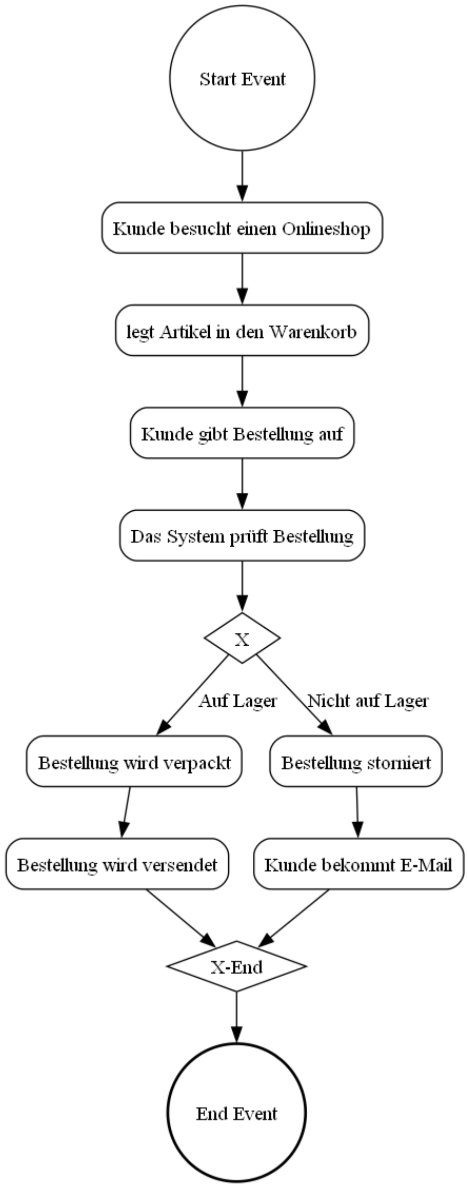

Figure 8.22: Generated output User 5

Erklärung zur selbständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit — bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.
Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: ____________________

Vorname: ____________________

dass ich die vorliegende Masterarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Chatbot-Assisted Business Process Modelling - From User Interaction to BPMN Diagrams

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der Masterarbeit ist erfolgt durch:

Ort	Datum	Unterschrift im Original

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, Julia Caroline Seufert, dass ich die vorliegende Masterarbeit mit dem Thema:

Chatbot-Assisted Business Process Modelling - From User Interaction to BPMN Diagrams

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

[redacted]

Ort	Datum	Unterschrift im Original