Business process automation through ChatBots implementation: A case study of an IT service process at Philips

by

Siying Han

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday, October 10, 2019, at 15:00.

Student number:	4735056	
Project duration:	March, 2019 – October, 2019	
Thesis committee:	Prof. dr. ir. M.F.W.H.A. Janssen,	TU Delft, chair
	Dr. ir. R. Binnekamp,	TU Delft, supervisor
	S.H. van Engelenburg,	TU Delft, supervisor

An electronic version of this thesis is available at .

Preface

THE thesis is the final part of the degree of MSc Construction Management and Engineering (CME) at Delft University of Technology (TU Delft). The CME courses are given at three faculties: the Faculty of Civil Engineering and Geosciences (CEG), the Faculty of Architecture (AR), and the Faculty of Technology, Policy and Management (TPM). And I choose to complete my thesis at TPM.

I enjoyed the new life and study atmosphere in TU Delft as a foreign student, and I looked forward to having an opportunity to work in companies in the Netherlands when I did my first-year master courses. I luckily got one internship opportunity in Philips IT Governance and started the internship from September 2018. The internship assignment is to improve the KPI reporting process by designing a digital system. The internship went well and I began to think about the thesis topic from the end of 2018. Having a good experience with the internship in Philips IT Governance, I still wanted to continue my research in Philips and the IT field. So I communicated with my manager at that time, and he helped me to get in touch with a newly established team about improving IT service in Philips IT Global Services (ITGS), and I might explore the research direction there.

Defining the research topic is difficult. Thanks to managers Maarten Colenbrander, Niels Schreuders, and Marc Hendrikx, they are patient to introduce the whole project and related problems to me. And I noticed that ChatBot implementation in the Call Management process is an urgent problem and it might be the thesis topic. I then contacted professor Marijn Janssen (I joined the course MOT1531 Business Process Management and Technology from professor Marijn and inspired a lot from the course) to be my thesis supervisor from the university. Doing the thesis is full of challenges and surprises, and I am happy to acquire knowledge and abilities from the whole journey.

I would like to express my gratitude to my thesis committee: professor Marijn Janssen, Ruud Binnekamp, and Sélinde van Engelenburg. Professor Marijn guides me with the direction of the research and supports me with insights and literature. Ruud and Sélinde help me with the thesis structure and provide me valuable feedback.

Besides, I would give thanks to everyone at Philips who assisted me during the internship. Especially to Maarten Colenbrander, Niels Schreuders, Marc Hendrikx, and Sergio Paya for being my managers and help me to expand all research activities in Philips. And thanks to the trainees in Philips IT: Reshmi Sarkar, Mark Maas, Corne Matthijssen, etc. for supporting my life and work during the one-year internship in Philips.

Last but most essential, I want to send thanks to my beloved family, who provide me with all the support to go abroad and pursue my master's in TU Delft. And thanks to Yilei Tian for being always there and his understanding, patience, and support.

Siying Han
Delft, October 2019

Summary

NOWADAYS, the term "automation" is gaining extensive interest in both industrial and business fields. There is also a trend of achieving deeper automation in the business processes, which is a part of Business Process Automation (BPA). Among various BPA technologies, Robotic Process Automation (RPA) has attracted increasing research attention. RPA is an emerging form of business process automation technology. The practice of performing RPA results in the deployment of attended or unattended software agents to an organization's environment. ChatBots, as the outcome of the development of artificial intelligence (AI) technology, has been regarded as a potential and powerful tool to achieve such cognitive automation in service-oriented business processes. However, limited knowledge on the design road-map and approaches of ChatBot implementation in the Business process hinders the ChatBots further deployment into company practical operation.

Herein, the research is focused on the question:

How should ChatBots be implemented in service-oriented processes?

Sub-questions are extended as:

- **Q1: What is the value of BPM theory for process redesign?**
- **Q2: What is the state of art of ChatBots?**
- **Q3: What are the current methods of implementing ChatBots in business processes?**
- **Q4: What should the ChatBot-implementation methodology look like?**
- **Q5: What are the values and limitations of the methodology in practice?**

Chapter 1 (introduction) contains the introduction of this study, including the background, problem statement, research question, and thesis structure.

Chapter 2 (literature review) describes the background of business process management (BPM), ChatBot technology, and the current state of ChatBot-implementation in business processes. BPM holds great potential for significant cost saving and productivity improvement, and the lifecycle of BPM serves as one of the guidelines for ChatBot-implementation methodology designed in the next chapter. ChatBots can be regarded as one business process automation tool, the main components are *Intent*, *Entities*, and *Dialog*. ChatBots have strengths such as 24/7 customer service, automation of communication, time and cost savings and relevant offers based on user preferences, etc. However, ChatBots also have weaknesses such as investments in IT infrastructure, lack of experience and understanding, biased personalized information, Social isolation concerns, etc. Regarding classification, ChatBots can be classified by intent: Pattern Matching, Algorithm and Neural Network, or by particular use: Support ChatBots, Skill ChatBot, and Assistant ChatBot. IBM Watson Assistant, Amazon Lex, Google Dialogflow, and Microsoft Azure are four main vendors for ChatBot-implementation in organizations.

Chapter 3 (methodology design) introduces the requirements, design strategy, description approach, and validation and evaluation of designing a ChatBot implementation methodology. The requirements of methodology design are described according to the limitation of the current ChatBot-implementation methodology. Then the methodology design strategy is proposed. The methodology can be developed based on the BPM lifecycle, which ensures the business processes covered by the BPM lead to consistently positive outcomes and delivers maximum value to the organization, together with the Quark ChatBot implementation methodology. Following is the approach to describe a methodology. It is crucial to depict the methodology with clear logic. The analytical framework of Sol can be regarded as a suitable approach to describe the ChatBot implementation methodology. The framework consists of three parts namely way of thinking, working, and modeling. Methodology validation and evaluation are described at last. To check the validity and evaluate the methodology, it is necessary to use a case study to execute the methodology. By employing the method of case study, the methodology can be practically performed to automate a process by implement ChatBots. Accordingly, the validity of methodology can be examined, and the limitation of the methodology can also be explored, laying the foundation for the further improvement of the methodology.

Chapter 4 (ChatBot-implementation methodology) designs the ChatBot-implementation methodology according to the design strategy and description approach in Chapter 3. It follows the analytical framework of Sol (1982). The way of thinking lays the fundamental principles and the underlying structure that the design methodology is based on. It begins with a clear vision of the target processes in which this process automation strategy can be applied. The goal of this methodology is to decrease the process complexity, to enhance the interaction experience of customers, and to increase the efficiency of service agents within the process. The way of working describes the routine from the current situation towards the new situation. It is based on the problem-solving process (Dumas, 2013) and the BPM lifecycle (Janssen, 2002). The ChatBot-implementation cycle consists of seven interactive activities (process identification, process discovery, process analysis, ChatBot implemented process redesign, evaluation, process implementation, and process monitoring and controlling) divided into an understanding phase, a design phase, and an implementation phase. The way of modeling states the approach of depicting the real situation with an abstract model that is suitable for the systematical analysis. The purpose of modeling is to reduce the complexity of a problem situation, to increase the understanding of the dynamic interactions between objects, and to assess the impact of changes. The Business Process Model and Notation (BPMN) was applied then.

Chapter 5 (methodology evaluation) evaluates the ChatBot-implementation methodology designed in Chapter 4. The *Call Management* process of Philips IT Global Services was chosen as an example case to apply the ChatBot-implementation methodology to obtain an automated process. Methodology evaluation starts with a brief introduction to Philips and its goal of process improvement including process automation. The *Call Management* process then was automated by following the routine described by the way of working in Section 4.2. *Process identification* introduced the goal, architecture, and major functions. The as-is process model then was illustrated in *process discovery*. After that, the process was analyzed to obtain facts and issues of the current process in the *process analysis* part. Then, *redesign of the process* was conducted, containing function evaluation, ChatBot selection, defining exception human operations, designing the conceptual framework, and to-be process modeling. The to-be process model, as well as the as-is model, was evaluated in *Evaluation* part. *Process implementation* and *process monitoring and controlling* have not been carried out due to the reality of the company and the duration of the thesis. The evaluation verified the methodology is feasible for the service-oriented process. Nevertheless, it also shows some limitations of the proposed methodology. The methodology does not provide the method of the to-be model converting into an executable model for the implementation. Furthermore, how to translate a process model to a commercial ChatBots dialog model remains to be explored.

In Chapter 6, conclusions are drawn. Research findings answer the previously mentioned research questions. The major contribution of this research is establishing a theoretical ChatBot-implementation methodology for service-oriented process automation. Although, there still are some limitations of the work. 1) The approach of translating the to-be process model to the executable process model for Process implementation is not given in this research. 2) The methodology currently does not take the impact on and from the other related process into consideration. 3) Process monitoring and controlling strategies are hard to determine without process implementation. 4) Strategies to deal with ChatBot-related errors are not provided in this methodology. 5) The implementation of ChatBots can be associated with investments in IT infrastructures and ChatBot facilities. 6) The implementation of ChatBots can lead a big change to the process and user acceptance is not discussed in this research. 7) Information security and data protection are not discussed in the research. For further research on ChatBot-implemented process automation, efforts can be focused on the executable process model, the business process model towards the ChatBot model, the extension of methodology application, the multi-process ChatBot implementation methodology, and ChatBot-based business process management.

Contents

List of Figures

List of Tables

LISTINGS

List of Abbreviations

ABPMP	Association of Business Process Management Professionals
AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
ARIS	Architecture of Integrated Information Systems
BP	Business Process
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
BPMS	Business Process Management Software
DX	Digital Transformation
EPCs	Event-driven Process Chains
I2M	Idea to Market
IETF	Internet Engineering Task Force
IT	Information Technology
ITGS	IT Global Services
ITIL	Information Technology Infrastructure Library
ITSM	IT service management
M2O	Market to Order
ML	Machine Learning
O2O	Order to Cash
OMG	Object Management Group
PBS	Philips Business System
PEPF	Philips Excellence Process Framework
RFC	Request for Change

RPA Robotic Process Automation

SIPs Service Improvements

SLA Service Level Agreement

SNOW Service Now Ticketing Tool

SHA Stakeholder Analysis

KPIs Key Performance Indicators

UML Unified Modeling Language

WfMC Workflow Management Coalition

WSFL Web Services Flow Language

XPDL XML Process Definition Language

List of Definitions

AI-based ChatBot ChatBot that is supported by machine learning, it can understand the context and intents of questions.

As-is model The model of the current process.

Business process automation The automation of complex business processes and functions beyond conventional data manipulation and record-keeping activities, usually through the use of advanced technologies.

Business process management A disciplined approach to identify, design, execute, document, measure, monitor, and control both automated and non-automated business process to achieve consistent, targeted results aligned with an organization's strategic goals.

ChatBot ChatBot is a computer program that is designed to communicate with human users through the internet. In this report, ChatBot also refers to a computer program that will perform automatic tasks.

ChatBot-implementation methodology The designed methodology of process automation through ChatBot implementation.

Call Management process A process in Philips IT Global Services. It is a service-oriented process providing IT support to employees of the company.

Knowledge-based ChatBot ChatBot that is trained by an amount of data or by a knowledge base.

Methodology The designed ChatBot-implementation methodology.

Rules-based ChatBot ChatBot that is powered by a series of defined rules.

Service desk A service desk is a communication center that provides a single point of contact between a company and its customers, employees, and business partners. The purpose of a service desk is to ensure that users receive appropriate help in a timely manner.

To-be model The model of the redesigned process.

1 Introduction

If you can't measure it, you can't improve it.

Peter Drucker

This chapter introduces the research background and problem statement. Waves of automation and artificial intelligence have triggered a trend of cognitive automation of business processes by the ChatBot technique. In this chapter, the background of process automation with ChatBots is firstly stated in Section 1.1. And the problem of current ChatBot implementation is briefly discussed in the Section 1.2. Section 1.3 lists the research questions of the thesis. And the structure of the report is presented in Section 1.4.

1

1.1. BACKGROUND

NOWADAYS, the term "automation" is gaining extensive interest in both industrial and business fields. Automation has been already implemented in the manufacturing sector, specifically now in the form of “Industry 4.0” technology[1, 2].

There is also a trend of achieving deeper automation in the business processes, which is a part of Business Process Automation (BPA)[3]. BPA has been widely employed in Business Process Management (BPM) to standardize and simplify processes, achieve digital transformation, increase service quality, and improve service delivery[4]. Among various BPA technologies, Robotic Process Automation (RPA) has attracted increasing research attention[5, 6]. RPA is an emerging form of business process automation technology. The practice of performing RPA results in the deployment of attended or unattended software agents to an organization’s environment. These software agents, or robots, are deployed to perform pre-defined structured and repetitive sets of business tasks or processes[7]. Currently, cognitive RPA, which uses more advanced technologies, such as natural language processing (NLP), text analytics, data mining, semantic technology, and machine learning, lead a gateway for the adoption of software agents or artificial intelligence in business environments[8, 9].

ChatBots, as the outcome of the development of artificial intelligence (AI) technology, has been regarded as a potential and powerful tool to achieve such cognitive automation in service-oriented business processes[10–12]. However, limited knowledge on the design road-map and approaches of ChatBot implementation in the Business process hinders the ChatBots further deployment into company practical operation. Herein, a methodology of ChatBot-implementation in service-oriented business processes is proposed in this thesis. The methodology sheds a light on the ChatBot-implementation framework with concerns of business process management.

1.2. PROBLEM STATEMENT

SERVICE-ORIENTED processes always face with a huge number of human interactions and involvements[13, 14]. These business processes are an integration of systems, services (including the third-party service), and humans. Such fact undoubtedly increases the complexity and difficulty of the process automation. Therefore, cognitive robots, or ChatBots, are gradually getting involved in the service-oriented process automation. Commercial ChatBot suppliers, such as IBM, Google, and Amazon, have released their frameworks to build ChatBots. However, these frameworks are lack of flexibility and may cause some conflict when ChatBots need to interact with a third-party service[15]. Hence, they are not suitable to directly applied in the service-oriented process.

Therefore, a methodology is thus needed to guide the implementation of ChatBot into service-oriented processes. The methodology shall combine the theories of process redesign in BPM and ChatBots engineering principles.

1.3. RESEARCH QUESTION

THE objective of this research is to establish a ChatBot-implementation methodology for the service-oriented process automation. The main research question can be described as:

How should ChatBots be implemented in service-oriented processes?

As mentioned above, the methodology shall be based on both BPM and ChatBot engineering theories. Hence, the following sub-questions need to be answered first:

- **Q1: What is the value of BPM theory for process redesign?**
  - Q1.1: What is the lifecycle of BPM?
  - Q1.2: How to model a process?
  - Q1.3: What are the criteria for the process automation (RPA)?
- **Q2: What is the state of art of ChatBots?**
  - Q2.1: What is a ChatBot?
  - Q2.2: What are the classification of ChatBots?
  - Q2.3: What are the main vendors for enterprise ChatBot service?
- **Q3: What are the current methods of implementing ChatBots in business processes?**
  - Q3.1: What are the current exiting methodologies?
  - Q3.2: What are limitations or drawbacks of these methodologies?

These questions are mainly knowledge-related. Hence, literature reviews would be conducted to answer them. Then, the sub-question of the methodology design should be dealt with:

- **Q4: What should the ChatBot-implementation methodology look like?**
  - Q4.1: What is the principle to design a methodology (way of thinking)?
  - Q4.2: What are the steps of the methodology? (way of working)
  - Q4.3: How to model the process? (way of modeling)

It is necessary to verify the proposed methodology in a real case. Hence, the following sub-question should be answered:

- **Q5: What are the values and limitations of the methodology in practice?**
  - Q5.1: What type of process can use the methodology?
  - Q5.2: How to carry out all the steps of the methodology?
  - Q5.3: What are the limitations and contributions of the methodology?

A case study on a real process can be the solution to these questions.

1

1.4. THESIS STRUCTURE

THE thesis includes five parts: *Introduction, Literature Review, Methodology Design, ChatBot Implementation Methodology, Methodology Evaluation,* and *Conclusion*. Figure 1.1 presents the thesis structure.

The introduction part introduces the research background and problem statement. Waves of automation and artificial intelligence have triggered a trend of cognitive automation of business processes by the ChatBot technique. In this chapter, the background of process automation with ChatBots is firstly stated in Section 1.1. And the problem of current ChatBot implementation is briefly discussed in the Section 1.2. Section 1.3 lists the research questions of the thesis. And the structure of the report is presented in Section 1.4.

The literature review part describes the background of business process management (BPM), ChatBot technology, and the current state of ChatBot-implementation in business processes. The aim of the literature review is to lay a theoretical foundation for the ChatBot implementation in Business process methodology. The establishment of the methodology is based on principles in the field of business process management and facts of ChatBot technology, and stem from the methodology for ChatBot design. The business process management concepts, including business process (BP), business process management (BPM), business process modeling, business process automation (BPA), and robotic process automation (RPA), are introduced in Section 2.1. Then, Section 2.2 gives a brief overview of the ChatBot technology, including its development history, architecture, classification, and main vendors providing ChatBot service. In the end, methodologies, which are proposed for ChatBot design, establishment, and application, are briefly introduced in Section 2.3. The limitations of those methodologies are also briefly discussed.

The methodology design part introduces the requirements, design strategy, description approach, and validation and evaluation of designing a ChatBot implementation methodology. Section 3.1 shows the requirements that the designed methodology needs to fulfill according to the limitation of the current methodology in Section 2.3. The methodology design strategy is described in Section 3.2. Section 3.3 introduces an approach to describe a methodology: ways of thinking, working, and modeling in the analytical framework. How to validate and evaluate the designed methodology is described in Section 3.4 at last.

The ChatBot-implementation methodology part designs the ChatBot-implementation methodology according to the design strategy (Section 3.2) and description approach (Section 3.3). The way of thinking introduces in Section 4.1. It guides in representing a system in the form of a model. Section 4.2 concerns the steps that need to be taken to cope with the specific characteristics in ChatBot-implementation. The ChatBot-implementation cycle consists of seven interactive activities divided into an understanding phase, a design phase, and an implementation phase. And Section 4.3 regards the choice of modeling techniques and the construction of models during a study of Curtis (1992). The purpose of modeling is to reduce the complexity of a problem situation, to increase the understanding of the dynamic interactions between objects, and to assess the impact of changes.

The methodology evaluation part is the evaluation of the designed ChatBot-implementation methodology in Chapter 4. *Call Management* process in Philips IT Global Services is chosen as an example case to apply the ChatBot-implementation methodology in order to obtain an automated process. The *Call Management* process is a service-oriented process providing IT support to employees of the company, which is suitable for applying the proposed methodology to achieve ChatBot-based process automation. The evaluation starts with a case introduction on Philips and its goal of process improvement including process automation (Section 5.1). The *Call Management* process then is automated by following the routine described by the way of working (Section 4.2). Section 5.2 is the Process identification, in which the goal, architecture, major functions are introduced. The as-is process model then is illustrated in the Process discovery (Section 5.3). After that, the process is analyzed to obtain facts and issues of the current process (Section 5.4). Then, redesign of the process is conducted in Section 5.5, containing functions evaluation, ChatBots selection, defining exception human operations, designing the conceptual framework, and to-be process modeling. The to-be process model, as well as the as-is model, are evaluated in Section 5.6. At last, conclusions together with some recommendations on the process implementation, process monitoring and controlling are provided in Section 5.7.

The last part is the conclusion. Research findings, contributions and limitations, and outlooks are briefly analyzed in the conclusion.

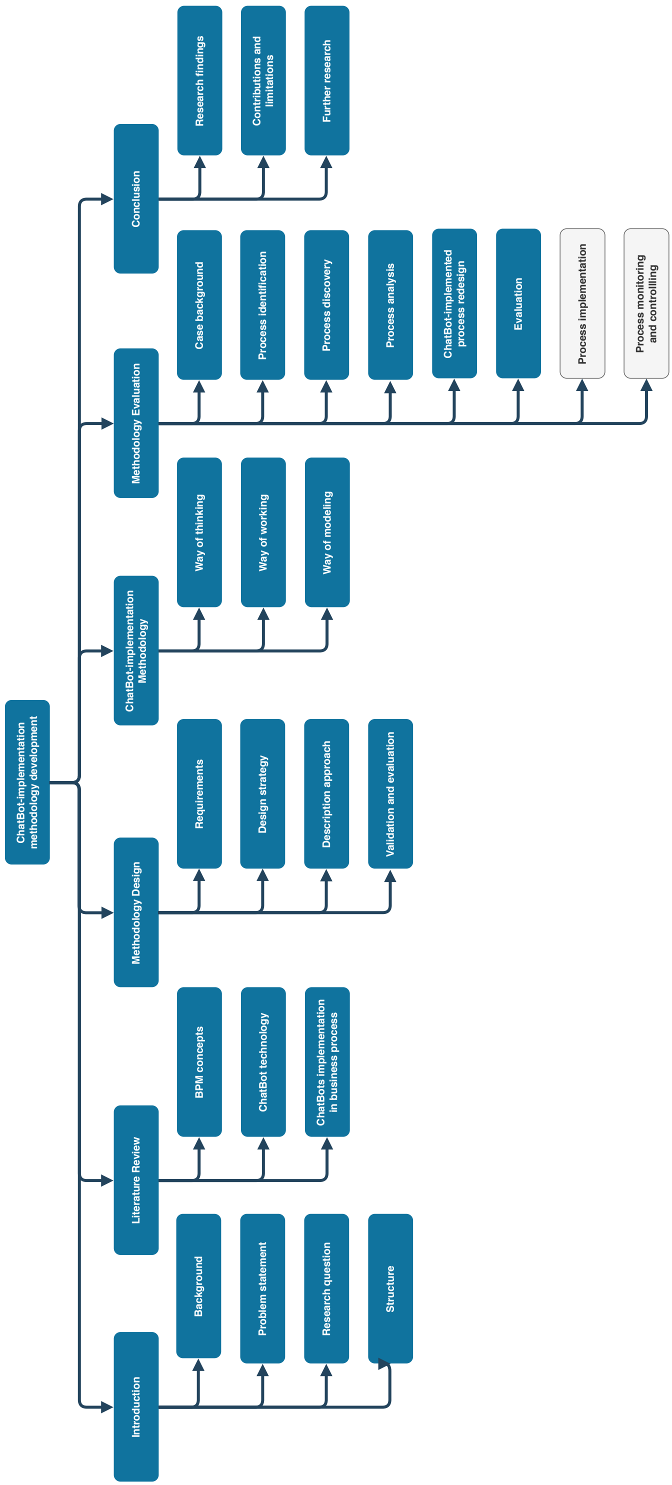


Figure 1.1: Thesis Structure

REFERENCES

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, *The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0,* IEEE industrial electronics magazine **11**, 17 (2017).

[2] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, *Industry 4.0: The future of productivity and growth in manufacturing industries,* Boston Consulting Group **9**, 54 (2015).

[3] A. H. Ter Hofstede, W. M. Van Der Aalst, M. Adams, and N. Russell, *Modern Business Process Automation: YAWL and its Support Environment*, edited by A. H. M. Hofstede, W. M. P. Aalst, M. Adams, and N. Russell (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 1–676.

[4] W. M. P. van der Aalst, M. La Rosa, and F. M. Santoro, *Business Process Management,* Business & Information Systems Engineering **58**, 1 (2016).

[5] R. K. Ko, *A computer scientist's introductory guide to business process management (bpm),* XRDS: Crossroads, The ACM Magazine for Students **15**, 4 (2009).

[6] N. Ayachitula, M. Buco, Y. Diao, S. Maheswaran, R. Pavuluri, L. Shwartz, and C. Ward, *It service management automation-a hybrid methodology to integrate and orchestrate collaborative human centric and automation centric workflows,* in *IEEE International Conference on Services Computing (SCC 2007)* (IEEE, 2007) pp. 574–581.

[7] L. P. Willcocks, M. Lacity, and A. Craig, *The it function and robotic process automation,* (2015).

[8] R. Ramamurthy and R. M. Cillanki, *Intelligent, interactive, and self-learning robotic process automation system,* (2017), uS Patent App. 15/614,646.

[9] I. A. Hameed, *Using natural language processing (nlp) for designing socially intelligent robots,* in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (IEEE, 2016) pp. 268–269.

[10] K. Chung and R. C. Park, *Chatbot-based heathcare service with a knowledge base for cloud computing,* Cluster Computing **22**, 1925 (2019).

[11] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, *A new chatbot for customer service on social media,* in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (ACM, 2017) pp. 3506–3510.

[12] M.-C. Jenkins, R. Churchill, S. Cox, and D. Smith, *Analysis of user interaction with service oriented chatbot systems,* in *International Conference on Human-Computer Interaction* (Springer, 2007) pp. 76–83.

[13] A. V. Hill, D. A. Collier, C. M. Froehle, J. C. Goodale, R. D. Metters, and R. Verma, *Research opportunities in service process design,* Journal of Operations Management **20**, 189 (2002).

1

[14] J.-S. Chen and H.-T. Tsou, *Performance effects of it capability, service process innovation, and the mediating role of customer service,* Journal of Engineering and Technology Management **29**, 71 (2012).

[15] P. R. Telang, A. K. Kalia, M. Vukovic, R. Pandita, and M. P. Singh, *A conceptual framework for engineering chatbots,* IEEE Internet Computing **22**, 54 (2018).

2

Literature Review

When everything gets answered, it's fake.

Sean Penn

This chapter describes the background of business process management (BPM), ChatBot technology, and the current state of ChatBot-implementation in business processes. The aim of the literature review is to lay a theoretical foundation for the ChatBot implementation in Business process methodology. The establishment of the methodology is based on principles in the field of business process management and facts of ChatBot technology, and stem from the methodology for ChatBot design. The business process management concepts, including business process (BP), business process management (BPM), business process modeling, business process automation (BPA), and robotic process automation (RPA), are introduced in Section 2.1. Then, Section 2.2 gives a brief overview on the ChatBot technology, including its development history, architecture, classification, and main vendors providing ChatBot service. In the end, methodologies, which are proposed for ChatBot design, establishment, and application, are brief introduced in the Section 2.3. The limitations of those methodology are also briefly discussed.

2.1. BUSINESS PROCESS MANAGEMENT CONCEPTS

THIS section introduces the concepts of business process, business process management, and several different business process modeling tools.

2.1.1. BUSINESS PROCESS

A business process (BP) consists of a set of related and structured activities or tasks that are formed in coordination in an organizational environment. Normally, a business process is triggered by a single organization, but it may interact with business processes performed by other organizations[1, 2]. Using business process can improve customer satisfaction as well as improve the agility for reacting to the changes in market[3].

2.1.2. BUSINESS PROCESS MANAGEMENT

Business Process Management (BPM) can be described as, based on the definition of the Association of Business Process Management Professionals (ABPMP)[4], a disciplined approach to identify, design, execute, document, measure, monitor, and control both automated and non-automated business processes to achieve consistent, targeted results aligned with an organization's strategic goals. BPM aims to provide and/or focus on the deliberate, collaborative and technology-aided definition, improvement, innovation, and management of end-to-end business processes that drive business results, create value, and enable an organization to meet its business objectives with more agility. With the assistance of BPM, an enterprise can align its business processes to its business strategy, leading to effective overall company performance through improvements of specific work activities either within a specific department, across the enterprise, or between organizations.

BPM can be viewed as a continuous cycle as shown in the Figure 2.1, which comprises the following phases[2]:

- **Process identification:** A business problem is posed in process identification. The problem-related processes are identified and delimited. After conducting a process identification, a new process architecture is provided showing the overall view of the processes in an organization.
- **Process discovery:** (or as-is process modeling). The current state of each relevant process is documented in process discovery. After conducting a process discovery, one or several as-is process models are formed.
- **Process analysis:** The issues that are associated with the as-is models are identified in process analysis. After conducting a process analysis, a structured collection of issues are presented and normally prioritized in terms of their impacts or the estimated efforts required to resolve them.
- **Process redesign:** (or process improvement). The changes to the process that would be helpful to address the issues identified in process analysis are identified in process redesign. After conducting a process redesign, change options are proposed, analyzed, and compared. Process redesign and process analysis need to go hand-in-hand, which means the newly proposed change options need to be analyzed by process analysis methods. The most promising change options leading to a redesigned process are summarized in the end. After conducting a process

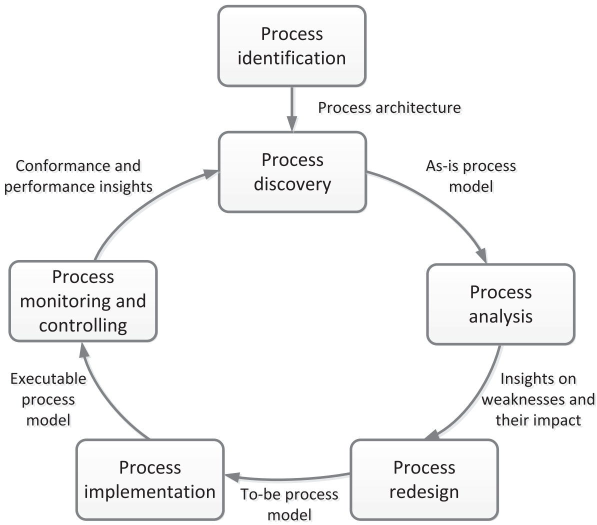


Figure 2.1: BPM Lifecycle[2]

redesign, a to-be process model is drawn. The to-be model also serves as a basis for the next step.

- **Process implementation:** The changes that need to be moved from the as-is model to the to-be model are prepared in process implementation. Organizational change management and process automation are two aspects of process implementation. Organizational change management
- **Process monitoring and controlling:** Process monitoring and controlling collects the relevant data in the running redesigned process, and determines the process performance. Bottlenecks, recurrent errors or deviations concerning the intended behavior are identified and corrective actions are undertaken. This phase may arise new issues, and the lifecycle should be repeated to solve the new issues. Efforts can be employed to solve issues or further improve the process.

2.1.3. BUSINESS PROCESS MODELING

Business process modeling is an analytical representation or illustration of the business process. It offers support for the analysis, improvement, and automation of current processes. Flowcharts, programs, hypertext, or scripts are major approaches that the business process modeling uses to express the business process. Currently, hundreds of market-available business process modeling tools have been proposed and published. By employing these tools, users are capable to model business processes, implement, execute and refine models[5]. Here is an introduction of the some most commonly used business process modeling tools, including Business Process modeling Notation (BPMN), Unified Modeling Language Activity Diagrams (UML ADs), Event-driven Process Chains (EPCs), and Web Services Business Process Execution Language (WS-BPEL).

2

Business Process modeling Notation (BPMN)

BPMN aims to provide users an easily understandable and graphical notation to represent complex process semantics [6]. It can work as a front end to various business processes execution methods. Instead of block-structured, BPMN is a graph-structured tool, which enables its support for control-flow dependency specification. Despite facts of its lack of provision, and no standard-organization accepted formalization, BPMN can be regarded as a great move in the path of developing expressive business process modeling language.

Unified Modeling Language Activity Diagrams (UML ADs)

UML ADs is an Object Management Group (OMG) standard, which is originally applied in software engineering [7]. Activity diagram is a behavioral diagram in UML, aiming to capture and depict the dynamic behavior of systems. UML ADs are applied to model sequential and concurrent activities. And its focus on condition and sequence of the flow. The OMG has already supplied a formal semantics of UML ADs yet without an official formalization.

Event-driven Process Chains (EPCs)

EPCs is flow-chart based approach for business process modeling. It is originally proposed for SAP R/3 modeling [8], are now widely used to illustrate business process workflow, after its being included in the Architecture of Integrated Information Systems (ARIS) framework [9]. There is no standardized procedure for EPCs modeling.

Web Services Business Process Execution Language (WS-BPEL, or BPEL)

WS-BPEL, normally known as BPEL, is an executable language which is standardized by OASIS (Organization for the Advancement of Structured Information Standards) [2]. It is used in the web-service business process to define process execution. Typically, a BPEL message can orchestrate process execution, invoke remote services, and manage events and exceptions. BPEL is always associated with BPMN. And BPMN specification provides a mapping from BPMN towards BPEL [10].

2.1.4. Business process automation

Business process automation (BPA), also known as business automation or digital transformation, is defined as the automation of complex business processes and functions beyond conventional data manipulation and record-keeping activities, usually through the use of advanced technologies[11]. It can streamline a business for simplicity, achieve digital transformation, increase service quality, improve service delivery or contain costs. It consists of integrating applications, restructuring labor resources and using software applications throughout the organization[12].

2.1.5. Robotic process automation

Robotic process automation (RPA) is a software-based approach to employ software to partially or fully automate highly-structured and repetitive tasks in business processes. RPA used to rely on basic technologies, such as screen scraping, macro scripts, and workflow automation. However, developments in data science, machine learning, and artificial intelligence trigger an RPA transition to cognitive RPA. Advanced technologies, such as natural language processing (NLP), text analytics, data mining, semantic technology and machine learning (ML), are used to make it possible for software robot complete more complex tasks. These tasks can include queries, calculations, transaction processing, IT management and automated online assistants. Cost reduction, quality improvement, and better compliance are the three main benefits expected from RPA. The substantial benefits of RPA can be attributed to areas including effectiveness, quality, compliance, scalability, risk optimization, and workforce performance[13]. By 2020, 65% of human employee requirements will be reduced by RPA, which will on the other hands build a 1-billion USD RPA market. It is predicted that 40% of large enterprises will adopt RPA solutions by that time [14].

Although the benefits that RPA introduces to the enterprise, not all business process are suitable for RPA. And criteria of business processes which are suitable for RPA have been proposed [13, 15]:

- Processes which have limited requirement of subjective judgment, creativity or interpretation skills;
- Process that are performed frequently, in huge numbers or with significant peaks in workload;
- Processes which need access to multiples applications and systems to accomplish the task;
- Processes that are highly standardized with limited or no exceptions to handle;
- Processes that are prone to human error owning to manual operation;
- Processes that can be broken down into unambiguous rules;
- Processes once started, need limited human intervention.

2.2. ChatBot technology

The word ChatBot is composed of the terms of chat and robot, which was originally used to describe a computer program simulating natural conversational language with the assistance of a dialogue system[16]. A ChatBot generally applies a text-based conversational interaction interface, which is capable to generate responses to natural language input and attempt to mimic a natural human interaction. ChatBots holds the abilities of natural language processing (NLP) and artificial intelligence (AI) to understand users' input and give suitable feedback or response. Currently, ChatBots have been gaining increasing importance of applications in scientific, commercial, and entertainment fields. A wide range of ChatBot-based applications in virtual assistance, e-commerce, and social network are emerging [17–20]. A trend of applying ChatBots in business processes for automation is rising.

2.2.1. ChatBot history

The first ChatBot, which can date back to 1966, was named ELIZA[21]. It was developed at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum. As an early natural language processing computer program, ELIZA simulated conversation by employing a pattern matching and substitution methodology. But it had no built-in framework for contextualizing events.

In 1972, another ChatBot, PARRY[22], was delivered by Kenneth Colby at Stanford

2

University. Different from ELIZA, PARRY implemented a crude model of the behavior of a person with paranoid schizophrenia based on concepts, conceptualizations, and beliefs, which made it a more advanced program.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) [23] was composed by Richard Wallace in 1995. The program was rewritten in Java beginning in 1998. It is one of the strongest programs of its type and has won the Loebner Prize, awarded to accomplished humanoid, talking robots, three times in 2000, 2001, and 2004[24].

Siri[25] was originally released as a stand-alone application for Apple Inc.'s iOS operating system in February 2010. It is an intelligent virtual assistant, which use voice queries and a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Internet services.

IBM Watson[26] is a question-answering computer system developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci. The computer system was initially developed to answer questions on a quiz show. And it was firstly used in commercial application in 2013, for utilization management decisions in lung cancer treatment.

Amazon Alexa[27] is an intelligent personal assistant inhabiting the Amazon Echo device, and it was initially released in November 2014. Alexa uses natural language processing algorithms for voice interaction. It uses these algorithms to receive, recognize and respond to voice commands.

2.2.2. ChatBot architecture

It is only in the last few years that the real power of service automation has been unleashed[28]. And ChatBot is a widely used method for service process automation. ChatBots have been around since the late sixties[1] but become popular in recent years. Two key developments have allowed ChatBots to become a viable business tool and such an in-demand skill[29]:

- **Messengers apps are incredibly popular:** Messengers apps are used by billions of people around the world, and Messenger platforms are even more popular than social media sites.
- **AI is becoming smarter:** Machine learning, deep learning, natural language processing, and artificial intelligence have all progressed rapidly over the past few years. ChatBots wouldn't be very useful if they weren't able to understand what the user wants and respond accordingly through these technologies.

ChatBot is a software agent capable of conversing with users through an interface. ChatBot will greet the user and invite them to take some action (like asking it a question) at first. And parse the input and figure out the intention of the user's question after receiving the user replies. Finally, it will respond in a consequential manner, either providing information or asking for further details before ultimately answering the question[30]. Three main components (*Intent*, *Entities*, *Dialog*) behind a ChatBot are used to determine how to interpret the user input and how to respond to it. The *Dialog Skill* (see in Figure 2.2) that contains these three components is essential for a ChatBot[31, 32].

[1] The first ChatBot, Eliza, was rudimentary but it proved the potential of ChatBots even back then.

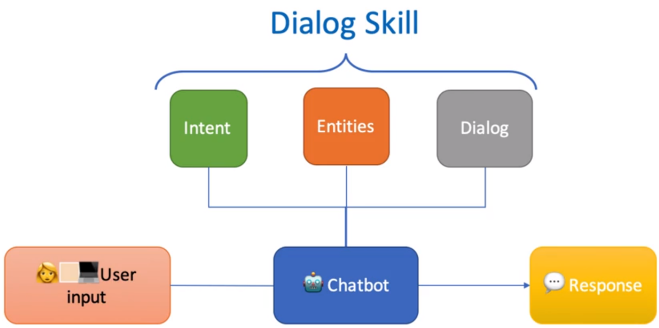


Figure 2.2: Dialog Skill[31]

Intent is the most important component, which captures the intent or goal of the user to determine the user wants. Figure 2.3 shows an example of defining a *greetings* intent. Once the ChatBot is trained on the intents that defined, it will look at the user input and try to determine if any of the intents match the user request[31].

Figure 2.3: Intent Example[31]

Entity is the second key component of dialog skill. Entities allow the ChatBot to capture specific values within the user utterance while intents capture the user goal. Figure 2.4 shows an example of *location* entity. Multiple values for the entity could be defined such as *Toronto*, *Montreal*, *Vancouver*, etc. Synonyms could also be defined within a given entity value[33]. ChatBot can detect the *intent* and *entity* of an user's question (see in Figure 2.5) and provide an appropriate and specific answer to the user then[33].

Dialog is the third component of dialog skill, which allows ChatBot to issue a response to the user based on their intents and the defined entities. A dialog is simply a tree of nodes, and each node will handle one particular scenario. Example in Figure 2.6 contains three nodes: *Welcome* handles the ChatBot prompt, *Greetings* node implement the response to the user greeting, and there is a special fall back node that will notify the user when ChatBot does not understand the user[34].

2.2.3. ChatBots Classification

Application classification

Based on the field that ChatBots are applied in, there are four types ChatBots including *Service ChatBots*, *Commercial ChatBots*, *Entertainment ChatBots* and *Advisory ChatBots*[35].

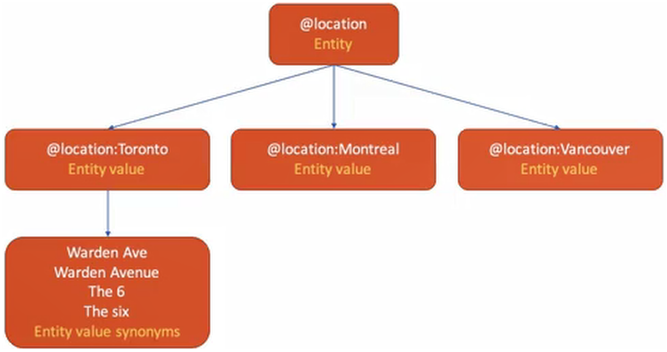


Figure 2.4: Entity Example[33]

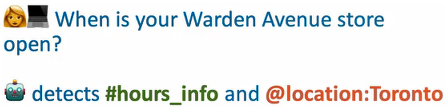


Figure 2.5: Example of Detecting Intent and Entity[33]

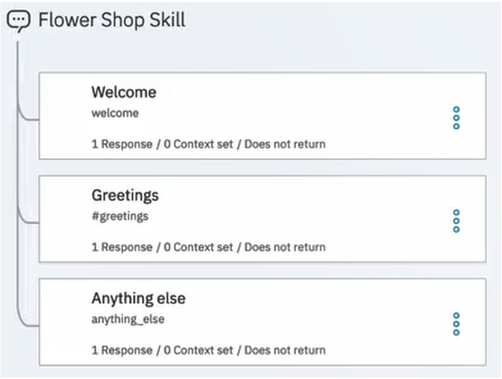


Figure 2.6: Dialog Example[34]

- **Service ChatBot** is designed to provide facilities to customers.
- **Commercial ChatBot** is designed to streamline purchases for customers.
- **Entertainment ChatBot** is designed to keep customers engaged with sports, favorite band, movies or other events. It offers the option of placing bets, detail on upcoming events and ticket deals.
- **Advisory ChatBot** is designed to provide suggestions, give recommendations on service, offer maintenance or repair goods. This type of ChatBots can contact people, offer support and advice tips when it is needed.

Function classification

ChatBots can also be grouped by their function, i.e., *goal-based*, *knowledge-based*, *service-based* and *response generated-based ChatBots*[36].

- **Goal-based ChatBot** is classified based on the primary goal aim to achieve. They are designed for a particular task and setup to have short conversations to get information from the user to complete the task.
- **Knowledge-based ChatBot** is classified based on the knowledge they access from the underlying data sources or the amount of data they are trained on. The two main data sources are open-domain and closed-domain. Open-domain data sources answer depends on general topics and respond appropriately.
- **Service-based ChatBot** is classified based on facilities provides to the customer. It could be a personal or commercial purpose.
- **Response Generated-based ChatBot** is classified based on what action they perform in response generation. The response models take input and output in natural language text. The dialogue manager is responsible for combining response models together. To generate a response, dialogue manager follows three steps. First, it uses all response models to generate a set of responses. Second, returns a response based on priority. Third, if no priority response, the response is selected by the model selection policy.

Business use classification

From a business perspective, ChatBots are categorized into three types including *Support ChatBot*, *Skill ChatBot*, and *Assistant ChatBot*[37].

- **Support ChatBot** Support ChatBot is built to master a single domain. Support ChatBot needs to have *personality*, *multi-turn capability*, and *context awareness* to walk a user through any major business processes and answer a wide range of FAQ-type questions. Personality, multi-turn capability, and context awareness are within support ChatBot. To build a support ChatBot, speech is an optional feature, while developers need to make sure it is easy to navigate the bot and the bot can execute the actions that users care about.

2

- **Skill ChatBot**: Skill ChatBot is typically a more single-turn-type bot, which does not require much contextual awareness. Commands are set intended to make life easier. Typically, speech functionality is recommended for skill ChatBot. They need to follow commands quickly but do not need to worry too much about contextual awareness, as a user will quickly learn to give an appropriately specific command. What's more, integration is important to skill ChatBot. Keep integration simple can make users interact with the ChatBot easily without worrying about the instruction.

- **Assistant ChatBot**: Assistant ChatBot is a middle ground between the above two ChatBot. They work best when they know a little bit about a variety of topics. In addition to being conversational, assistant ChatBot needs to be entertaining as well. In this way, training is important.

2.2.4. CHATBOT SERVICE VENDORS

DIFFERENT from the early development of ChatBots, people and companies nowadays can simply use some commercial services to build up their ChatBots. There are mainly four vendors for ChatBots implementation in organizations including IBM (Watson Assistant), Amazon (Lex), Google (Dialogflow) and Microsoft (Azure Bot).

IBM Watson Assistant[38] can be used to build an own branded assistant (ChatBot) into any device, application, or channel which connects to the customer engagement resources.

Figure 2.7 shows a typical workflow of ChatBots based on IBM Watson Assistant, including the following steps:

- Users can interact with the ChatBot through one or more of these integration points including i) conversational interfaces published directly to an existing social media messaging platform, such as Slack or Facebook Messenger, ii) A simple Chatbot user interface hosted by IBM Cloud, and iii) custom applications.
- The ChatBot receives user input and routes it to the dialog skill.
- The dialog skill interprets the user input further, then directs the flow of the conversation. The dialog gathers any information it needs to respond or perform a transaction on the user's behalf.

Amazon Lex[39] is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are available to any developer, enabling you to quickly and easily build sophisticated, natural language, conversational bots. Amazon Lex democratizes deep learning technologies by putting the power of Amazon Alexa within reach of all developers. Harnessing these technologies, Amazon Lex enables you to define entirely new categories of products made possible through conversational interfaces. As a fully managed service, Amazon Lex scales automatically, so customers do not need to worry about managing infrastructure.

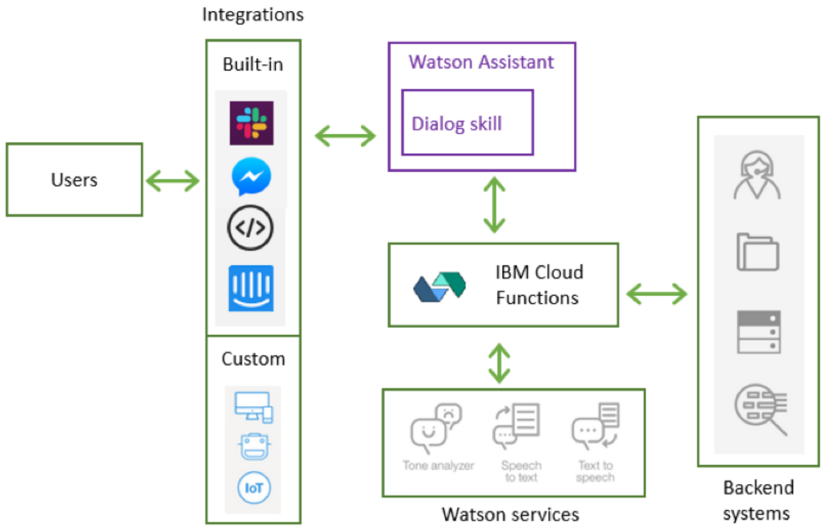


Figure 2.7: Work flow of ChatBots based on IBM Watson Assistant[38]

Google Dialogflow[40] is a developer of human-computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant, a virtual buddy for Android, iOS, and Windows Phone smartphones that perform tasks and answers users' question in a natural language. It has also created a natural language processing engine that incorporates conversation context like dialogue history, location, and user preferences.

Microsoft Azure[41] is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. Azure was announced in October 2008, and released on February 1, 2010, as Windows Azure before being renamed Microsoft Azure on March 25, 2014. Microsoft Azure Bot[42] provides a service of developing intelligent and enterprise-grade ChatBots, ranging from a QA bot to an own-branded virtual assistant. The Azure Bot Service is integrated with Azure Cognitive Services, providing the ChatBots natural language understanding abilities as well as enriched customer support, letting users express their needs through speech and images.

2.2.5. ADVANTAGES AND LIMITATION OF CHATBOTS

The application of ChatBots in the commercial field would change the way of informing, communicating and transacting between the company and its customers or other external stakeholders. As to the internal aspect, it may strongly influence future organizations, communications, and collaborations in the company. The main advantages and challenges of the application of ChatBot have been summarized in Table 2.1.

Owing to ChatBots, it becomes more feasible to have a one-to-one and 24/7 interaction between companies and their customers. Companies can effectively save human

Table 2.1: Advantages and Challenges of ChatBots[43]

	Strengths & Opportunities	Weaknesses & Risks
For providers companies	- 24/7 customer service (anytime/anywhere) - New & direct customer contact points - New method & types of data collection - High amount of personal user/usage data - Personalization & automation of communication - Reduction of service & support costs	- Malfunctioning chatbots & unanswered questions - Investment in IT infrastructure & chatbot tools - Extension of IT & analytics architectures - Lack of awareness & acceptance by users - Information security & data protection - Image & reputation risks
For users/ customers	- 24/7 customer services & support - One-to-one communication on personal device - High convenience & ease of use - Time- & cost-saving - Reduction on relevant information & services - Relevant offers based on user preferences	- Privacy - Data protection of personal & sensitive data - Lack of experience & understanding - Biased personalized information - Artificiaal/non-human conversation - Social isolation & ethical concerns

costs in customer services and get rid of the risk of missing customers' requests outside the office hours. ChatBots also leads to an easier approach for the company to collect and analyze customers' data, based on which the company could have a better understanding of customers and provide better and personalized serveries. However, companies need investments in building and maintaining ChatBots in both hardware and software levels. And the implementation of new technology such as ChatBots would somehow increase the complexity of the company architecture. And more attention needs to be put on information security and data management.

In respect to customers, ChatBots can no doubt save a lot of time and cost on communication with the company, and increase the efficiency of the interaction. Even though, customers may face the risk of misuse of personal information and privacy. And sometimes it may cost extra time and efforts of customers to get used to the ChatBots conversational interface, both technically and emotionally.

2.3. CHATBOTS IMPLEMENTATION IN BUSINESS PROCESS

HUMAN is a key cost factor in today's business processes. To reduce labor, there is a trend of converting people-driven processes to a ChatBot service. A structured and principled manner of ChatBot implementation in a business process can ensure the desired outcome. Before the discussion on the ChatBots implementation method, we firstly focus on the conversation model and multiagent system design, which are the foundations of human-ChatBot interaction design.

A conversation model that enables a conversation flow between the web-service consumers and the web-service providers was proposed by Ardissono et al.[44]. A formal model of conversations by using concepts of commitments and arguments and a Colored Petri Nets (CPN) model were reported by Bentahar et al.[45] and Cost et al.[46], respectively.

As respect to multiagent systems, the Tropos methodology[47] was introduced for building agent oriented software systems, which spans the software development process from early requirements to implementation for agent-oriented software. The Gaia methodology allows developers to design a system by using responsibilities, permission, activities, and protocols[48]. The Comma methodology[49] is a commitment-based approach that enables developers to capture business scenarios using commitments and creates a process that is sound with respect to commitments.

A. Kalia et al. proposed a Quark methodology to transform people-driven processes to ChatBot service[50]. Quark takes a business process model as its input and produces an IBM Watson model of human-ChatBot interactions.The methodology consists of the following steps:

- **M1**: Identify roles served by humans that can be automated
- **M2**: Identify goals of each role
- **M3**: Identify commitments between roles
- **M4**: Produce a set of interactions
- **M5**: Repeat steps **M2** and **M3** to produce additional goals and commitments
- **M6**: Translate the interactions to ChatBots vendor's model

Quark is a well-established methodology to implement ChatBot into business process by using existing business process model. However, there are some limitations of the methodology. The main goal of the methodology is to provide an approach to apply ChatBot into a process to replace human labor. However, it pays less attention on the business process management. The ChatBot implementation is not carried out in the level of the business process. One should keep it in mind that the goal of ChatBot implementation is to enhance process performance and efficiency. The input of the methodology is a business process model, which can narrow the application of the methodology.

And business process management strategies would be powerful tools to guide the design of process-centered ChatBot implementation method.

2.4. CONCLUSIONS

IN this chapter, literature reviews were focused on concepts related to business process management (BPM), ChatBot technology, and ChatBot implementation methodology. In this research, BPM and ChatBot are the two most important keywords. The definition of BPM was firstly introduced. The lifecycle of BPM serves as one of the guidelines for ChatBot implementation methodology designed in the next Methodology chapter. Business process modeling was also discussed, which is the key process representation approach in the process design. Concepts such as BPA and RPA are further introduced. These techniques emerge as a trend for future business process redesign with intelligent robots such as ChatBots. The ChatBot technology was then discussed, including its history, architecture, classification, and main vendors. This knowledge serves for the ChatBot selection in the process design. Some methodologies to building a ChatBot system were then introduced. The proposed ChatBot-implementation methodology in this report is derived from those methodologies yet based on the BPM lifecycle.

2

REFERENCES

[1] M. Weske, *Business process management–concepts, languages, architectures, verlag,* (2007).

[2] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management* (2013).

[3] M. Von Rosing, H. Von Scheel, and A.-W. Scheer, *The complete business process handbook: body of knowledge from process modeling to BPM*, Vol. 1 (Morgan Kaufmann, 2014).

[4] *ABPMP Standards for Business Process Management (BPM),* .

[5] *Business process modeling,* .

[6] S. White, *Business process modeling notation v1. 0." for the business process management initiative (bpmi),* (2004).

[7] O. M. Group, *Unified Modeling Language (UML) Version 2.4.1,* (2011).

[8] T. A. Curran and G. Keller, *SAP R/3 business blueprint: Business engineering mit den R/3-Referenzprozessen* (Addison-Wesley, 1999).

[9] R. Davis and E. Brabander, *ARIS design platform: getting started with BPM* (Springer Science & Business Media, 2007).

[10] O. M. Group, *Business Process Model and Notation (BPMN), Version 2.0,* (2011).

[11] Gartner, *Business process automation (bpa),* .

[12] *Business process automation,* .

[13] C. Kroll, A. Bujak, V. Darius, W. Enders, and M. Esser, *Robotic process automation-robots conquer business processes in back offices,* Capgemini Consulting. Hentet fra https://www. capgemini. com/consulting-de/wpcontent/uploads/sites/32/2017/08/robotic-process-automation-study. pdf (2016).

[14] C. Boulton, *What is RPA? A revolution in business process automation*, Tech. Rep. (2018).

[15] H. P. Fung, *Criteria, use cases and effects of information technology process automation (itpa),* Advances in Robotics & Automation **3** (2014).

[16] Y. F. Wang and S. Petrina, *Using learning analytics to understand the design of an intelligent language tutor–chatbot lucy,* Editorial Preface **4** (2013).

[17] J. Beck, M. Stern, and E. Haugsjaa, *Applications of ai in education,* Crossroads **3**, 11 (1996).

[18] T. Lee, S. Wilde, and Y. Chen, *Artificially Intelligently (AI) Tutors in the Classroom :,* (2019).

[19] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, *Superagent: A customer service chatbot for e-commerce websites,* in *Proceedings of ACL 2017, System Demonstrations* (2017) pp. 97–102.

[20] H.-Y. Shum, X.-d. He, and D. Li, *From eliza to xiaoice: challenges and opportunities with social chatbots,* Frontiers of Information Technology & Electronic Engineering **19**, 10 (2018).

[21] *ELIZA,* .

[22] *PARRY,* .

[23] *A.L.I.C.E.* .

[24] H. Henderson, *Artificial intelligence: mirrors for the mind* (Infobase Publishing, 2007).

[25] *Siri,* .

[26] *IBM Watson,* .

[27] *Amazon Alexa,* ().

[28] P. Leslie Willcocks and A. Craig, *The Outsourcing Unit Working Research Paper Series Paper 15/02 Robotic Process Automation at Telefónica O2 Research on Business Services Automation Research Objective*, Tech. Rep. (2015).

[29] A. Cangiano, *Chatbots are Trending,* (2018).

[30] A. Cangiano, *Introduction to Chatbots,* (2018).

[31] A. Cangiano, *Understanding Intents,* (2018).

[32] A. Rahman, A. Al Mamun, and A. Islam, *Programming challenges of chatbot: Current and future prospective,* in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)* (IEEE, 2017) pp. 75–78.

[33] A. Cangiano, *Understanding Entities,* (2018).

[34] A. Cangiano, *Putting it All Together,* (2018).

[35] S. Barker *et al.*, *How chatbots help,* MHD Supply Chain Solutions **47**, 30 (2017).

[36] M. Nuruzzaman and O. K. Hussain, *A survey on chatbot implementation in customer service industry through deep neural networks,* in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)* (IEEE, 2018) pp. 54–61.

[37] M. Mason, *3 Types of Business Chatbots You can Build*, Tech. Rep. (2017).

[38] *Watson Assistant,* .

[39] *Amazon Lex,* ().

[40] *Google Dialogflow,* .

[41] *Microsoft Azure,* ().

[42] *Microsoft Azure Bot,* ().

[43] D. Zumstein and S. Hundertmark, *Chatbots–an interactive technology for personalized communication, transactions and services.* IADIS International Journal on WWW/Internet **15** (2017).

[44] L. Ardissono, D. Cardinio, G. Petrone, and M. Segnan, *A framework for the server-side management of conversations with web services,* in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (ACM, 2004) pp. 124–133.

[45] J. Bentahar, B. Moulin, and B. Chaib-draa, *Towards a formal framework for conversational agents,* in *Proceedings of Agent Communication Languages and Conversation Policies AAMAS 2003 Workshop* (Citeseer, 2003).

[46] R. S. Cost, Y. Chen, T. Finin, Y. Labrou, and Y. Peng, *Using colored petri nets for conversation modeling,* in *Issues in agent communication* (Springer, 2000) pp. 178–192.

[47] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, *Tropos: An agent-oriented software development methodology,* Autonomous Agents and Multi-Agent Systems **8**, 203 (2004).

[48] M. Wooldridge, N. R. Jennings, and D. Kinny, *The gaia methodology for agent-oriented analysis and design,* Autonomous Agents and multi-agent systems **3**, 285 (2000).

[49] P. R. Telang, A. K. Kalia, and M. P. Singh, *Engineering service engagements via commitments,* IEEE Internet Computing **18**, 46 (2013).

[50] A. K. Kalia, P. R. Telang, J. Xiao, and M. Vukovic, *Quark: a methodology to transform people-driven processes to chatbot services,* in *International Conference on Service-Oriented Computing* (Springer, 2017) pp. 53–61.

3

Methodology Design

Success is most often achieved by those who don't know that failure is inevitable.

Coco Chanel

This chapter introduces the requirements, design strategy, description approach, and validation and evaluation of designing a ChatBot implementation methodology. Section 3.1 shows the requirements that the designed methodology needs to fulfill according to the limitation of the current methodology in Section 2.3. The methodology design strategy is described in Section 3.2. Section 3.3 introduces an approach to describe a methodology: ways of thinking, working, and modeling in the analytical framework. How to validate and evaluate the designed methodology is described in Section 3.4 at last.

3.1. REQUIREMENTS

AMONG the methodologies discussed in Section 2.3, Quark methodology is a well-established one to implement ChatBot into business processes by using the existing business process model. However, there are some limitations to the methodology. The main goal of the methodology is to provide an approach to apply ChatBot into a process to replace human labor. However, it pays less attention to business process management. The ChatBot implementation is not carried out in the level of the business process. One should keep it in mind that the goal of ChatBot implementation is to enhance process performance and efficiency. The input of the methodology is a business process model, which can narrow the application of the methodology. According to the limitation of Quark methodology, the designed methodology needs to fulfill the following requirements.

- The main goal of the methodology is to increase the efficiency of a business process which involves high-volume human interactions.
- Cost-effectiveness is one of the cores of ChatBot implementation methodology.
- The methodology is process-based to ensure practical executability.
- The necessity of ChatBot implementation should be investigated before the process automation.
- The input of the methodology can be a process rather than a model (for example, a BPMN model), which could extend the application situation of the methodology.
- Discovering issues and problems of the current situation are the foundation of process (re-)design
- The methodology should be able to describe the interactions among ChatBot and human agents within the process in a formal way.
- Effectiveness of methodology should be measured.

3.2. DESIGN STRATEGY

HEREIN, a design strategy is proposed to fulfill the above requirements. The methodology can be developed based on the BPM lifecycle, which ensures the business processes covered by the BPM lead to consistently positive outcomes and deliver maximum value to the organization [1], together with the Quark ChatBot implementation methodology [2] (see Section 2.3).

The BPM theory ensures a process-based methodology, which takes a process as its input. As discussed in Section 2.1.2, the BPM lifecycle is composed of process identification, discovery, analysis, redesign, implementation, and monitoring and controlling. Process identification studies the possibility of the methodology applying to the process. Process discovery and process analysis investigate the necessity of ChatBot implementation into the process and uncover the issues as well as problems of the current process. Process redesign offers a feasible route of the process improvement. Process implementation provides the guideline for the practical execution of the redesigned process. Process monitoring and controlling guarantees continuous improvement of the process.

During the process redesign stage, ChatBots are embedded in the process. The approach of applying ChatBots can be initiated from the Quark methodology [2], which establishes a basic outline of ChatBot implementation. Additionally, the methodology in this research needs to establish approaches to determine which tasks can and should be handled by ChatBots and to determine what kind of ChatBots should be applied for specific tasks. Meanwhile, methods to illustrate interactions within the process also need to be provided.

3.3. DESCRIPTION APPROACH

IT is crucial to depict the methodology with a clear logic. The analytical framework of Sol[3] can be regarded as suitable approach to describe the ChatBot implementation methodology. The framework consists of three parts namely **way of thinking**, **working**, and **modeling**, presented in Figure 3.1.

The way of thinking concerns the organization of ideas and the underlying structure of the methodology, forming the outline of the methodology. It provides fundamental principles and philosophies that guide the methodology design and implementation.

The way of working focuses on the steps of a methodology that need to be executed to improve the performance of the current situation. These steps form a routine from the current situation towards the new situation. The way of working provides an executable guideline of the methodology implementation.

The way of modeling states the approach of depicting the real situation with an abstract model that is suitable for the systematical analysis. A number of modeling techniques that can be employed in the methodology have been introduced in Section 2.1.3. In the different stages of the methodology, various modeling approaches can be applied. The selection of suitable modeling approach is based on specific requirements of each stage.

Figure 3.1 illustrates correlations among the way of thinking, working, and modeling. Way of working and modeling are the piratical activities of the methodology implementation. During the methodology execution, way of working needs the support of way of modeling, and way of modeling can be changed in the different stages of the way of working. Meanwhile, the way of thinking provides the guideline for the way of working and modeling. Way of working and modeling are executed with the philosophies of the way of thinking.

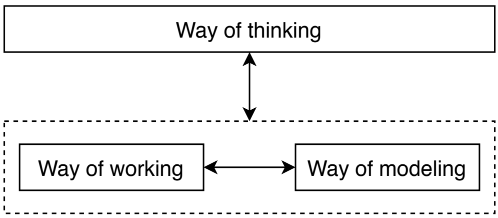


Figure 3.1: Analytical Framework for ChatBots implementation methodology

3.4. VALIDATION AND EVALUATION

TO check the validity and evaluate the methodology, it is necessary to use a case study to execute the methodology. By employing the method of case study, the methodology can be practically performed to automate a process by implement ChatBots. Accordingly, the validity of methodology can be examined, and the limitation of the methodology can also be explored, laying the foundation for the further improvement of the methodology.

The selection of a suitable case is of great importance. Herein, the methodology aims to implement ChatBots to replace human labor in the business process. Comparing to traditional software programs, ChatBots possess the ability of handling complex tasks involved human-machine interactions. The ChatBot-implementation methodology initially is designed for the process improvement of service-oriented processes. Normally, service-oriented processes which aim to provide service to customers involve a huge amount of interactions with humans (customers) and consume a high volume of human labor to support those services. The implementation of ChatBots in a service-oriented process triggers a possibility of process improvement including cost reduction and efficiency enhancement. Hence, a service-oriented process can be regarded as a good choice for the validation and evaluation of the ChatBot-implementation methodology. However, other processes that engage complex human-machine interactions and need to be automated can also be regarded as a potential candidate for the case study.

During the case study, it is essential to build a suitable indicator to evaluate the effects of the methodology. Evaluation indicators be should be considered based on the specific type of process. The different processes can hold various core values and requirements of the process automation, which should be reflected by the built evaluation indicators.

Employment of validation and evaluation of the methodology benefits both the process improvement and development of the methodology itself. It provides the measures of the methodology's effectiveness and chances to figure out the limitations and drawbacks of the methodology.

REFERENCES

[1] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management* (2013).

[2] A. K. Kalia, P. R. Telang, J. Xiao, and M. Vukovic, *Quark: a methodology to transform people-driven processes to chatbot services,* in *International Conference on Service-Oriented Computing* (Springer, 2017) pp. 53–61.

[3] H. G. Sol, *Information systems development: a problem-solving approach,* in *Challenges and strategies for research in systems development* (John Wiley & Sons, Inc., 1992) pp. 151–161.

4

ChatBot-implementation Methodology

If you do what you have always done, you'll get what you have always gotten.

Tony Robbins

ChatBot Implementation Methodology part designs the ChatBot-implementation methodology according to the design strategy (Section 3.2) and description approach (Section 3.3). The way of thinking introduces in Section 4.1. It guides in representing a system in the form of a model. Section 4.2 concerns the steps that need to be taken to cope with the specific characteristics in ChatBot-implementation. The ChatBot-implementation cycle consists of seven interactive activities divided into an understanding phase, a design phase, and an implementation phase. And Section 4.3 regards the choice of modeling techniques, and the construction of models during a study of Curtis (1992). The purpose of modeling is to reduce the complexity of a problem situation, to increase the understanding of the dynamic interactions between objects, and to assess the impact of changes.

4.1. WAY OF THINKING

THE way of thinking guides the process automation methodology, it represents the underlying structure and ideas that can form the outline[1].

The way of thinking begins with a clear vision of the target processes in which this process automation strategy can be applied. The methodology aims to automate the existing service-oriented business process, which is partly- or non-automated, contain intensive human-human and/or human-machine interactions, and involve multi-system corporation. The goal of this methodology is to decrease the process complexity, to enhance the interaction experience of customers, and to increase the efficiency of service agents within the process.

A comprehensive and structured understanding of the current situation is the basis of this methodology. Such understanding can be obtained by either qualitative or quantitative analysis. The goal and major functions of the process should be carefully identified. Limitations and drawbacks of the current situation need to be figured out as the basis for the process improvement.

ChatBot implementation is of great importance in this service process automation methodology. ChatBots are employed to accomplished some tasks of the process to increase the process efficiency and performance. And cost-effectiveness is the core consideration during the ChatBot implementation. It means that suitable ChatBots should be placed in a suitable position to make the best use of system resource and achieve all the required functions.

However, it is not necessary nor wise to replace all human labor of the process by ChatBots. The human involvement is an important facet of the business process which is an integration of both service and human. As a result, it is hard to predict all of the scenarios of such a dynamic process. Hence, a fully-automated process is not necessary or cost-effective. In this methodology, exceptional human handling is also an important concern to make sure the process works smoothly. This methodology provides a framework of collaborative human-ChatBots integrated process automation.

4.2. WAY OF WORKING

THE way of working with regard to the steps need to be taken with the specific characteristics in ChatBots implementation. It on the basis of the problem-solving process and BPM lifecycle[2, 3], as present in Section 2.1.2. Figure 4.1 is the ChatBots implementation cycle consists of seven interactive activities divided into an understanding phase, a design phase, and an implementation phase.

The understanding phase consists of three activities. The first one is the process identification. The second one is process discovery, the as-is model (model of the current process) is constructed at the end of this activity. The third one is the process analysis, investigating the issues and facts of the current process. The design phase contains two activities: ChatBot-implementation process redesign and evaluation. The to-be model (model for the redesigned process) is constructed during the ChatBot implemented process redesign activity. And the to-be model needs to be evaluated together with the as-is model. Process implementation, and process monitoring and controlling are the last two activities in the implementation phase. The new situation is formed after process imple-

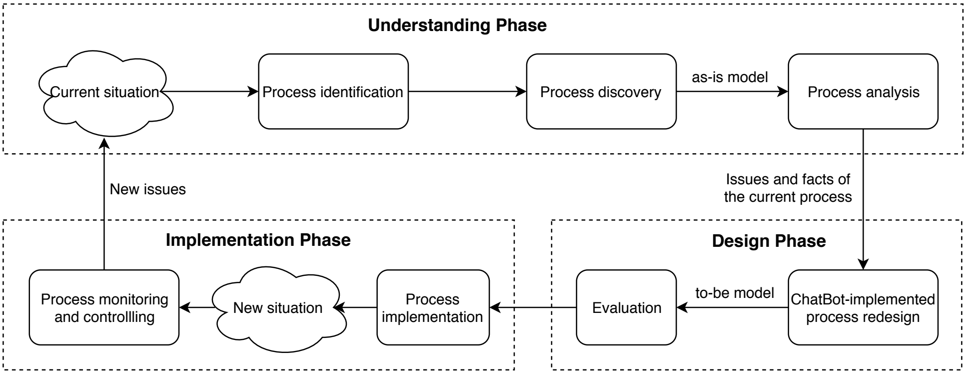


Figure 4.1: Way of Working

4

mentation. And process monitoring and controlling could discover new issues and the ChatBots implementation cycle will run again then.

4.2.1. Process identification

In this methodology, the process identification aims to systematically define the current process, including the process workflow, stakeholders, objectives (or functions) and activities. It serves as a foundation for further process modeling and redesigning.

4.2.2. Process discovery (as-is process modeling)

In this step, the current situation is documented and illustrated in the form of the as-is process model. Among various method of process modeling introduced in the Literature review, BPMN is chosen for process modeling in this research, which will be further explained in the way of modeling.

4.2.3. Process analysis

In the phase of process analysis, issues and drawbacks related to the as-is process are listed and investigated. The analysis can be carried out either by qualitative or quantitative approaches, such as data analysis on the as-process and interview with stakeholders. The collection of those issues and drawbacks works as a guide for further process improvement steps.

4.2.4. ChatBot-implemented process redesign

Function evaluation

Prior to the integration of ChatBots, the basic functions of the process need to be first evaluated to determine which function can be and need to be accomplished by ChatBots. The evaluation of functions can be based on the results of the process analysis.

ChatBot selection

Based on the evaluation results, suitable ChatBots for each function can be chosen.

Herein, ChatBots are categorized into three types namely *Rules-based ChatBots*, *Knowledge-based ChatBots*, and *AI-based ChatBots*, based on the tasks that ChatBots can handle with.

- **Rules-based ChatBots**: As suggested by the name, *Rules-based ChatBots* are powered by a series of defined rules. These rules are the basis for the ChatBots to handle those works which are well-structured, highly-patterned, and less complex.*Rules-based ChatBots* only perform the trained scenarios without learning from the interactions, which makes them generally easy and fast to train and thus less expensive. *Rules-based ChatBots* can be easily applied to handling high-volume and repetitive jobs.

- **Knowledge-based ChatBots**: They are trained by an amount of data or by a knowledge base. They need to perform machine learning during training. Thus *Knowledge-based ChatBots* can deal with works which are more complex, for example, QA service. However, they can only handle works with patterned input due to a lower learning level.

- **AI-based ChatBots**: *AI-based ChatBots*, in comparison, are built to understand the context and intents of questions before formulating a response, which is supported by machine learning. They are employed in much more complicated cases. *AI-based ChatBots* are always equipped with Natural Language Processing (NLP) to understand and generate natural-language requests or responses. *AI-based ChatBots* can learn from information gathered and continuously improve themselves as more data comes in. Hence they pose a broader range of decision-making skills and understand many languages.

To make the best use of the ChatBots, the selection of ChatBots from the above-mentioned types are carried out based on the process complexity and the input type, as indicated in Figure 4.2. For structured and patterned input with less complexity work, *Rules-based ChatBots* can be a good choice. When it goes to more complex work with unstructured input, *Knowledge-based ChatBots* would deal with them. *AI-based ChatBots* generally are used to handle the highly complicated job with unstructured input in free forms, such as interaction with a human.

EXCEPTION HUMAN OPERATIONS

It needs to be emphasized that not all the activities or functions can be or need to be performed by ChatBots. Hence, the exceptional human operations should be identified to ensure the smooth running of the process. Human needs to handle issues which are either unexpected or technically difficult to solve by ChatBots.

CONCEPTUAL FRAMEWORK DESIGN

In this step, links and interactions among all process-engaged ChatBots and human are built in the form of a conceptual framework. Herein, the notion of commitments[4, 5] is used to describe these interactions. The notion of commitments enables the flexibility in constructing communications constructing and ordering while correctly picturing agent interactions. A commitment is a four-place relation with the form of

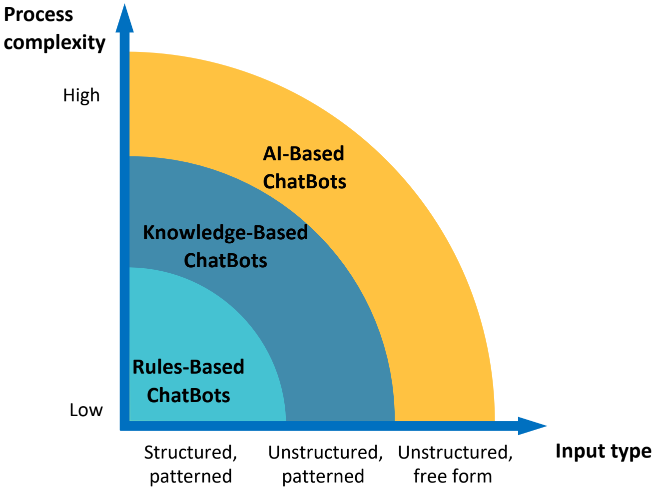


Figure 4.2: ChatBots selection criteria

C (*debtor, creditor, antecedent, consequent*),

where the *debtor* is the agent who is committed, and the *creditor* is the agent who receives the commitment. In the case that the *antecedent* condition provided by the creditor holds true, the debtor promise to hand out a *consequent* condition. For example, the situation of a phone user asks a ChatBot agent (like Siri[6]) to build a calendar item in his phone can be described by the notion of commitments as C (Siri, user, request, calendar item). This conceptual framework represents architecture and workflow the redesigned process and serves as the foundation for further process modeling.

To-be process modeling

Similar to Section 4.2.2, the to-be process model is established here. Those redesign and improvement are represented by the as-model. The as-is model is the outcome of the *Process redesign* phase and serves as a fundament for the next phase.

4.2.5. Evaluation

Here, evaluations are carried out based on the process model to determine how successfully the to-be process achieved the goals and objectives. A comparison between the as-is and to-be process can indicate the effects and benefits of the process redesign. The outcome of evaluation can help the following implementation phase to figure out potential implementation failures and theory failures[7].

4.2.6. Process Implementation

Process implementation prepares the changes between the as-is process model and the to-be process model and performs the changes. This phase includes two parts:

- **organizational change management:** the set of activities that need to change the way of all involved participants working in the process,

- **practical process automation**, the development and deployment of IT system based on the to-be process model.

The practical process automation may be carried out with the assistance or based on service from ChatBot vendors, listed in Section 2.2.4 An executable process model will be gained from to-be process model and will be deployed in a BPMS (Business Process Management Software).

4.2.7. PROCESS MONITORING AND CONTROLLING

After being running, the redesigned process based on the to-be model can be further analyzed by relevant data collected from the process operations. It can help to determine the process performance. Bottlenecks, recurrent errors or deviations concerning the intended behavior are identified and corrective actions are undertaken. This phase may arise new issues, and the lifecycle should be repeated to solve the new issues. Efforts can be employed to solve issues or further improve the process.

4.3. WAY OF MODELING

THE way of modeling aims to reduce the complexity of the current process situation and to access the impacts of the changes between the as-is process model and the to-be process model[2]. The way of modeling with regard to the selection of modeling techniques, and the construction of models through a thorough study[8]. Various business process modeling techniques have been introduced in Section **??**. One can choose different modeling techniques according to the different cases or stages of processes.

In this thesis, the Business Process Model and Notation (BPMN) was applied to specify the as-is process and the to-be process. BPMN is a standard for business process modeling, based on traditional flow-charting techniques[9]. It can provide notations to business users and then represent complex process semantics[10]. BPMN serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation[11]. The objective of BPMN is to provide a readily understandable notation and support business process modeling for both technical users and business users[10]. BPMN version 1.0 was officially accepted as an OMG standard in 2006. After some smaller changes in versions 1.1 and 1.2, version 2.0 brought more comprehensive changes and extensions. It was published in 2011. In 2013, BPMN also became an official ISO standard [ISO 2013].

The Business Process Diagram is the graphical representation of the BPMN. Its language constructs are grouped in four basic categories of elements, i.e., **Flow Objects**, **Connecting Objects**, **Swimlanes**, and **Artifacts** (see Figure 4.3). The notation is further divided into a core element set and an extended element set. The intention of the core element set is to support the requirements of simple notations and most business processes should be modeled adequately with the core set. The extended set provides additional graphical notations for the modeling of more complex processes[12]. Here is some brief introduction on the Flow Objects, Connecting Objects, Swimlanes and Artifacts in BPMN diagrams.

Four basic elements constitute BPMN: **Flow Objects**, **Connecting Objects**, **Swimlanes**, and **Artifacts** (see Figure 4.3).

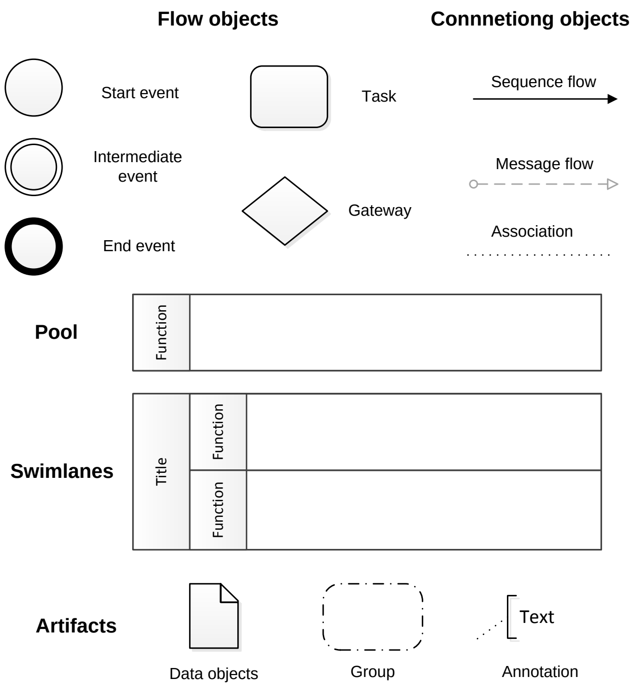


Figure 4.3: Basic elements in BPMN diagram including *Flow Objects*, *Connecting Objects*, *Swimlanes*, and *Artifacts*

REFERENCES

[1] H. G. Sol, *Simulation in information systems development.*, Ph.D. thesis, University of Groningen (1982).

[2] M. Janssen, *Designing electronic intermediaries: An agent-based approach for designing interorganizational coordination mechanisms.* (2002).

[3] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management* (2013).

[4] M. P. Singh, *An ontology for commitments in multiagent systems,* Artificial intelligence and law **7**, 97 (1999).

[5] P. R. Telang, A. K. Kalia, M. Vukovic, R. Pandita, and M. P. Singh, *A conceptual framework for engineering chatbots,* IEEE Internet Computing **22**, 54 (2018).

[6] *Siri,* .

[7] C. H. Weiss, *Theory-based evaluation: Past, present, and future,* New directions for evaluation **1997**, 41 (1997).

[8] B. Curtis, M. Kellner, and J. Over, *Process modeling. communications of acm,* vol **35**, 75 (1992).

[9] S. Simpson, *An xml representation for crew procedures, final report nasa faculty fellowship program (johnson space center),* (2004).

[10] S. White, *Business process modeling notation v1. 0." for the business process management initiative (bpmi),* (2004).

[11] *Business process model and notation—bpmn,* in *The Complete Business Process Handbook*, edited by M. von Rosing, A.-W. Scheer, and H. von Scheel (Morgan Kaufmann, Boston, 2015) pp. 433 – 457.

[12] G. Aagesen and J. Krogstie, *Analysis and design of business processes using bpmn,* in *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, edited by J. v. Brocke and M. Rosemann (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 213–235.

5

Methodology Evaluation

Only those who will risk going too far can possibly find out how far one can go.

T. S. Eliot

This chapter is the evaluation of the designed ChatBot-implementation methodology in Chapter 4. *Call Management* process in Philips IT Global Services is chosen as an example case to apply the ChatBot-implementation methodology in order to obtain an automated process. The *Call Management* process is a service-oriented process providing IT support to employees of the company, which is suitable for applying the proposed methodology to achieve ChatBot-based process automation. The evaluation starts with a case introduction on Philips and its goal of process improvement including process automation (Section 5.1). The *Call Management* process then is automated by following the routine described by the way of working (Section 4.2). Section 5.2 is the Process identification, in which the goal, architecture, major functions are introduced. The as-is process model then is illustrated in the Process discovery (Section 5.3). After that, the process is analyzed to obtain facts and issues of the current process (Section 5.4). Then, redesign of the process is conducted in Section 5.5, containing functions evaluation, ChatBots selection, defining exception human operations, designing the conceptual framework, and to-be process modeling. The to-be process model, as well as the as-is model, are evaluated in Section 5.6. At last, conclusions together with some recommendations on the process implementation, process monitoring and controlling are provided in Section 5.7.

5.1. BACKGROUND

PHILIPS is a leading health technology company focused on improving people's health and enabling better outcomes across the health continuum – thereby creating value for its stakeholders. With the focus on delivering meaningful innovation, Philips serve both professional and consumer markets throughout the world in the areas of health systems and personal health. Philips is a leader in diagnostic imaging, image-guided therapy, patient monitoring, and health informatics, as well as in consumer health and home care[1].

The mission of Philips is to improve people's lives through meaningful innovation. About the vision, Philips strives to make the world healthier and more sustainable through innovation. And the goal is to improve the lives of 3 billion people by 2025[2]. The Philips Business System (PBS) is designed to help deliver on mission and vision with its four interlocking elements: *Strategy*, *CAPs*, *Excellence*, and *Path to Value*. See in Figure 5.1.

Figure 5.1: Philips Business System[3]

Philips Excellence is a part of PBS including five operating principles: *Processes and systems*, *People and culture*, *Capabilities and learning*, *Performance and reward*, and *Structure and governance*.

Standardization if a key aspect of Philips Excellence. Standardization of *Processes and systems* can make it more effective and efficient with quality and compliance improvement. The basis for standardization of processes and systems is *Philips Excellence Process Framework* (PEPF)[3]. PEPF describes all business processes for Philips. The PEPF flows seamlessly, from Idea to Market (I2M), Market to Order (M2O), Order to Cash (O2C), supported by a set of management and enabling processes. See in Figure 5.2. Information Technology (IT) is one part of enabling processes, and processes within IT need to be improved to achieve PEPF.

This research is part of the *Service Desk and Process Automation* team in Philips IT Global Services (ITGS). The service desk is an essential Information Technology Infras-

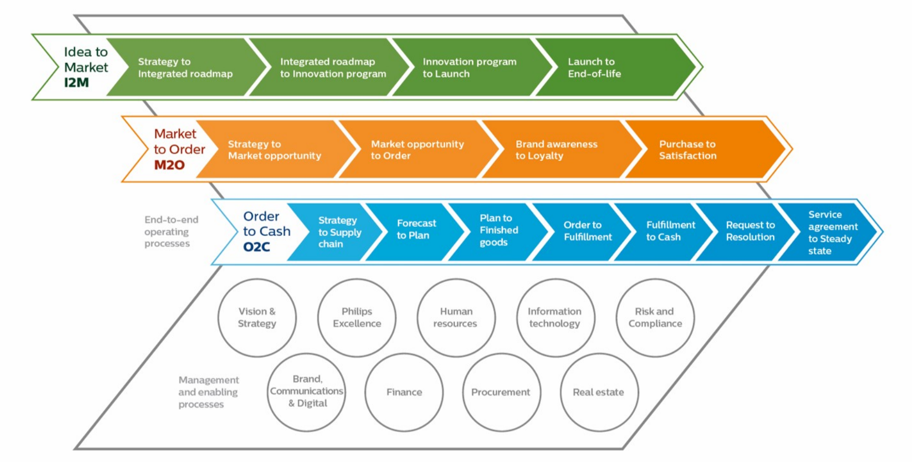


Figure 5.2: Philips Excellence Process Framework describes processes[3]

5

tructure Library (ITSM)[1] tool defined by the Information Technology Infrastructure Library (ITIL)[2]. A service desk is a communication center that provides a single point of contact between a company and its customers, employees, and business partners. The purpose of a service desk is to ensure that users receive appropriate help in a timely manner. Service desks are designed to handle both incidents (events that result in a disruption in service availability or quality) and service requests (such as helping a user change a password or getting a new user set up in work systems)[6].

Service Desk and Process Automation team in ITGS is a newly organized team in this year, including *Service Desk* part, *Process Excellence* part, and *Automation* part to realize end-user satisfaction, process standardization, process automation, cost reduction, service performance, and business continuity (Figure 5.3).

Figure 5.3: Service Desk and Process Automation Result[7]

All the processes in service operations need to be improved to meet PEPF standards, including: *Incident Management*, *Request Fulfillment*, *Problem Management*, *Data Retention and Archival*, *Access Management*, *Operations Management*, and *Event Manage-*

[1] ITSM is a general term that describes a strategic approach to design, deliver, manage and improve the way businesses use IT. ITSM includes all the discrete activities and processes that support a service throughout its lifecycle, from service management to change management, problem and incident management, asset management, and knowledge management[4].

[2] The ITIL is a framework designed to standardize the selection, planning, delivery and maintenance of IT services within a business[5].

ment. How to apply process improvement methodology and automation is the current problem needs to be analyzed.

5.2. PROCESS IDENTIFICATION

THE evaluation case focuses on a *Call Management* process in Philips IT service operations. The *Call Management* process is a typical IT service business process, which engages a huge amount of human labor to provide global IT support within the Philips.

Firstly, the goal and procedure of the *Call Management* process are identified. Herein, the information of the process are collected from the internal documents, interview with process managers and *Service Desk Agents*. The *Call Management* process aims to handle all *Calls* from business users (*Callers*) promptly and to answer or solve as many *Calls* as possible using knowledge in the system without involving other levels of support[8]. Figure 5.4 shows a standard *Call Management* process, including *Call Logging*, *Call Solving*, and *Call Closure*.

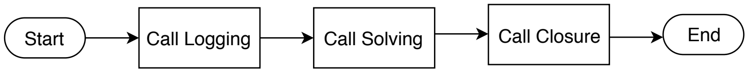


Figure 5.4: *Call Management* Process

Caller and *Service Desk Agent* are two main roles[8] related to *Call Management* process. Any user who contacts Service Desk is called *Caller*. And *Service Desk Agent* is the frontline of the IT department. He/She will be the first point of contact for all users. *Service Desk Agent* has the knowledge of the IT environment and the service offerings provided.

In *Call Logging* part, *Service Desk Agent* first gathers information through *Phone*, Chat, *Email*, or *Web portal*. The information includes the name and other basic information of the *Caller*. The *Service Desk Agent* then registers a new *Call* by filling the required fields in a *Call* ticket. If this *Call* have relation to an existing *Call* ticket, the *Service Desk Agent* need to reopen the existing *Call* ticket (if closed) and update its status. In the next step, the *Service Desk Agent* classifies the *Call* ticket by categorizing it appropriately as a *Complaint*, *RFC*, or *Incident*.

During *Call Solving* phase, the *Service Desk Agent* needs to review the request from the *caller*. In case that the solution to the caller's request is available in the knowledge base or can be solved directly, the *Service Desk Agent* would provide the solution to the *Caller* and fill in the *Call* ticket. Otherwise, the *Service Desk Agent* needs to create a new *Incident* to send the *Call* to the next level support.

Once the *Call* is solved, either by the *Service Desk Agent* or the next level support, the *Service Desk Agent* would notify the *Caller* by e-mail or web portal.

In the whole process, *Service Desk Agent* is responsible for receive problem from *Caller*, extract *Caller* information, extract problem information, organize problem information, fill in *Call* ticket, manage ticket status, find solution from database, ask for solution from the next level support, and provide solution to *Caller*. All the functions can be summarized as *Caller communication*, *Call information management*, *Ticket management*, and *Solution management*:

- F1 Caller communication
  - F1.1 Receive problems
    - *Choose contact type*
    - *Contact Service Desk*
    - *Receive the problem*
  - F1.2 Provide solutions
    - *Provide Call number and the result to Caller*
    - *Provide the final report to Caller*
    - *Receive the problem final report*
- F2 Call information management
  - F2.1 Extract callers information
    - *Select Caller name*
    - *Obtain sufficient information to fill in Call ticket*
  - F2.2 Extract problems information
    - *Check if the Call is a new Call*
    - *Determine the category of the Call ticket (Incident, RFC, Complaint*
  - F2.3 Organize problems information
- F3 Ticket management
  - F3.1 Tickets status operation (new, reopen, or close)
    - *Check if the existing open Call closed or not*
    - *Update status of existing open Call*
    - *Reopen the closed Call and update information*
  - F3.2 Fill in new tickets
    - *Check if Call Model available*
    - *Select Call Model*
    - *Fill in Call ticket*
    - *Create Incident from Call*
    - *Create Change from Call*
    - *Create Complaint from Call*
- F4 Solution management
  - F4.1 Find solutions from database
    - *Check if the Call related to Open Incident*
    - *Check if the Call match to Open Known Error*
    - *Check if the solution could be found in Knowledge Base*
    - *Relate Call in Knowledge Record*
    - *Check if the Service Desk Agent able to solve*
    - *Document solution*
  - F4.2 Save new solutions to database

5.3. PROCESS DISCOVERY (AS-IS PROCESS MODELING)

THE *Call Management* process is triggered by problem occurring from *Caller*, and completes with solution provided to *Caller*. Figure 5.5 shows the boundaries and main activities of *Call Management* process.

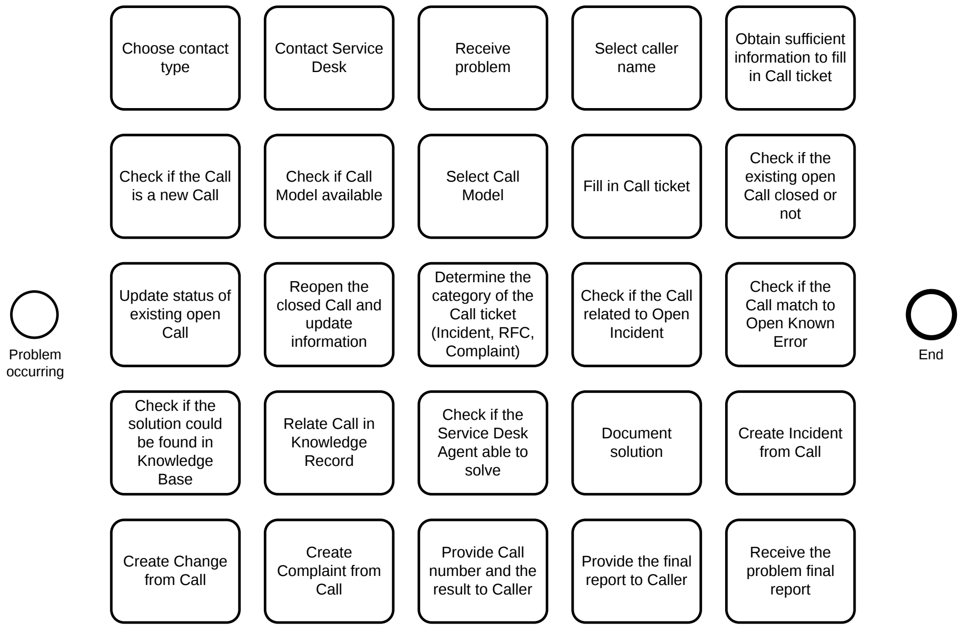


Figure 5.5: The Main Activities of *Call Management* Process

Then these activities are distributed to responsible roles. the Figure 5.6 shows the set of activities of the *Call Management* process being assigned to pools and lanes. The sequence flows indicate handover points.

The final *Call Management* as-is process model presents in Figure 5.7 and Figure 5.8. The model is separated into *Call Logging* and *Call Solving and Call Closure* due to the complexity. Following are the details of the as-is process model.

CALL LOGGING

Figure 5.7 is the BPMN of *Call Logging* part in current *Call Management*. *Call Logging* registers calls (incidents, questions, request for change (RFC), or complaints[3]) from business users[8]. There are two kinds of roles in *Call Logging* process:

- **Service Desk Agent:** The Service Desk Agent is the frontline of the IT department. He/She will be the first point of contact for all users. They will have the knowledge of the IT environment and the service offerings provided.
- **Caller:** Any user who contacts Service Desk.

[3]Complaints are contacts from a business user to the service desk expressing dissatisfaction with the service offerings or level of support received from the IT department.

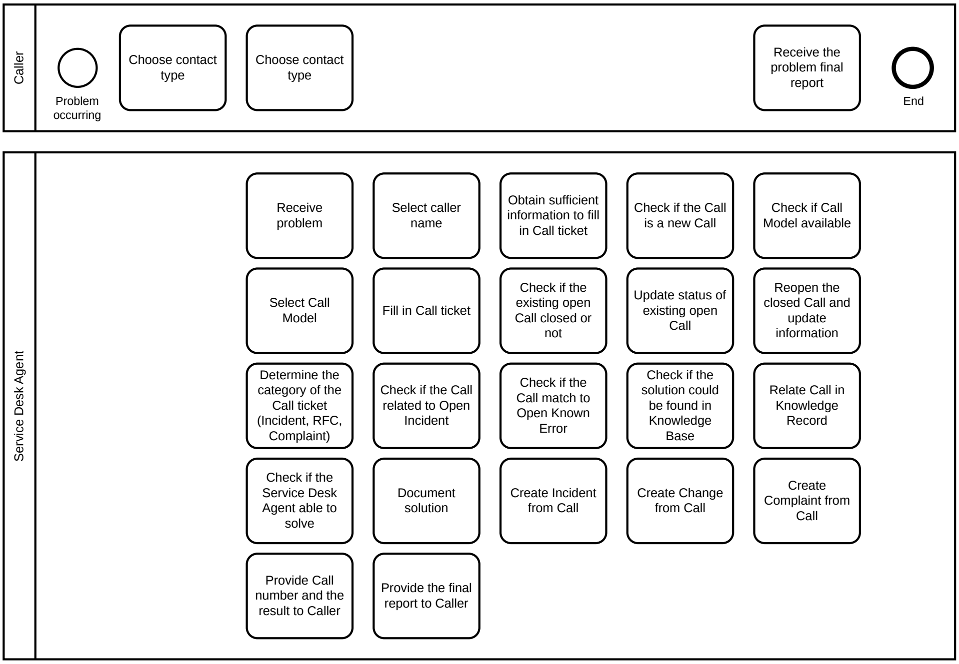


Figure 5.6: The Main Activities of *Call Management* Process Assigned to Pools and Lanes

There are four different methods for callers to contact service desk: *Phone*, *Chat*, *Email*, and *Web portal*. When receiving a new *Call*, service desk agent collects the caller's name and other basic information first if the caller contacts them with *Phone* and *Chat*. On the other hand, basic information is offered to service desk agents if callers contact them with *Email* and *Web portal*. But agents need to confirm if the provided information is sufficient, otherwise, it is still needed for agents to contact callers to obtain the missing information.

After confirming the basic information of the caller, the second step for service desk agent is to decide whether this *Call* is related to other existing *Calls*. If the *Call* is a new *Call* that does not have relation to other *Calls*, agent start to register it. Some available *Call* templates can be selected to speed up the registration. While the *Call* does have relation to existing *Calls*, agent could update status of the existing *Call* or reopen the closed existing *Call*.

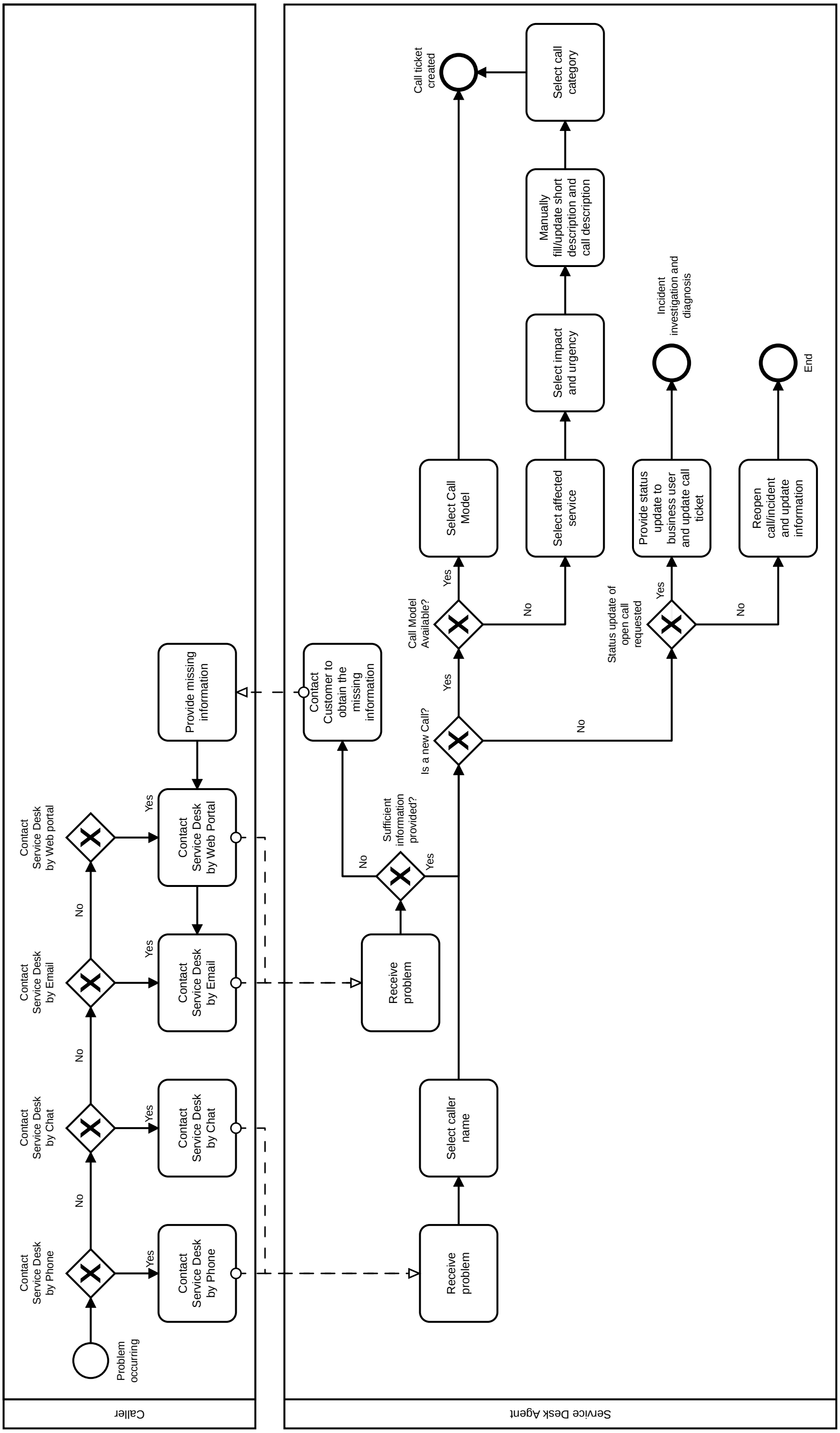


Figure 5.7: BPMN of Call Logging in Call Management Process

CALL SOLVING AND CALL CLOSURE

Figure 5.8 is the BPMN of *Call Solving* and *Call Closure* part in current *Call Management* process. The main purpose of *Call Solving* is **a)** to review *Complaint*, investigate the cause, and determine the actions required to prevent the complaint from recurring, **b)** to log in the follow-up actions as *Problem* with category as 'CSI Action', **c)** to execute actions and verify acceptance from the Caller. And *Call Closure* aims to close the *Call* and provide the final report to *Caller* after problem solving.

The main roles in this part are *Service Desk Agent* and *Caller*, they are the same with the *Call Logging* part.

Call Solving starts with the creation of *Call* ticket. After receiving the *Call* ticket, a service desk agent should first determine the category of the problem (incident, request for change, or complaint). Take incident category for example to solve, if the *Call* ticket matches to an open incident, the ticket should be related to the open incident, and the service desk agent provides the number and current state to *Caller*. Otherwise the *Call* is regarded as a new problem for service desk agents to solve.

If the service desk agent could match the *Call* to an open known error, find the solution in the knowledge base, or solve the problem by himself/herself, the problem solution and final report is implemented to the *Caller* directly and the *Call* is ending. If the service desk agent cannot find a solution or solve the problem by himself/herself, this *Call* ticket is escalated to an incident for other suppliers to solve. In this case, the solution and final report is implemented to the *Caller* later after the problem is solved by other suppliers.

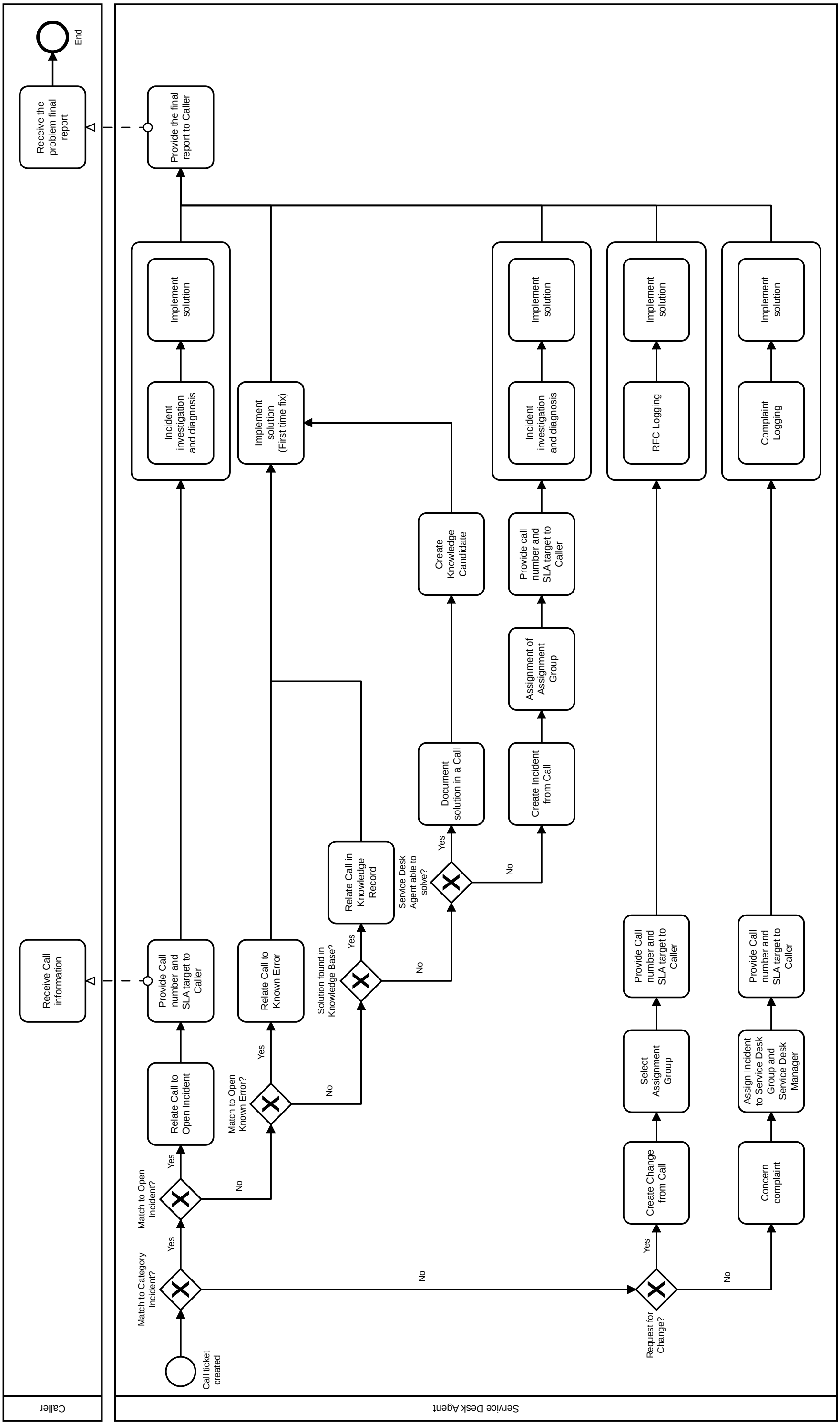


Figure 5.8: BPMN of *Call Solving* in *Call Management* Process

5.4. PROCESS ANALYSIS

SERVICENOW [4] is a platform with numerous applications and features aimed at optimizing business workflows and eliminating tool switching across the enterprise, which is moving towards making work efficient with ticketing[10]. Figure 5.9 is a ticket life cycle sample showing how ServiceNow ticketing works. In this case, the ServiceNow platform is used to manage all tickets in Philips ITGS service desk. All the following data used in the analysis is retrieved from Philips ServiceNow [5].

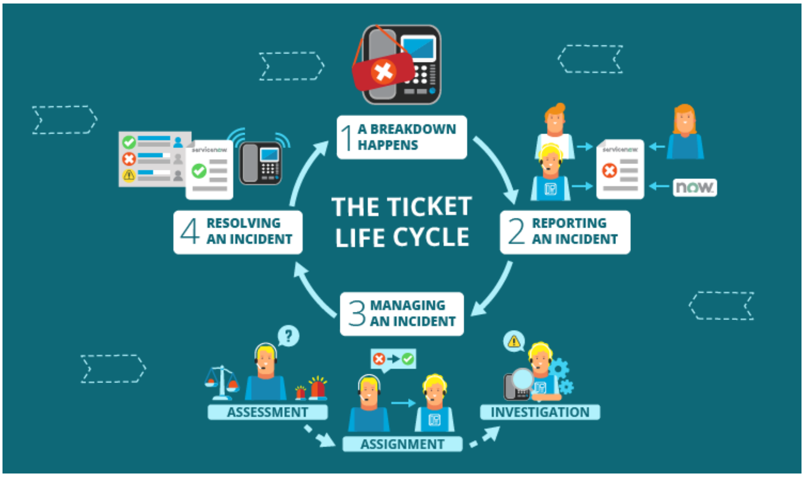


Figure 5.9: ServiceNow Ticket Life Cycle[10]

In order to obtain comprehensive results, this research is focused on all tickets related to *Call Management* from last 6 months (January 2019 to June 2019), 223878 items in total (see part of tickets in Figure 5.10). Log files of the present *Call Management* process were collected from Philips ServiceNow. The analysis of these data was executed by Python (version 3.7.1) coding. During the data processing, libraries Pandas, NumPy, and Matplotlib were employed.

- **Pandas[11]:** A software library for data manipulation and analysis, which offers data structures and operations for manipulating numerical tables and time series.
- **NumPy[12]:** A library for adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Matplotlib[13]:** A plotting library provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

Following functions were used for file and data organizing, operating and plotting:

[4] ServiceNow, Inc. is an American cloud computing company with its headquarters in Santa Clara, California[9].
[5] https://philipsit.service-now.com

```
#file operating:
pd.read_csv  #Read CSV (comma-separated) file into DataFrame
#data organizing and operating
pd.df.rename #Alter axes labels.
pd.df.drop #Drop specified labels from rows or columns.
pd.df.unique() #Check  unique values
pd.df.groupby #Group DataFrame using a mapper or by a Series of columns.
pd.df.set_index #Set the DataFrame index using existing columns.
pd.pivot_table # Create a spreadsheet-style pivot table as a DataFrame.
#data plotting
matplotlib.plt.bar # Make a bar plot.
matplotlib.pyplot.plot # Plot y versus x as lines and/or markers.
matplotlib.pyplot.pie # Plot a pie chart.
```

Listing 5.1: Main functions used in the data analysis

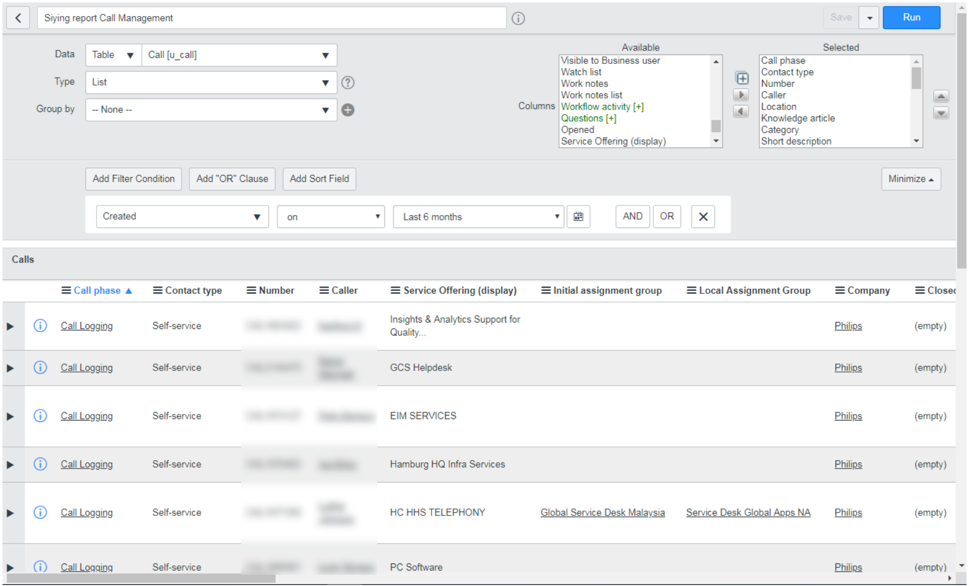


Figure 5.10: Call Management Tickets in ServiceNow

5.4.1. PEOPLE TO SUPPORT PROCESS

A huge number of people are engaged in the process, including *Callers*, *Service Desk Agents*, and external service providers. It results in a great labor costs. Moreover, these people are located in more than [redacted] cities around the world, which further increases the complexity of human resource management as well as the cost of the management. Removing spare labor and improving human resource management hence can be significantly important to enhance the process performance.

5.4.2. PROCESS WORKLOAD

A pretty high volume of workload can also be noticed in the process. Based on the *Call* tickets from January 2019 to June 2019, there are averagely [redacted] tickets per month that need to be handled. As present in Figure 5.11, the number of *Calls*, instead of being equally distributed during the workdays, forms a significant count peak in the Monday of each week and gradually decreases till Friday. This non-equal workload may result in overtime or delay of dealing *Calls*. It also needs to pay attention to that the *Call* created are quite more than the *Call* resolved at the beginning of the year 2019. It can be caused by the holiday, which further increases the workload of the first a few months after the day-off. Another phoneme should be mentioned is that there are still some *Calls* happened (created) in the weekends (see Table 5.1). It also may lead the overtime or workload unbalance of the following week.

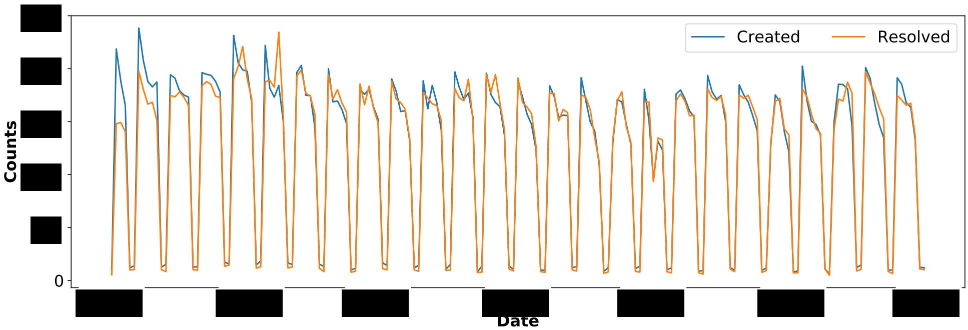


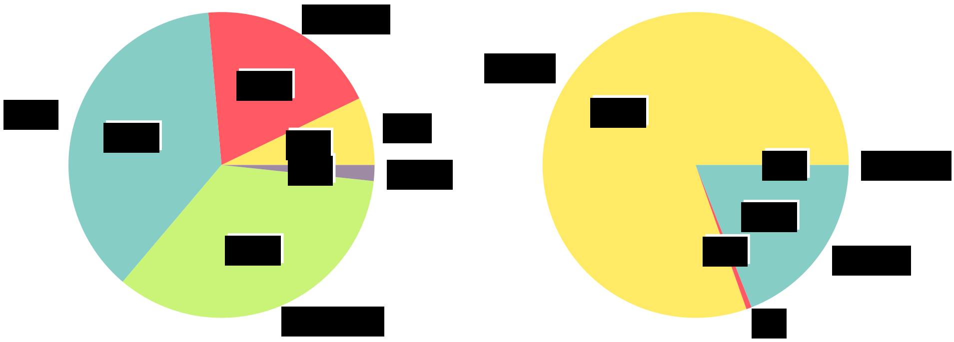
Figure 5.11: *Created* and *Resolved* Counts Line Chart

Table 5.1: *Contact type* Counts for Weekend and Workday

	Weekend	Workday
Email	[redacted]	[redacted]
Lync/Chat	[redacted]	[redacted]
Phone	[redacted]	[redacted]
Self-service	[redacted]	[redacted]
Walk-in	[redacted]	[redacted]

5.4.3. CONTACT TYPE AND CALL CATEGORY

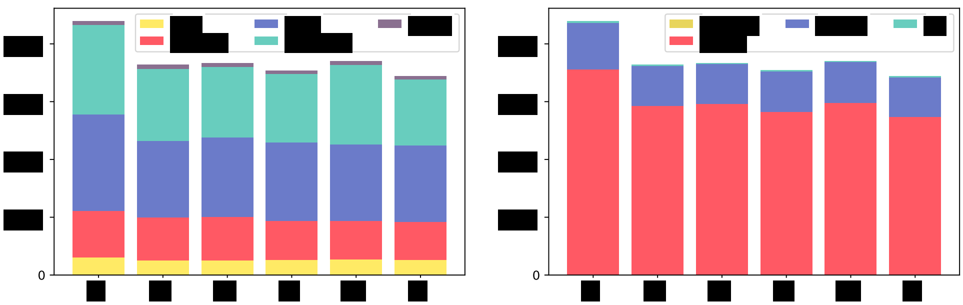
There are five approaches that *Callers* can contact with the *Service Desk Agent*, including *Email*, *Lync/Chat*, *Phone*, *Self-service*, and *Walk-in*. As presented in Figure 5.12, *Callers* are inclined to report their requests by *Phone* which occupies [redacted] of all *Calls*. *Self-service* counts the second share of [redacted] Comparing to the *Phone*, *Self-service* definitely leads less processing complexity, time, and workload. And the shares of each *Contact type* stays nearly constant in each month, as shown in Figure 5.13.



(a) *Contact type* (b) *Category*

Figure 5.12: *Contact type* and *Category* Pie Chart

Regarding to the *Call* category, namely *Incident*, *Question*, *Complaint*, and *RFC*, *Incident* and *Question* take much higher shares, [redacted] and [redacted] respectively, than *Complaint* ([redacted]) and *RFC* ([redacted]) (Figure 5.12). The shares also do not change too much with the time (Figure 5.12). By further investigating how these calls are solved, it is worthy to note that [redacted] of *Question* and [redacted] of *Incident* can be easily solved by the *Knowledge base*, which occupies [redacted] of all the Calls.



(a) *Contact type* (b) *Category*

Figure 5.13: *Contact type* and *Category* Cumulative Bar Chart

5.4.4. PROCESS COLLABORATIONS

The integration of communication and coordination are also required in the process. Different *Service Desk Agent* should be able to share, monitor, communicate, coordinate and collaborate the process progresses. In the process, there are more than [redacted] *Calls* need more than one *Service Desk Agent* to be involved. Due to such collaboration, the solving time of the *Calls* are normally prolonged. There are even some *Calls* keep for more a few months. Besides, the process also are associated with other systems and process, for *Knowledge Management*, *Change Management*, and *Incident Management* processes. The process requires a strategy to handle such frequent collaborations to avoid potential process delay or human error.

5.4.5. LONG-RUNNING PROCESS, TIME AWARENESS AND CALENDARING

As mentioned above and displayed in Figure 5.14, the task of a *Call* generally completed in an undetermined amount of time. Hence, the process should have the strategy to wait, pause, and be interruptible. Such long-running process must be able to support multiple transactions in parallel. Besides, time awareness and calendaring are also essential for the process to keep it moving on and continue when errors occur. Timers are always useful for the process. For example, it is always preferable to give up on a stalled service or non-responding call, rather than putting the whole process at jeopardy, and let it keep running for a long time.

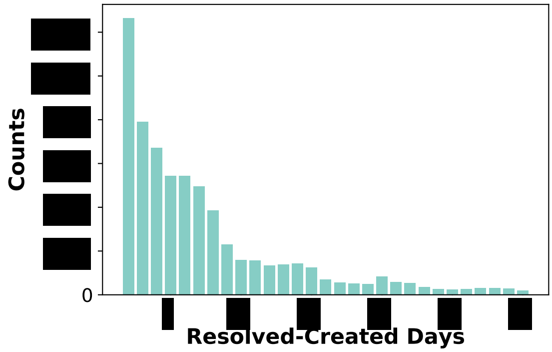


Figure 5.14: *Resolved-Created Days* Bar Chart

5

5.4.6. HUMAN ERROR

During investigating the process, there are some human errors or places are found. The most frequent one is ticket information error. For example, there [redacted] of *Calls* are solved within 1 minute. As displayed in Figure 5.15, most of such *Calls* are solved only within [redacted] which is abnormal. It may result from the *Service Desk Agent* not filling in time correctly and timely. However, such error should be avoided to provide trustful and accurate data for other processes. Besides, other parts of *Call* tickets also exist some errors, for example, the *Contact type* is different from the type recorded in the detail description. And the long drop-down list of ticket filling process also enhances the chance of human errors.

5.4.7. CONCLUSION

Based on the above analysis on the process, implementing ChatBots can be regarded as a good choice to enhance the process preference and efficiency. Employing ChatBots in the process can directly decrease human labor. ChatBots can also work for 24/7, which further decreases the overtime, increases employee satisfaction, and increases process

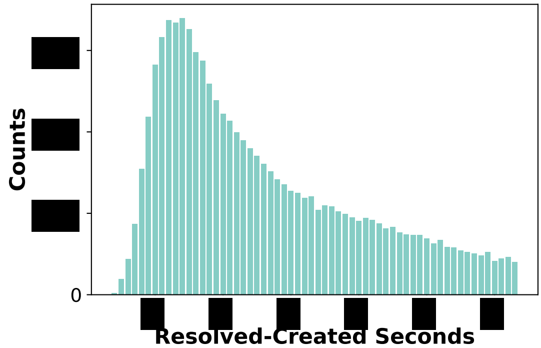


Figure 5.15: *Resolved-Created Days* in 60 Seconds Bar Chart

efficiency. As a result of ChatBots dealing with those repetitive work, such as filling the tickets, people can spend more on work needs more creativity, such as solving new problems or issues. ChatBots are good at solving the knowledge-based problem and handling process associated with other systems, which is also the situation of the *Call Management* process. Timers are also easy to be set and excited by ChatBots. Moreover, ChatBots can efficiently avoid those human errors. In summary, ChatBots is a suitable solution to achieve the Call management process automation and further enhance its efficiency and performance.

5.5. ChatBot-implemented process redesign

The process redesign follows functions evaluation, ChatBots selection, exception human operations, conceptual framework design, and to-be process model. An assessment of the functions is conducted firstly in Section 5.5.1. Then, Section 5.5.2 selects ChatBots based on the functions assessment. Section 5.5.3 describes the exception human operations. Finally, A process redesign conceptual framework is provided in Section 5.5.4 and the to-be process model is described in Section 5.5.5.

5.5.1. Functions evaluation

Choosing suitable ChatBots is significantly essential for the further process redesign. A suitable ChatBot means the ChatBot can deal with the task while using limited resource. For example, an *AI-based ChatBot* can easily fill in a form, however, it is a huge waste of investment and intelligence of the ChatBot in this case. Moreover, *AI-based ChatBot* can perform less efficiently in this case comparing to a *Rules-based ChatBot*, which only performs the task with learning. ChatBots for *Call Management* are selected based on the basic functions of the process: F1 Caller communication, F2 Call information management, F3 Ticket management, F4 Solution management. The complexity and input of those functions or tasks are evaluated by the following criteria:

- Processes which have limited requirement of subjective judgment, creativity or interpretation skills,

- Process that are performed frequently, in huge numbers or with significant peaks in workload,
- Processes which need access to multiples applications and systems to accomplish the task,
- Processes that are highly standardized with limited or no exceptions to handle.
- Processes that are prone to human error owning to manual operation,
- Processes once started, need limited human intervention.

The evaluation result is shown in the Table 5.2.

Table 5.2: Assessment of *Call Management* process functions

	F1 Caller communication	F2 Call information management	F3 Ticket management	F4 Solution management
Processes which have limited requirement of subjective judgment, creativity or interpretation skills	+	++	+++	++
Process that are performed frequently, in huge numbers or with significant peaks in workload	+	++	+++	++
Processes which need access to multiples applications and systems to accomplish the task	+++	++	+	+
Processes that are highly standardized with limited or no exceptions to handle	+	+	+++	++
Processes that are prone to human error owning to manual operation	+	+	+++	+
Processes once started, need limited human intervention	+	+	+++	++

5.5.2. CHATBOTS SELECTION

Based on the complexity and input of each functions, following ChatBots are selected (see Figure 5.16):

F1 Caller communication. An *AI-based ChatBot* is selected for this task. As communication with a human is a complex process, ChatBot needs to handle with natural language, which is always unstructured and in a free form. Hence, an intelligent AI ChatBot is necessary to finish this task.

F2 Call information management. The *AI-based ChatBot* is also used this task. In this process, the ChatBot needs to handle with *Caller*'s request and extract useful information. It also needs support from AI. Meanwhile, the organization of information is a patterned job, which can be done a *Knowledge-based ChatBot*.

F3 Ticket management. A *Rules-based ChatBot* is suitable to achieve this function. Here, the ChatBot only hand with information of *Call*, which has been organized by *AI-based ChatBot* before. The information is well-structured. And the work of filling in the tickets are quite patterned, yet are prone to human error. The task requires less intelligence as well. Moreover, it also frequently performed with the high-volume workload, which is quite suitable to be done by the *Rules-Based ChatBot*.

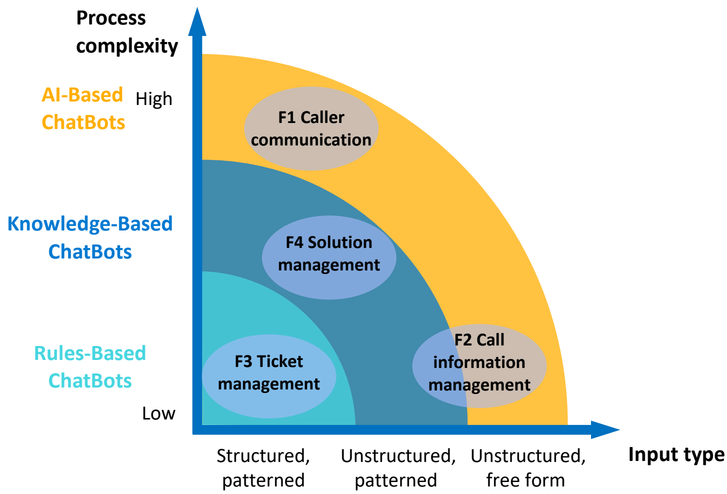


Figure 5.16: Selection of ChatBots for *Call Management* functions

F4 Solution management. The functions of match solution would be executed by a Knowledge-based ChatBot. The ChatBot must match the problem provided by AI-based ChatBot with the solution in the *Knowledge base*. This work deal with unstructured information but in the patterned approach. Therefore, a Knowledge-based can play a role in this task.

5.5.3. EXCEPTION HUMAN OPERATIONS

It is unavoidable that ChatBot-related errors may occur during the process performing. Hence, exception human operations are needed to deal with those situations. Here, *External service* (different from service provided by ChatBots) and *Process manager*. *External service* can deal with *Calls* that ChatBots cannot successfully solved, for example, ChatBots cannot not understanding *Callers*' requests. *Process manager* is responsible to the process-related errors and issues, for example, ChatBots fail to work due to hardware or software issues.

5.5.4. CONCEPTUAL FRAMEWORK DESIGN

Figure 5.17 illustrates the redesigned conceptual framework of automated *Call Management* process by implementing three types ChatBot agents, i.e., *Dialog Manager*, *Inference Engine*, and *Ticket Manager*. Besides, *Caller*, *External Service*, *Process Manager*, together with a *Knowledge Base* are also involved in the new process. The notion of commitments are used to present the interaction among these seven components as presented in Figure 5.17. These interactions are:

C1 = C(*Dialog Manager, Caller, request, information*),
C2 = C(*Inference Engine, Dialog Manager, Information, intent*),
C3 = C(*Knowledge Base, Inference Engine, intent, solution or no matched solution*),
C4 = C(*Dialog Manager, Inference Engine, no matched solution, new problem*),

C5 = C(*External Service, Dialog Manager, new problem, new solution information*),
C6 = C(*Inference Engine, Dialog Manager, new solution information, solution*),
C7 = C(*Knowledge Base, Inference Engine, new problem and solution, knowledge*),
C8 = C(*Ticket Manager, Inference Engine, intent and solution, ticket*),
C9 = C(*Dialog Manager, Inference Engine, intent and solution, report*),
C10 = C(*Caller, Dialog Manager, report, confirmation*),
C11 = C(*Dialog Manager, Inference Engine, error, error alert*),
C12 = C(*Process Manager, Dialog Manager, error alert, operation*).

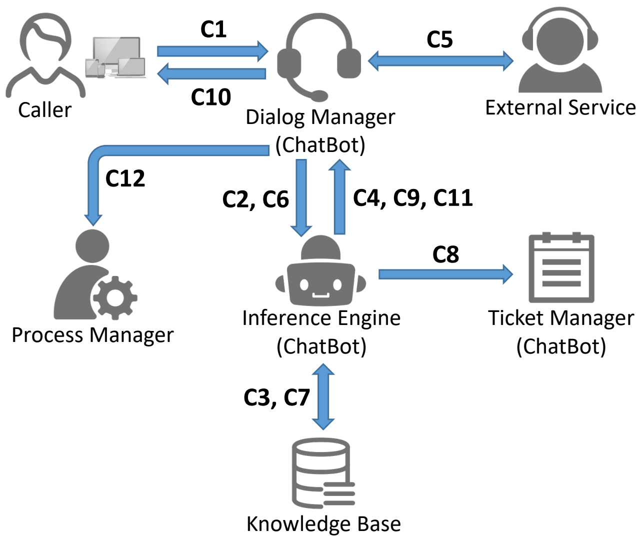


Figure 5.17: *Call Management* Process Redesign Conceptual Framework

The *Dialog Manager* is an *AI-based ChatBot*, it builds a bridge to connect human users (*Caller*, *Process Manager* and *External service*) and machines. To communicate with human, the *Dialog Manager* ChatBot should be equipped with a natural language processing (NPL) engine. The *Dialog Manager* can both extract information from human's natural language request, which can be further handled by machines, and send the solution back in the form of natural language.

The *Inference Engine* is a *Knowledge-based ChatBot*, which holds abilities of machines learning and/or deep learning. Such abilities enable the *Inference Engine* to identify a closest possible intent for a given request. The *Inference Engine* can infer an appropriate solution from *Knowledge Base* and deliver the solution to *Dialog Manager* and *Ticket Manager*.

The *Ticket Manager* is a *Rules-based ChatBot*. It is a less simple ChatBot compared to above two. It manages the contents and status of all *Call* tickets. The *Ticket Manager* receive the information of *Caller*, request, and request solution from *Inference Engine*, and fill in them into *Call* ticket.

Any user who contacts *Service Desk* is *Caller*. *External Service* is human service desk agents, handles the requests cannot be solved by ChatBots (interact with *Caller* directly when *Dialog Manager* having trouble to communicate with *Caller* and solve the problem when *Inference Engine* cannot find solution from *Knowledge Base*). *Process Manager* treats errors from all ChatBots. And *Knowledge Base* is a database stored request solution knowledge.

According to this automated conceptual framework of *Call Management* process. *Caller* contacts *Dialog Manager* when problem occurring, *Caller* provides request and *Dialog Manager* translates the request into machine-understandable information (C1). *Dialog Manager* then transfers the machine-understandable request information to *Inference Engine* (C2). *Inference Engine* extracts intent of the request information and try to match solution in *Knowledge Base* (C3). If there is matched solution in *Knowledge Base*, the request information as well as the solution will be provided to *Ticket Manager* to fill in *Call* ticket (C8). If *inference Engine* finds no solution in *Knowledge Base*, the "no matched solution" information will be transferred to *Dialog Manager* (C4). *Dialog Manager* then reports this "no matched solution" as a new problem to *External Service*, and waits for solution from *External Service* (C5). After receiving new problem solution, *Dialog Manager* contacts *Inference Engine* again to provide the new problem solution information (C6). *Inference Engine* creates new knowledge in *Knowledge Base* based on the new problem solution information (C7) and transfers all the information to *Ticket Manager* to fill in *Call* ticket (C8). In addition, *Inference Engine* provides all the request information (solution included) to *Dialog Manager* (C9). *Dialog Manager* creates a final human language report to *Caller* at the end (C10).

In case of any unexpected error happens with ChatBots, *Dialog Manager* will create an error alert to remind *Process Manager* for further operation (C11, C12).

5.5.5. TO-BE PROCESS MODEL

The *Call Management* to-be process model is drawn according to the process redesign conceptual framework (Figure 5.17 in Section 5.5.4), and presents in Figure 5.18. In this to-be process model, there are five different roles: *Caller*, *External Service*, *Dialog Manager*, *Inference Engine*, and *Ticket Manager*. The first two are human agents, and the other three represent three different types of ChatBots.

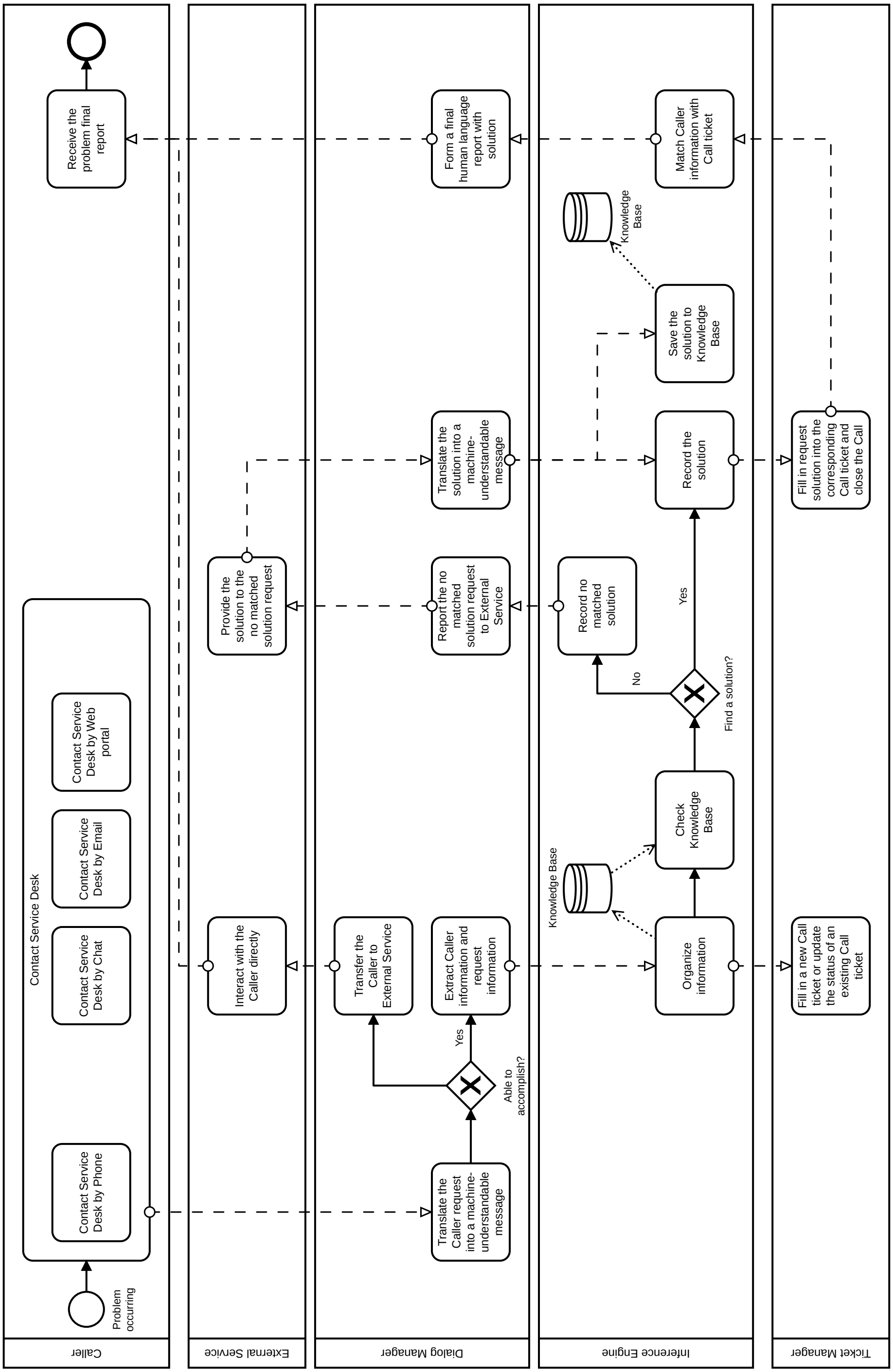


Figure 5.18: BPMN of ChatBots Implemented *Call Management* Process

5

Caller contact *Dialog Manager* through *Phone*, *Chat*, *Email*, or *Web portal* when problem occurring. *Dialog Manager* interacts with *Caller* to obtain sufficient information and translates the information into machine-understandable message. In situations that *Dialog Manager* cannot understand the *Caller* or obtain the essential information, the *Caller* will be transferred to human agents (*External Service*) directly. The *Call* would then be handled by human agents.

After receiving sufficient and correct information from *Dialog Manager*, *Inference Engine* would organize the message for *Ticket Manager* filling *Call* ticket. *Ticket Manager* fills a new *Call* ticket or update the status of an existing *Call* ticket if this problem have relation to an existing *Call*. *Inference Engine* should also check the *Knowledge Base* to see whether a matched solution can be found and record the solution directly if there is a matched solution. If solution cannot be found in *Knowledge Base*, this problem should be solved by human agents (*External Service*). In this case, *Inference Engine* contacts *Dialog Manager*, and *Dialog Manager* interacts with *External Service* to obtain the solution. The new solution from *External Service* should be saved in *Knowledge Base* by *Inference Engine*. At last, *Inference Engine* passes on the solution to *Ticket Manager*. *Ticket Manager* complete the *Call* ticket and close it. *Dialog Manager* will contact *Caller* to express the final report.

5

There can be some situations when errors appear and ChatBots cannot complete the current activity. In this case, *Process Manager* (human agents) can help to handle errors. The error related procedures are removed in the above to-be process model (Figure 5.18) to make the graph more readable. Figure 5.19 is the partial model including error procedures.

5.6. EVALUATION

THE evaluation part evaluates the redesigned process by comparing the process complexity of the as-is process model and the to-be process model. Three aspects of process complexity, execution complexity, coordination complexity, and business item complexity are introduced and computed here.

According to the description of BPMN of to-be model, basic functions of the *Call management* process can be theoretically completed by the to-be process. Then, the performance of the to-be model was evaluated to check the effectiveness of the methodology. The evaluation was based on the IT service process complexity framework[14, 15]. The complexity of the IT process represents a major impediment to efficient, high-quality, error-free, and cost-effective service delivery [15, 16]. A quantitative analysis on the IT service process complexity can provide the a measurable process performance improvement indicator[16]. The framework offers a quantitative approach to determine IT service process complexity, which includes three aspects of the process complexity: **Execution Complexity**, **Coordination Complexity**, and **Business Item Complexity** [15].

Execution Complexity refers to the complexity involved in performing the activities that make up the process. It indicates the complexity of the activity according to its execution type. Values for this score are assigned according to a weighting scale of different types. For an activity involving R roles ($r = 1, 2, \ldots, R$), its execution complexity is computes as

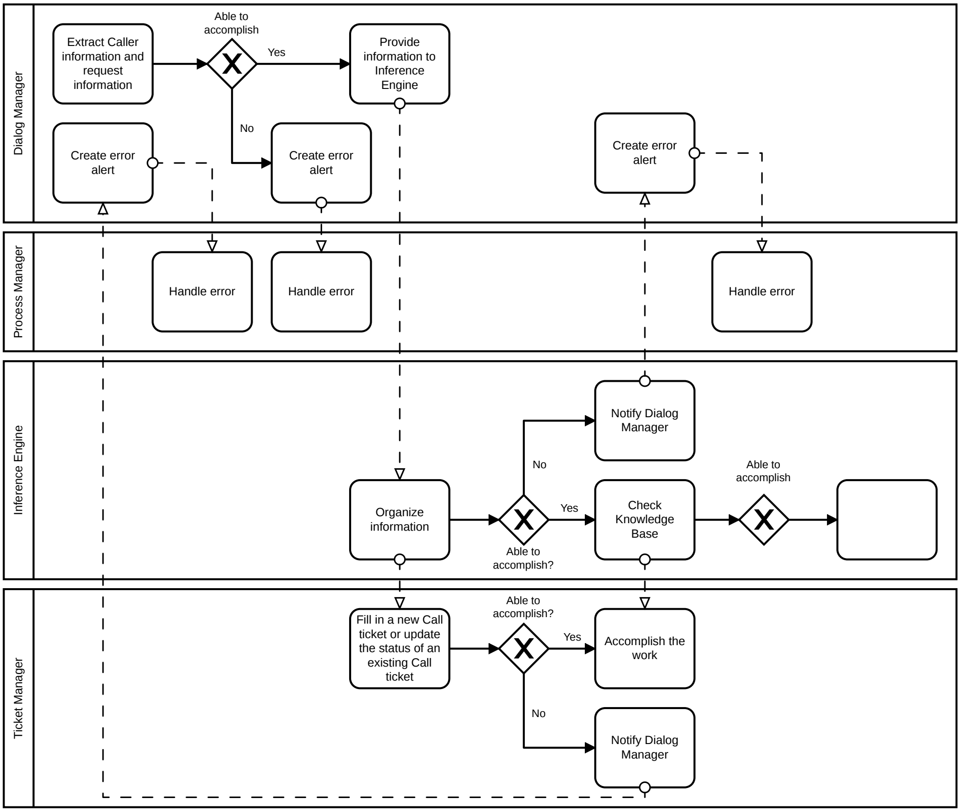


Figure 5.19: Partial BPMN of ChatBots Implemented *Call Management* Process (including Error Procedures)

$$E_{base} = \sum_{r=1}^{R} execType(r) \tag{5.1}$$

where the execution type *execType(r)* is defined as three types: *automatic [0]*, *tooAssisted [1]*, and *manual [2]*.

Coordination Complexity represents the complexity of coordination between multiple roles. Score values are assigned according to a weighting scale of different coordination link complexity. The coordination complexity for an activity is the sum of values from all the links *(l = 1, 2,..., L)* multiplied by the number of roles *(R)* involved in.

$$C_{link} = R \times \sum_{l=1}^{L} linkType(l) \tag{5.2}$$

where the link type *linkType(l)* is defined as xx type: *autoLink [0]*, *controlLink [1]*, *dataTransferred [2]*, and *dataAdapted [3]*.

Business Item Complexity indicates the complexity involving business items. Values for this score are assigned according to a weighting scale of different source complexities. For an activity involving *R* roles *(r = 1, 2,..., R)*, producing business items *(i = 1, 2,...,* I_p*)*, and fields *(f = 1, 2,...,* F_i*)*, the business item complexity is computes as

5

$$B_{source} = R \times \sum_{i=1}^{I_p} \sum_{f=1}^{F_i} sourceScore(i, f) \tag{5.3}$$

where *sourceScore(i, f)* is defined as: *internal [0]*, *freeChoice [1]*, *documentationDirect [2]*, *documentationAdapted [2]*, *bestPractice [4]*, *environmentFixed [5]*, and *environmentConstrained [6]*.

5.6.1. AS-IS PROCESS

The as-is *Call Management* process model is described in section 5.3 (Figure B.1 and Figure B.2). All activities in this as-is model is numbered from 1 to 39 for evaluation, and the numbered as-is model presents in Appendix B. The complexity evaluation of as-is model per-activity shows in Figure 5.20.

For the **Execution Complexity**, only one role (*Caller* or *Service Desk Agent*) involves in each activity, so the value of E_{base} is equal to the summation value of *execType(r)*. The *Call* ticket will be filled automatically if the *Service Desk Agent* select the suitable *Call Model*. So the activity *Select Call Model* is manual but tool-assisted, its *execType(r)* value is *toolAssisted [1]*. And all the other activities are fully manual work, their values of *execType(r)* are all *manual [2]*. The summation value of *execType(r)* as well as the value of E_{base} is 77 then.

Like the execution complexity, the value of **Coordination Complexity** (C_{link}) is equal to the summation value of *linkType(r)*. The activity *Select Call Model* is linking to an automated task, its *linkType(r)* value is *autoLink [0]*. *linkType(r)* value of activities (*Contract Service Desk by Phone*, *Receive problem*, *Provide missing information*, *Receive Call information*, etc.) involving message transferring are *dataTransferred [2]*. While the other activities are control flows link to non-automated tasks, their *linkType(r)* value are *controlLink [1]*. The C_{link} value is 56 in the end.

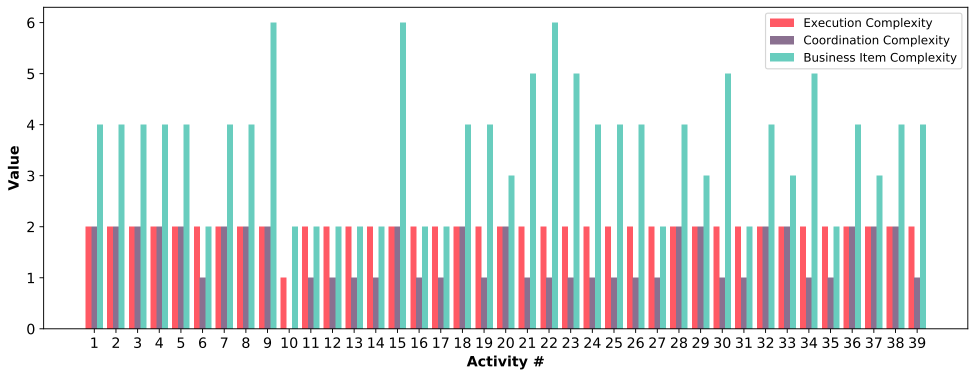


Figure 5.20: Complexity Evaluation of as-is *Call Management* Process Model

For **Business Item Complexity**, *sourceScore(i, f)* have seven different types of values. Activities whose field values are taken directly from documentation such as *Select Caller name, Select affected service, Select Assignment Group, Assign Incident to Service Desk Group and Service Desk Manager* have *sourceScore(i, f)* of *documentationDirect [2]*. *bestPractice [4]* is suitable for *Contract Service Desk by Phone, Receive problem, Receive Call information, Provide Call number to Caller*, etc. *Relate Call to Known Error* and *Relate Call in Knowledge Record* are the activities whose field values are constrained by the environment, *sourceScore(i, f)* value of is *environmentFixed [5]*. And *sourceScore(i, f)* value of *Provide missing informaion, Provide status update to business user and update call ticket*, and *Implement solution* are *environmentConstrained [6]*, the field values are constrained by the environment to a limited set of possible choices. The B_{source} value is 140 in the end.

5.6.2. TO-BE PROCESS

The to-be *Call Management* process model have five different roles, *Caller* and *External Service* are human, *Dialog Manager*, *Inference Engine*, and *Ticket Manager* are three different kinds of ChatBots. Apart from drawn activities in Figure 5.18, every automated activity also has an additional activity with *Process Manager* responsibility. *Process Manager* will handle errors happen with machine. Figure 5.21 is the complexity evaluation of to-be model per-activity. Activities 5 to 26 are have been automated (ChatBots related activities), so have zero complexity associated with them.

For the **Execution Complexity**, human related (*Caller* and *External Service*) activities are all manual work, their value of *execType(r)* is *manual [2]*. ChatBots related (*Dialog Manager*, *Inference Engine*, and *Ticket Manager*) activities are all automatic work, their value of *execType(r)* is *automatic [0]*. The execution complexity value of the to-be model is 32 as a result.

As for **Coordination Complexity**, the *linkType(r)* value of human related activities are *dataAdapted [3]*, the transferred business items need to be adapted. And ChatBots related activities are linking to automated tasks, the values of *linkType(r)* are *autoLink [0]*. The coordination complexity value of the to-be model is 48 as a result.

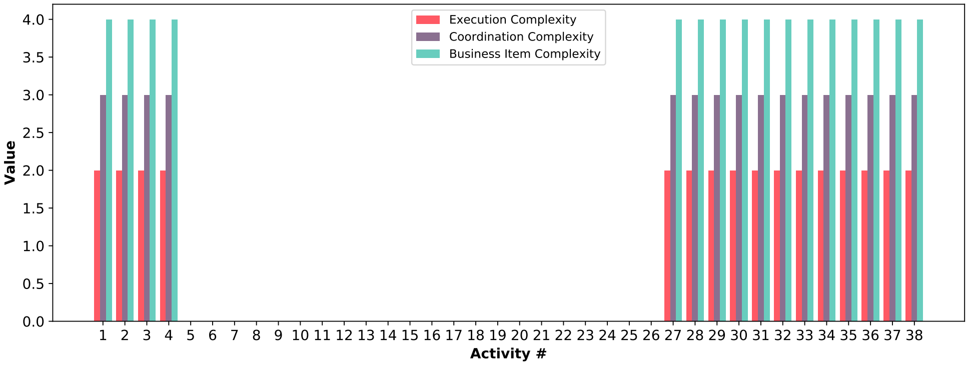


Figure 5.21: Complexity Evaluation of to-be *Call Management* Process Model

For the **Business Item Complexity**, the *sourceScore(i, f)* value of human related activities are *bestPractice [4]*. The field values of ChatBots related activities are produced from automation, the *sourceScore(i, f)* values are *internal [0]*. The business item complexity value of the to-be model is 64 as a result.

5.6.3. EVALUATION CONCLUSION

According to the per-activity complexity results show in Figure 5.20 and Figure 5.21, The overall *Call Management* process complexity of as-is model and to-be model are summarized in Table 5.3.

Table 5.3: Complexity Comparison of *Call Management* as-is Model and to-be Model

	as-is Model	to-be Model
Execution Complexity	77	32
Coordination Complexity	56	48
Business Item Complexity	140	64

Since *AI-based ChatBot*, *Knowledge-based ChatBot*, and *Rules-based ChatBot* are implemented in the redesigned process, 22 out of 38 activities in the to-be model are fully automated work. As a result, the execution complexity and business item complexity have reduced by more than half. The execution complexity is 77 for the as-is model, and 32 for the to-be model. The business item complexity is 140 for the as-is model and 64 for the to-be model. The level of automation and field's value supplying have a huge improvement with *Call Management* redesigned process then.

Concerning the coordination complexity, there is no much difference of the as-is model and the to-be model, whose value is 56 and 48 respectively. This is mainly because of coordination complexity concerning not only coordination links but also business items transferring. Business items need to be adapted when they are transferred between human and ChatBots. Although coordination links are declined with ChatBots implementation (the *linkType(l)* of ChatBots related activities are *autoLink [0]*), business items transferring complexity is increased for the other activities.

5.7. CONCLUSIONS

DUE to the reality of the company and duration of the thesis, steps including Process implementation, Process monitoring and controlling have not been carried out. Nevertheless, some recommendations on the execution of these steps would be provided.

The evaluation of ChatBot-implementation methodology starts from explaining a goal about process automation of Philips IT Global Services (ITGS). And the *Call Management* process was chosen as the case for the application of proposed ChatBot-implementation methodology. As *Call Management* is a service-oriented process involving plenty of interactions with human (customers and agents), it can be a good example of ChatBot implementation.

The current situation of the *Call Management* process was firstly identified, including the major functions and activities of the process. The functions of the process can be summarized as *Caller communication*, *Call information management*, *Ticket management*, and *Solution management*. And the as-is model was established by BPMN to depicted the current situation.

Then the process analysis was carried out based on the study of data generating from the process. Facts and issues of the current process were the outcome of the process analysis, including huge human labor involved in the process, heavy and unbalanced workload, preference phone and self-services, major knowledge-based questions or incident, required integration of communication and coordination, process working time, and some human-related errors. Those facts and issues made ChatBot a suitable tool to improve the situation. And in the Process redesign step, functions were evaluated to select the suitable ChatBots. Caller communication and Call information management can be carried out by a *AI-based ChatBot*, Ticket management can be executed by a *Rules-based ChatBot*, and a *Knowledge-based ChatBot* can be applied for Solution management. Some potential issues were also briefly discussed and introduced some human exceptional operations to deal with those situations. The conceptual ChatBot implementation framework was then established. The interactions among different component were described by the notion of commitment. To-be model was illustrated based on the conceptual framework. The to-be process could complete all the basic functions and decrease the process complexity based on the evaluation results.

However, further Process implementation, Process monitoring and controlling remain to be carried in the future. And here are some suggestions for these steps. The to-be model needs to be converted into an executable model for the implementation. The selection of the executable model should be based on the vendors of ChatBot service, owning to the different ChatBot building framework. The practical implementation of ChatBot can start from the low-cognitive bot (such as Rules-based ChatBot), which are much easier to achieve and has a lower impact on the currently running process. Fully automation is almost impossible, other situation of human engagement shall be carefully studied during process implementation. Monitoring and controlling of the new process are of great importance, especially in the case of applying ChatBot tools. The process issue related to ChatBot can be easily observed through process-running data.

Gathering and analyzing the process running data can help the process manager to predict potential process incident, to further improve the process and ChatBot efficiency.

The evaluation part provides an example of an IT service process (*Call management*) automation based on the proposed ChatBot-implementation methodology. The methodology depicts a feasible and clear routine to carry out the process automation by using ChatBots. A human and ChatBot collaborative *Call management* process model was established by following the methodology. The implementation of ChatBots reduce human labor in the process and decrease the process complexity. Further recommendations on practical process implementation were also provided.

REFERENCES

[1] Royal Philips, *More than a century of innovation and entrepreneurship,* .

[2] G. Hinde, *Company profile,* .

[3] Royal Philips, *Philips Business System*, Tech. Rep. (2016).

[4] M. Rouse, *Definition of ITSM,* (2016).

[5] M. Rouse, *Definition of ITIL,* (2014).

[6] M. Rouse, *Definition of Service Desk,* (2017).

[7] M. Colenbrander, *Service Desk and Process Automation Kickoff*, Tech. Rep. (2018).

[8] B. Gurram, *IT Process Description - Call Management (Philips ITGS internal use)*, Tech. Rep. (2016).

[9] *ServiceNow,* .

[10] P. Rodin, *ServiceNow Ticketing – Superfluous or Essential?* .

[11] *pandas (software),* .

[12] *NumPy,* .

[13] *Matplotlib,* .

[14] N. Ayachitula, M. Buco, Y. Diao, S. Maheswaran, R. Pavuluri, L. Shwartz, and C. Ward, *IT service management automation - A hybrid methodology to integrate and orchestrate collaborative human centric and automation centric workflows,* Proceedings - 2007 IEEE International Conference on Services Computing, SCC 2007 , 574 (2007).

[15] Y. Diao and A. Keller, *Quantifying the complexity of IT service management processes,* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2006).

[16] Y. Diao and K. Bhattacharya, *Estimating business value of it services through process complexity analysis,* in *NOMS 2008-2008 IEEE Network Operations and Management Symposium* (IEEE, 2008) pp. 208–215.

5

6
Conclusions

If you don't build your dream, someone else will hire you to help them build theirs.

Dhirubhai Ambani

This chapter concludes the research findings, contributions, limitations, and further research of this research. The answers of the five sub-questions are answered in Section 6.1. Section 6.2 describes the contributions and lists the limitations of the research. For further research on ChatBot-implemented process automation, Section 6.3 focuses on the executable process model, the business process model towards the ChatBot model, the extension of methodology application, the multi-process ChatBot implementation methodology, and ChatBot-based business process management.

6.1. RESEARCH FINDINGS

THE research findings would be introduced by answering the research questions proposed in Section 1.3.

Q1: What is the value of BPM theory for process redesign?

A business process can be redesigned based on the BPM lifecycle, which is introduced in the literature review. The lifecycle consists of six steps, including process identification, process discovery, process analysis, process redesign, process implementation, and process monitoring and controlling. The architecture is obtained from process identification, and then the process as-is model is acquired by the process discovery. Process analysis aims to get insights on the process weaknesses and their impact. After that, the process is redesigned to overcome these weaknesses and is represented by the to-be process model. The to-be model then is converted into an executable model in the process implementation stage. Process monitoring and controlling are carried out with the new situation of the process to discover the new issues. The new process can still get into another new cycle to further improve it.

Business process modeling is one of the key parts of the BPM. It provides methods to depict the process in a structured and analyzable form. Some of the most common modeling techniques, including BPMN, UML ADs, EPCs, and WS-BPEL, are introduced in Section 2.1.3.

Q2: What is the state of art of ChatBots?

Selecting suitable ChatBots is of great importance to make ChatBot implementation cost-effective. In this thesis, ChatBots are categorized into three types namely *Rules-based ChatBots*, *Knowledge-based ChatBots*, and *AI-based ChatBots*. The ChatBot is chosen based on the complexity and input of the task. *Rules-based ChatBots* can deal with well-structured, highly-patterned, and less complex tasks. *Knowledge-based ChatBots* can deal with works that are more complex but still have patterned inputs. Complicated tasks with unstructured input can be handled by *AI-based ChatBots*. The selection of ChatBot vendors, such as IBM, Microsoft, Google, and Amazon, should be conducted based on the real situation of the company. As indicated in Section 2.2.4, all of them provide similar intelligent ChatBots services. Other factors, such as cost, system compatibility, collaboration etc., should be considered to choose the vendor.

Q3: What are the current methods of implementing ChatBots in business processes?

Among the limited methodologies of ChatBot implementation in the business process, Quark methodology provides a systematic approach. It mainly consists of following steps, 1) identify roles that can be automated by ChatBot, 2) identify goals of those roles, 3) identify commitments of the roles, 4) produce a set of interactions, 5) produce additional goals and commitments, and 6) Translate the interactions to ChatBots vendor's model. This methodology establishes a feasible routine to conduction ChatBot implementation. However, it does not design the ChatBot implementation in the whole process level, analysis and redesign of the process and exceptional human engagement are missed in the methodology.

Q4: What should the ChatBot-implementation methodology look like?

The proposed ChatBot-implementation methodology is described in Chapter 4, following the analytical framework including the way of thinking, way of working, and way of controlling. The way of thinking lays the fundamental principles and the underlying structure that the design methodology is based on. The way of working describes the routine from the current situation towards the new situation. The way of modeling states the approach of depicting the real situation with an abstract model that is suitable for the systematical analysis. The methodology consists of three phases, namely understanding, design, and implementation phase. They are proposed based on the BPM lifecycle. The understanding phase includes process identification, discovery, and analysis. The goal of the understanding phase is to acquire the key functions of the process, an as-is process model, and limitations or issues of the as-is process. After that, the process is redesigned in the design phase, in which the suitable ChatBots would be chosen for specific functions, interactions between within the system would be presented in a conceptual ChatBot design, a to-be process model would be provided evaluated in the end. In the final implementation phase, the to-be model would be translated into an executable model and ChatBot design model for the real process. During the new process running, process monitoring and controlling would be done to figure out risks and issues, and to further improve the process.

Q5: What are the values and limitations of the methodology in practice?

The Call Management of Philips IT service was used for the evaluation of the proposed methodology. The objectives and key functions of Call Management were first obtained by process identification. And the as-is model was illustrated by BPMN. Process analysis was carried out by investigating the process-generated data to find out issues of the current process. After that, ChatBots were selected for each function, i.e., an AI-based ChatBot for Caller communication and part of Call information management, a Rules-based ChatBot for Ticket management, and a Knowledge-based ChatBot for Solution management and part of Call information management. Interactions within the process were described by the notion of commitments. And a to-be process model was established. Based on the model-based evaluation, a decrease in process complexity can be noticed. The understanding and design phases can be smoothly carried out in the real service-oriented process. However, the implementation could not be finished owning to reality and time. But some recommendations on implementation were provided in Section 5.7.

6.2. CONTRIBUTIONS AND LIMITATIONS

THIS part analyzes the contributions and limitations of this research. Seven aspects of limitations are discussed here.

6.2.1. CONTRIBUTIONS

The proposed methodology established a theoretical ChatBot-implementation framework in service-oriented processes. Comparing to other ChatBot-implemented process automation methodologies, the proposed methodology in this research is a combination of BPM and ChatBot design methods. The BPM methods ensure a standardized redesigned process. The ChatBot design enables the cost-effective of the new process.

Moreover, the methodology is not human- nor ChatBot-centric. The collaboration of human and ChatBots are the other core of the methodology. ChatBots are not designed to replace part of human labor, but to assist human. The methodology emphasis that full automation is impossible in the current stage. Exceptional human operations should be considered during process design.

6.2.2. LIMITATIONS

This research is the first step of development on the ChatBot-implementation methodology. Some limitations remain to be overcome.

1) The approach of building an executable process model

The approach of translating the to-be process model to the executable process model for Process implementation is not given in this research. Owning to the involvement of ChatBots, the executable process model should take the design model of ChatBot into consideration. And the ChatBot design model is always determined by the service vendor. The design of the model may only be done after choosing the ChatBot vendor.

2) Impacts on related processes

The methodology currently does not take the impacts on and from the other related processes into consideration. Actually, all the business processes are kind of related to other processes. Hence the relations of them may also influence the process redesign. The introduction of ChatBot may also affect other processes. Without comprehensive process architecture investigation and practical process implementation, impacts of those relations can be difficult to find out.

3) Process monitoring and controlling

Process monitoring and controlling strategies are hard to determine without process implementation. New measures and indicators may need to be established for the new ChatBot-implemented process to monitor and control the process. They might be kind of different from the conventional case. Because of the behavior of ChatBots, which are unpredictable, they also need to be monitored. And the evaluation of the process may not be limited in time, cost, efficiency, et al., other indicators should be established to evaluate ChatBots.

4) ChatBot-related errors

In this methodology, strategies to deal with ChatBot-related errors are not provided. It is inevitable that some issues can occur with the ChatBots, such as the equipment power down and network disconnected to ChatBot servers. Those fatal issues can lead the whole process system down and may induce a huge loss of the company.

5) Additional investments

The implementation of ChatBots can be associated with investments in IT infrastructures and ChatBot facilities. Such investments have not been discussed in the research. It can influence the cost-effectiveness of the ChatBot implementation. The investments should be estimated prior to the ChatBot implementations.

6) Users acceptance

The implementation of ChatBots can lead to a big change to the process. The users acceptance of the new process needs to be considered during the evaluation, which is not discussed in this research. Users include agents who provide service and customers who require the service. A lower degree of user acceptance can induce a poor efficiency of the process. Meanwhile, additional costs may be needed to train users using the new process in order to help them to accept the new process.

7) Information security and data protection

Information security and data protection are not discussed in the research. Massive data are generated and dealt with in the ChatBot-implemented process. The management and protection of those data are of great importance to ensure the information security both for users and the company. Meanwhile, the ChatBot service is normally provided by commercial vendors, which may arise additional information and data security issues.

6.3. FURTHER RESEARCH

IN this part, further research is discussed on the executable process model, the business process model towards the ChatBot model, the extension of methodology application, the multi-process ChatBot implementation methodology, and ChatBot-based business process management.

1) Executable process model

A suitable executable ChatBot-implemented process model is crucial from the process implementation. Developing an approach of generating an executable process model from the to-be process model (BPMN) can be helpful for further process implementation. In the case of the IT service process, BPEL, as discussed in Section 2.1.3 can be a good choice for the process implementation. Also, the mapping from BPMN to BPEL is available. Further research can focus on integration the BPEL modeling method into the methodology to provide the approach for process implementation.

2) Business process model towards ChatBot model

Business process modeling is powerful to represent a business process, but it cannot be directly used for ChatBot design. To realize the ChatBot application in the process, an approach is needed to translate the business process model (in the form of BPNM or BPEL) to a ChatBot model provided by the ChatBot vendors. However, different vendors provide various routines to establish a ChatBot system. The model-translation approach needs to be designed based on the ChatBot vendor. It somehow increases the workload of the methodology development. A general translation strategy can be beneficial for ChatBot-implementation methodology development.

3) Methodology application field extension

In this research, the methodology is applied in the service-oriented process. Further research can attempt to deploy the methodology to other types of business processes, such as finance and accounting, human resource (HR) management, procurement, and warehousing, etc. These types of processes also involve high-volume human labor as well as interactions with the human. The implementation of ChatBot can increase the efficiency of those processes. Research is needed to explore the possibility of apply methodology in this report to those fields.

4) Multi-process ChatBot implementation methodology

A process is normally connected with more than one other processes. Hence, the implementation of ChatBot into a single process can induced impacts on correlated processes, which can be either positive or negative. A methodology for multi-process ChatBot implementation, which considers the whole system, can be beneficial to the processes automation and ensure high consistency of all related processes. Such an attempt is associated with a highly complex and complicated investigation. Even though, the multi-process ChatBot implementation methodology can accelerate a systematic process automation proceeding of the enterprise.

5) ChatBot-based business process management

The application of ChatBots should not be limited to provide service to customers, but also to the service providers. For example, ChatBots can be a good tool to conduct process monitoring. As the machine is sensitive to variable change in the system, ChatBots can monitor the process running by those process-related variables. The ability of ChatBots gathering and analyzing process-running data can also help the process manager to predict potential risks and issues of the process.

A

DATA ANALYSIS CODES

```
# Load the data file
df = pd.read_csv('call.csv', encoding = 'ISO-8859-1')

# Rename the values of columns
df.rename(columns={'u_call_phase':'Call phase', 'u_call_category':'Category', 'contact_type':'Contact type', 'Opened_at':'Opened', 'sys_created_on':'Created', 'u_resolved_at':'Resolved', 'sys_updated_on':'Updated', 'u_call_state':'Call state', 'impact':'Impact', 'priority':'Priority', 'location':'Location', 'assignment_group':'Assignment group', 'time_worked':'Time worked', 'u_is_reassigned':'Is (re)assigned', 'number':'Number', 'u_service_offering__report_':'Service Offering', 'u_knowledge_article':'Knowledge article', 'short_description':'Short description', 'description':'Description', 'sys_mod_count':'Updates', 'u_solution':'Solution', 'u_caller':'Caller', 'assigned_to':'Assigned to'}, inplace=True)

# Set the date to date type value
df['Opened'] = pd.to_datetime(df['Opened'], dayfirst=True)
df['Created'] = pd.to_datetime(df['Created'], dayfirst=True)
df['Resolved'] = pd.to_datetime(df['Resolved'], dayfirst=True)
df['Updated'] = pd.to_datetime(df['Updated'], dayfirst=True)

# Create new columns for sorting data
df['Created date'] = df['Created'].dt.date
df['Resolved date'] = df['Resolved'].dt.date
df['Created month'] = df.apply(lambda x:x['Created date'].strftime('%Y-%m'), axis=1)

# Create new columns for date string
df['Created date str'] = df.apply(lambda x:x['Created date'].strftime('%Y-%m-%d'), axis=1)
# Some items are still unresolved and don't have resolved dates
def transfer(row):
    try:
        datestr = row['Resolved date'].strftime('%Y-%m-%d')
    except:
        datestr = None
    return datestr
```

A

```
df['Resolved date str'] = df.apply(transfer, axis=1)

# Keep data from January 2019 to June 2019 only
df = df.drop(df[df['Created month']=='2019-07'].index)
```

Listing A.1: Codes for Loading Data

```
# Check the unique values of columns
df['Call phase'].unique()
df['Contact type'].unique()
df['Category'].unique()
df['Call state'].unique()
df['Impact'].unique()
df['Priority'].unique()
df['Assignment group'].unique()
df['Service Offering'].unique()
df['Location'].unique()
```

Listing A.2: Codes for Generating *Category* Pie Chart

```
# Set valus of the pie chart of "Contact type"
contacttype = df.groupby('Contact type').size()
labels = contacttype.index.format()
values = contacttype.values.tolist()
explode = [0, 0, 0, 0, 0]
colors = ['#ffea65', '#ff5964', '#86cdc5', '#c9f477', '#9F8AA4']

# Generate and show the pie chart
plt.pie(values, labels = labels, autopct='%0.1f%%', explode=explode, colors=colors)
plt.show()
```

Listing A.3: Codes for Generating *Contact type* Pie Chart

```
# Set valus of the pie chart of 'Category'
callcategory = df.groupby('Category').size()
labels = callcategory.index.format()
values = callcategory.values.tolist()
a1, b1, c1, d1 = labels
labels = [a1, b1, d1, c1]
a2, b2, c2, d2 = values
values = [a2, b2, d2, c2]
explode = [0, 0, 0, 0]
colors = ['#ffea65','#ffea65', '#ff5964', '#86cdc5']

# Generate and show the pie chart
plt.pie(values, labels = labels, autopct='%0.1f%%', explode=explode, colors=colors)
plt.show()
```

Listing A.4: Codes for Generating *Category* Pie Chart

```
# Create a new DataFrame df2. Set "Created" as the index and transform created dates
    to period
df2 = df.set_index('Created')
df2.index = df2.index.to_period('M')
df2['Count'] = 1

# Create a pivot table for checking
```

```
df3 = pd.pivot_table(df2, index = 'Created', columns = 'Contact type', values = 'Count', aggfunc='sum')

email = df3['Email'].values.tolist()
chat = df3['Lync/Chat'].values.tolist()
phone = df3['Phone'].values.tolist()
self = df3['Self-service'].values.tolist()
walkin = df3['Walk-in'].values.tolist()
lst = np.array((email, chat, phone, self, walkin))

colors = ['#ffea65', '#ff5964', '#6b7bc9', '#68CDBE', '#8A7090']
x = np.arange(6)
dates = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']

plt.bar(x, lst[0], bottom=np.sum(lst[:0], axis=0), label='Email', color=colors[0])
plt.bar(x, lst[1], bottom=np.sum(lst[:1], axis=0), label='Lync/Chat', color=colors[1])
plt.bar(x, lst[2], bottom=np.sum(lst[:2], axis=0), label='Phone', color=colors[2])
plt.bar(x, lst[3], bottom=np.sum(lst[:3], axis=0), label='Self-service', color=colors[3])
plt.bar(x, lst[4], bottom=np.sum(lst[:4], axis=0), label='Walk-in', color=colors[4])
plt.xticks(x, dates)
plt.legend(loc=(1.05, 0.4))

plt.show()
```

Listing A.5: Codes for Generating *Contact type* Bar Chart

```
# Create a new DataFrame df2. Set "Created" as the index and transform created dates to period
df2 = df.set_index('Created')
df2.index = df2.index.to_period('M')
df2['Count'] = 1

df3 = pd.pivot_table(df2, index = 'Created', columns = 'Category', values = 'Count', aggfunc='sum')

complaint = df3['Complaint'].values.tolist()
incident = df3['Incident'].values.tolist()
question = df3['Question'].values.tolist()
rfc = df3['RFC'].values.tolist()
lst = np.array((complaint, incident, question, rfc))

colors = ['#E8D55C', '#ff5964', '#6b7bc9', '#68CDBE']
x = np.arange(6)
dates = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']

plt.bar(x, lst[0], bottom=np.sum(lst[:0], axis=0), label='Complaint', color=colors[0])
plt.bar(x, lst[1], bottom=np.sum(lst[:1], axis=0), label='Incident', color=colors[1])
plt.bar(x, lst[2], bottom=np.sum(lst[:2], axis=0), label='Question', color=colors[2])
plt.bar(x, lst[3], bottom=np.sum(lst[:3], axis=0), label='RFC', color=colors[3])
plt.xticks(x, dates)

plt.show()
```

Listing A.6: Codes for Generating *Category* Bar Chart

A

```
# Create a new DataFrame df2 showing "Created" number and "Resolved" number in each
    day
dates = df.groupby('Created date').count().index.tolist()
created = df.groupby('Created date')['Number'].count().values
resolved = df.groupby('Resolved date')['Number'].count().values[:181]
df2 = pd.DataFrame({'Created':created, 'Resolved':resolved}, index = dates)

# Set chart size
plt.figure(figsize=(16, 5))
# Set label size
plt.tick_params(labelsize=16)

plt.plot(df2.index, df2['Created'])
plt.plot(df2.index, df2['Resolved'])

plt.xlabel('Date', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.legend(ncol=2, fontsize=16)

plt.show()
```

Listing A.7: Codes for Generating *Created* and *Resolved* Counts Line Chart

```
# Create a new DataFrame df2 showing "Created" number and "Resolved" number in each
    day
dates = df.groupby('Created date').count().index.tolist()
created = df.groupby('Created date')['Number'].count().values
resolved = df.groupby('Resolved date')['Number'].count().values[:181]
df2 = pd.DataFrame({'Created':created, 'Resolved':resolved}, index = dates)

# Set chart size
plt.figure(figsize=(16, 5))
# Set lebel size
plt.tick_params(labelsize=16)

plt.plot(df2.index, df2['Created'], '.-')

plt.xlabel('Date', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.legend(fontsize=16)

plt.show()
```

Listing A.8: Codes for Generating *Created* Counts Line Chart

```
# Create a new DataFrame df2 keeping tickets from April 2019 only
df2 = df[df['Created month'] == '2019-04']

# Create a new DataFrame df3 presenting "Created" number and "Resolved" number in
    each day
dates = df2.groupby('Created date').count().index.tolist()
created = df2.groupby('Created date')['Number'].count().values
resolved = df2.groupby('Resolved date')['Number'].count().values[:30]
df3 = pd.DataFrame({'Created':created, 'Resolved':resolved}, index = dates)

```

```python
# Set chart size
plt.figure(figsize=(16, 5))
# Set lebel size
plt.tick_params(labelsize=16)

plt.plot(df3.index, df3['Created'], '.-')
plt.plot(df3.index, df3['Resolved'], '.-')

plt.xticks(df3.index[::7])

plt.xlabel('Date', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.legend(ncol=2, fontsize=12)

plt.show()
```

Listing A.9: Codes for Generating *Created* Counts for April Line Chart

```python
# Create a new column to determine if the create date is weekday
def weekday(row):
    if row['Created'].weekday()==5 or row['Created'].weekday()==6:
        weekday = 'Weekend'
    else:
        weekday = 'Workday'
    row['Weekday'] = weekday
    return row
df = df.apply(weekday, axis=1)
```

Listing A.10: Codes for Creating a New Column to Determine Weekday

```python
# Create a new DataFrame df2. Set "Weekday" as the index
df2 = df.set_index('Weekday')
df2['Count'] = 1

# Create a pivot table for checking
df3 = pd.pivot_table(df2, index = 'Contact type', columns = 'Weekday', values = 'Count', aggfunc='sum')

# Set valus for "Workday"
contacttype = df.groupby('Contact type').size()
labels = df3.index.format()
values = df3['Workday'].values.tolist()
explode = [0, 0, 0, 0, 0]
colors = ['#ffea65', '#ff5964', '#86cdc5', '#c9f477', '#9F8AA4']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors=colors)
plt.show()
```

Listing A.11: Codes for Generating *Contact type* workday Pie Chart

```python
# Create a new DataFrame df2. Set "Weekday" as the index
df2 = df.set_index('Weekday')
df2['Count'] = 1

```

A

```
# Create a pivot table for checking
df3 = pd.pivot_table(df2, index = 'Contact type', columns = 'Weekday', values = 'Count', aggfunc='sum')

# Set valus for "Weekend"
contacttype = df.groupby('Contact type').size()
labels = df3.index.format()
values = df3['Weekend'].values.tolist()
a1, b1, c1, d1, e1 = labels
labels = [a1, b1, c1, e1, d1]
a2, b2, c2, d2, e2 = values
values = [a2, b2, c2, e2, d2]
explode = [0, 0, 0, 0, 0]
colors = ['#ffea65', '#ff5964', '#86cdc5', '#9F8AA4', '#c9f477']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors=colors)
plt.show()
```

Listing A.12: Codes for Generating *Contact type* weekend Pie Chart

```
# Create new columns for checking time delta
df['Resolved-Created'] = df['Resolved'] - df['Created']
df['Created-Opened'] = df['Created'] - df['Opened']
```

Listing A.13: Codes for Creating New Columns for Checking Time Delta

```
# Create new columns for checking time delta
df['Resolved-Created'] = df['Resolved'] - df['Created']
df['Resolved-Opened'] = df['Resolved'] - df['Opened']
df['Created-Opened'] = df['Created'] - df['Opened']
```

Listing A.14: Codes for Creating New Columns for Checking Time Delta

```
# Create a new DataFrame df2
df2 = df.copy()
# Create a new column "Resolved-Created-days"
df2['Resolved-Created-days'] = df2['Resolved-Created'].astype('timedelta64[D]')+1

# Set values for bar plot
count = df2.groupby('Resolved-Created-days')['Number'].count().values.tolist()
timedeltaD = df2.groupby('Resolved-Created-days')['Number'].count().index.tolist()

# Set label size
plt.tick_params(labelsize=14)
plt.xlabel('Resolved-Created Days', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.bar(timedeltaD, count, color='#86cdc5')

plt.show()
```

Listing A.15: Codes for Generating *Resolved-Created Days* Bar Chart

```
# Create a new column "Resolved-Created-hours"
df2['Resolved-Created-hours'] = df2['Resolved-Created'].astype('timedelta64[h]')+1
```

```

# Set values for bar plot
count = df2.groupby('Resolved-Created-hours')['Number'].count().values.tolist()
timedeltaD = df2.groupby('Resolved-Created-hours')['Number'].count().index.tolist()

# Set label size
plt.tick_params(labelsize=14)
plt.xlabel('Resolved-Created Hours', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.bar(timedeltaD[:24], count[:24], color='#86cdc5')

plt.show()
```

Listing A.16: Codes for Generating *Resolved-Created 24 Hours* Bar Chart

```
# Create a new column "Resolved-Created-minutes"
df2['Resolved-Created-minutes'] = df2['Resolved-Created'].astype('timedelta64[m]')+1

# Set values for bar plot
count = df2.groupby('Resolved-Created-minutes')['Number'].count().values.tolist()
timedeltaD = df2.groupby('Resolved-Created-minutes')['Number'].count().index.tolist()

# Set label size
plt.tick_params(labelsize=14)
plt.xlabel('Resolved-Created Minutes', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.bar(timedeltaD[:60], count[:60], color='#86cdc5')

plt.show()
```

Listing A.17: Codes for Generating *Resolved-Created 60 Minutes* Bar Chart

```
# Create a new column "Resolved-Created-seconds"
df2['Resolved-Created-seconds'] = df2['Resolved-Created'].astype('timedelta64[s]')+1

# Set values for bar plot
count = df2.groupby('Resolved-Created-seconds')['Number'].count().values.tolist()
timedeltaD = df2.groupby('Resolved-Created-seconds')['Number'].count().index.tolist()

# Set label size
plt.tick_params(labelsize=14)
plt.xlabel('Resolved-Created Seconds', fontdict={'fontweight':'bold', 'fontsize':16})
plt.ylabel('Counts', fontdict={'fontweight':'bold', 'fontsize':16})

plt.bar(timedeltaD[:60], count[:60], color='#86cdc5')

plt.show()
```

Listing A.18: Codes for Generating *Resolved-Created 60 Seconds* Bar Chart

```
# Created a new column df3 showing if time delta is less than 1 day or more than 1
    day
def deltaRC(row):
    if row['Resolved-Created'] < pd.Timedelta('1D'):
        deltaRC = 'Less than 1 Day'
```

```
    else:
        deltaRC = 'More than 1 Day'
    return pd.Series({'deltaRC':deltaRC})
df3 = df.apply(deltaRC, axis=1)

# Set valus of the pie chart
contacttype = df3.groupby('deltaRC').size()
labels = contacttype.index.format()
values = contacttype.values.tolist()
explode = [0, 0]
colors = ['#86cdc5', '#ff5964']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors
    =colors)
plt.show()
```

Listing A.19: Codes for Generating *Resolved-Created Days* Pie Chart

```
# Created a new column showing if time delta is less than 1 minute or more than 1
    minute
def deltaRC(row):
    if row['Resolved-Created'] < pd.Timedelta('1m'):
        deltaRC = 'Less than 1 Minute'
    else:
        deltaRC = 'More than 1 Minute'
    return pd.Series({'deltaRC':deltaRC})
df3 = df.apply(deltaRC, axis=1)

# Set valus of the pie chart
contacttype = df3.groupby('deltaRC').size()
labels = contacttype.index.format()
values = contacttype.values.tolist()
explode = [0, 0]
colors = ['#c9f477', '#9F8AA4']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors
    =colors)
plt.show()
```

Listing A.20: Codes for Generating *Resolved-Created Minutes* Pie Chart

```
# Set valus of the pie chart
contacttype = df2[df2['Resolved-Created-minutes']<=1].groupby('Contact type').size()
labels = contacttype.index.format()
values = contacttype.values.tolist()
a1, b1, c1, d1, e1 = labels
labels = [a1, b1, e1, c1, d1]
a2, b2, c2, d2, e2 = values
values = [a2, b2, e2, c2, d2]
explode = [0, 0, 0, 0, 0]
colors = ['#ffea65', '#ff5964', '#9F8AA4', '#86cdc5', '#c9f477']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors
    =colors)
```

```
plt.show()
```

Listing A.21: Codes for Generating Resolved Less Than 1 Minute *Contact type* Pie Chart

```
# Set valus of the pie chart
contacttype = df2[df2['Resolved-Created-minutes']<=1].groupby('Category').size()
labels = contacttype.index.format()
values = contacttype.values.tolist()
a1, b1, c1, d1 = labels
labels = [a1, b1, d1, c1]
a2, b2, c2, d2 = values
values = [a2, b2, d2, c2]
explode = [0, 0, 0, 0]
colors = ['#ffea65','#ffea65', '#ff5964', '#86cdc5']

# Generate and show the pie chart
plt.pie(values, labels = labels, radius=1, autopct='%0.1f%%', explode=explode, colors
    =colors)
plt.show()
```

Listing A.22: Codes for Generating Resolved Less Than 1 Minute *Category* Pie Chart

B

NUMBERED AS-IS MODEL

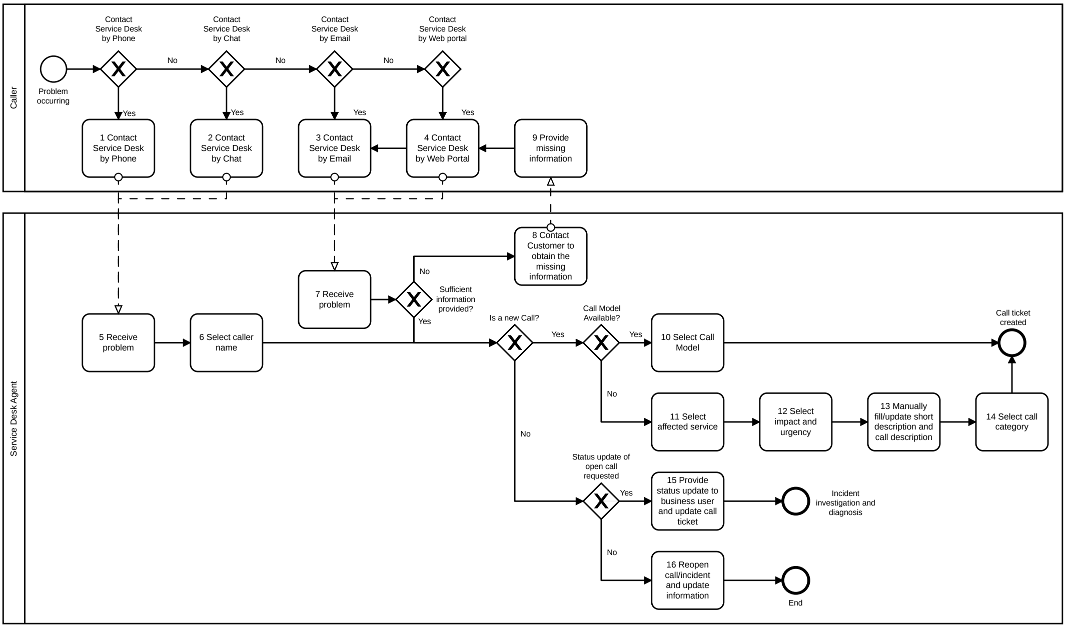


Figure B.1: BPMN of *Call Logging* in *Call Management* Process (Activities Numbered)

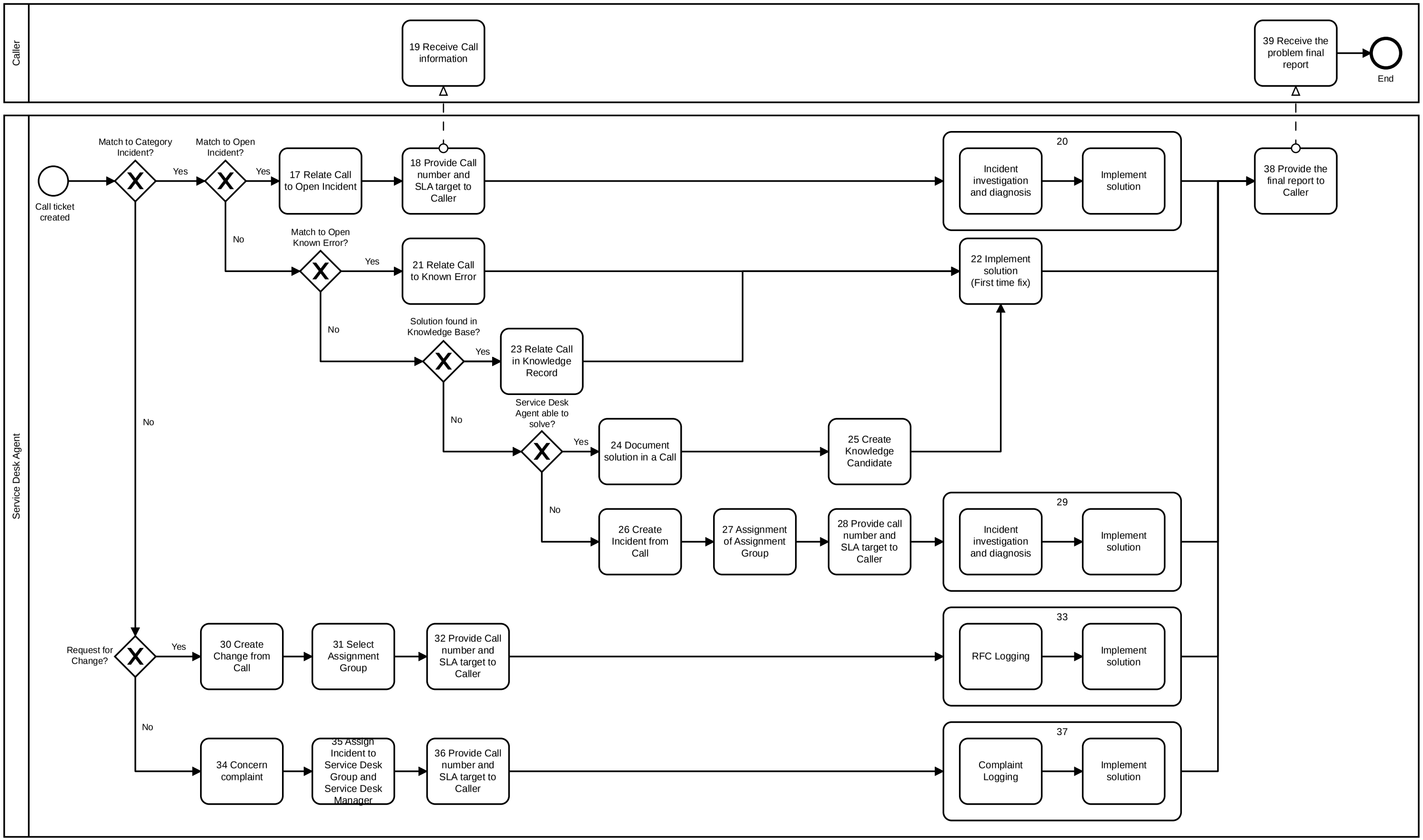


Figure B.2: BPMN of *Call Solving* in *Call Management* Process (Activities Numbered)

B