UCLouvain

LOUVAIN
School of Management

Louvain School of Management

In which specific situations and according to which criteria are BPMN, Coloured Petri Nets and UML Use Case each most appropriate for effectively modelling business processes?

Authors: Chloé Paternotte & Luci Six
Supervisor: Per Joakim Agrell
Academic year 2023-2024

Declaration Regarding AI Tool Usage in Master's Thesis

We recognize that AI tools might be valuable aids during the master's thesis work, but they are not infallible. Remember that transparency fosters trust, and acknowledging AI's role enhances the credibility of your work.

Therefore, when deciding to use such a tool, you need to adhere to the following principles of responsible use of AI.

1. **Critical Evaluation :**
    - We critically assessed the AI-generated output, ensuring its alignment with our research objectives.
    - Any modifications or corrections were made based on our expertise and domain knowledge.
2. **Transparency :**
    - We acknowledge the use of Phind, Quillbot, Perplexity, ChatGPT transparently, emphasizing that it contributed to our work but did not replace human judgment.
    - Our commitment to transparency ensures the integrity of this thesis.
3. **Ethical Considerations :**
    - We actively monitored for biases or unintended consequences introduced by the AI tool.
    - Our ethical responsibility guided our decisions throughout the research process.

Declaration **(This declaration is mandatory and must appear on the first page (after the title page) of the document.**

During the preparation of this master's thesis, the author(s) utilized Phind, Quillbot, Perplexity, ChatGPT for the following purpose:

1. Phind, Perplexity: speed up our search for documentation for the literary review so that we have more examples to illustrate the theory for our understanding

2. Quillbot: help us reformulate certain ideas

3. ChatGPT: help us understand certain technical concepts

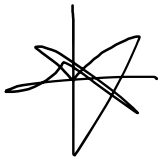
After using Phind, Quillbot, Perplexity, ChatGPT author(s) diligently reviewed and edited the content produced by the tool. We take full responsibility for the final content presented in this thesis.

By signing this declaration, we affirm that the content of this master's thesis reflects our original work, augmented by the responsible use of AI.

07/08/2024

Chloé Paternotte

Luci Six



Abstract

Business Process Model and Notation (BPMN) is widely used for business process modelling because of its intuitive graphical notation and its ability to bridge the gap between business users and technical developers. This Master's thesis presents a critical examination of BPMN and compares it with Use Case diagrams and Colored Petri Nets.

The study begins with an in-depth analysis of BPMN, highlighting its strengths as a standardised and comprehensive approach to business process visualisation. Next, Unified Modeling Language (UML) Use Case diagrams are examined for their effectiveness in capturing functional requirements and their simplicity in representing user interactions. Petri Nets are then analysed for their robustness in modelling processes and their mathematical rigour in validating processes.

The comparative analysis reveals that while BPMN excels at providing a detailed, standardised description of business processes, UML Use Case offers a simpler approach to capturing user requirements, and Colored Petri Nets are a powerful tool for modelling complex and concurrent systems. The results suggest that the choice of modelling technique should be guided by the complexity of the processes, the audience for the modelling and the level of detail required.

This thesis contributes to the ongoing discussion on business process modelling by providing a complete evaluation of BPMN, UML Use Case and Colored Petri Nets, offering insight into their respective advantages and limitations. The comparative analyses are intended to guide practitioners in choosing the most appropriate modelling technique for their specific business process needs.

ACKNOWLEDGMENTS

This thesis represents the end of our academic career at UCLouvain and we would like to express our gratitude to all those who have contributed to its completion.

First of all, we would like to thank our thesis promoter, Professor Per Joakim Agrell, for his availability and precious advice throughout this project. His knowledge of the field and his academic rigour were essential to the success of our project.

Then, we extend our thanks to the LSM Professors, Mr Kolp and Mr Vanderdonckt. Their expertise and passion for the field have enriched our understanding of modelling models.

We would also like to thank the consultants at Headmind Partners, who played an active role in the writing of this thesis, through interviews and a special thank you to Terence and Guillaume who validated some of the writings.

We send our thanks to our families and friends who guided and supported us during the writing of this dissertation, as well as for proofreading it.

Finally, we would like to express our gratitude to all those who, directly or indirectly, helped us to complete this work.

Glossary

BPMI	Business Process Management Initiative
BPM	Business Process Management
BPML	Business Process Modeling Language
BPMN	Business Process Modeling and Notation
CPN	Colored Petri Nets
DEDS	Discrete event dynamic systems
GSPN	Generalised stochastic Petri Net
HLPN	High level Petri Net
HPN	Hierarchical Petri Net
OCL	Object Constraint Language
OMG	Object Management Group
PN	Petri Net
RPA	Robotics Process Automation
SPN	Stochastic Petri Net
TPN	Timed Petri Net
UML	Unified Modeling Language
WFMS	Workflow Management System

Table of contents

List of figures

List of tables

List of appendices

1. Introduction

The introduction provides an overview of Business Process Management (BPM) and its historical evolution, highlighting the key milestones and developments that have shaped the discipline. It sets the framework for the research question and outlines the objective of the study. The introduction also discusses the limitations encountered in this study, as well as the methodology and outline used for this paper.

1.1. Business Process Management

Although Business Process[1] Management has become a mature discipline over the last few decades, but the roots of the notion reach much further back in history (van Der Aalst et al., 2016). In 1776, Adam Smith sought to improve business efficiency by proposing to divide work into sub-sections. With the same aim, and also to improve product quality, Frederick Taylor then introduced the "Time and Motion" method (Alotaibi, 2014). During the second industrial revolution, with the purpose of increasing production (van der Aalst et al., 2016), S. Williams used the term BPM in systems engineering. The third industrial revolution saw the advent of IT systems (van der Aalst et al., 2016), which led to a desire for process automation and thus techniques such as workflow technologies and transaction processing systems came into being (Alotaibi, 2014). The first Workflow Management Systems (WFM) had found their origins in the production processes of the manufacturing industry (van der Aalst et al., 2016). Total Quality Management then arrived in the 1980s followed by Business Process Reengineering in the 1990s (Alotaibi, 2014) accompanied by a rebound in advances related to workflow technology. This triggered, a few years later, a significant expansion in the number of commercial WFMSs[2] (Vaisman, 2013). It is then in the 21st century, where the focus is on creating intelligent manufacturing systems (van der Aalst et al., 2016), that Business Process Modelling is used to optimise, redesign and ensure the quality of operations, enabling companies to adapt quickly to changing customer needs. Business Process Modelling is therefore a useful tool that helps companies to achieve several objectives: explain the business process, taking into account that it has been designed to allow for both human and machine

[1]A business process is the combination of several internal operations with a framework outlining their dependencies and logical sequence with the aim of achieving a specific outcome (Aguilar-Savén, 2004).

[2] Systems such as Staffware, COSA and IBM MQ Series Workflow (Vaisman, 2013).

description; analyse the business process using quantitative or qualitative method and finally carry out simulations or support the execution of the business process (Alotaibi, 2014).

In an economic environment where companies are constantly looking for new ways to reduce costs, the focus is on improving the way things are done (Recker et al., 2009). This has led to the growing popularity of process orientation (Aguilar-Savén, 2004). An internship at Headmind Partners (for more information about Headmind Partners, see the website of the company : https://www.headmind.com/fr/), a digital consultancy, confirmed this observation and showed that process modelling is at the heart of certain assignments. After discussions with the consultants, it emerged that the only model they use for process modelling is BPMN, simply because it is the only one they are familiar with. It also appeared that many of them were applying BPMN in their own way, without respecting all the semantics of the language, due to a lack of training in BPMN. However, they have access to a wide choice of models, with which they might be more comfortable. Indeed, the growing interest in process orientation has led to an increase in the number of modelling methodologies and tools (Aguilar-Savén, 2004).

1.2. Research question

In the context of the above observations, we wanted to better understand how the BPMN language could be optimised and standardised to meet the needs of professionals, and what other methods might be more appropriate in certain contexts. After a search of the scientific literature and a discussion with two Professors from UCLouvain, Manuel Kolp and Jean Vanderdonckt (see Appendix A), it emerged that UML Use Case and Petri Nets models were the most appropriate to be compared in a digital consultancy context. The research question for this Master's thesis is therefore:

"*In which specific situations and according to which criteria are BPMN, Coloured Petri Nets and UML Use Case each most appropriate for effectively modelling business processes?"*

The aim of this paper is therefore to compare the three models and define the appropriate context for using each of them. Indeed, given the large number of process modelling methods available today, the choice of the appropriate technique can be complex. Hence the need to guide practitioners in this area (Aguilar-Savén, 2004).

1.3. Limitations

In order to carry out this study, some limits were set.

First of all, there are a multitude of modelling methods, and analysing them all would not have been relevant. Therefore, this paper focuses only on Petri Nets and UML Use Case, as these methods are the most frequently cited in the scientific literature in the context of business process modelling. Moreover, even among Petri Nets, there are many extensions to this model. We have chosen to focus on the Colored Petri Nets extension, one of the extensions most often mentioned in the scientific literature and the most relevant to the chosen practical case: the customer complaints management process.

Furthermore, in Chapter 3, the analysis chapter, we make a theoretical assessment of the modelling tools, but this criterion will not be used as a point of comparison between the models. Indeed, a comparison of the assessments of the different modelling tools would be outside the scope of this paper.

It should also be noted that the diagrams analysed in Chapter 4 are taken from the literature. It was not possible to obtain the diagrams modelled by the Headmind Partners consultants because of the sensitivity of the data they contained. Consequently, we selected a process commonly encountered by the consultants in their missions and for which diagrams were available in the scientific literature.

In the end, we decided to interview only consultants who had worked or were working at Headmind, the company where we did our internship during the last four months, for ease of reference, as most of them were available to us at the office.

1.4. Methodology

Firstly, as it is shown in Appendix B, we conducted several interviews with Headmind Partners consultants who had used process modelling in one or more of their assignments. In the course of these discussions, we were able to note that the current practice within this company is the BPMN model, we were able to assess their level of mastery of the models they employ and understand why they use some models in certain specific cases. These interviews enabled us to establish the scope of this study. During our interviews, we quickly realised that the consultants at Headmind Partners did not know of many modelling models comparable to BPMN, so we searched the scientific literature for the most studied and

relevant models to compare with BPMN. To confirm our choice (Petri Nets and UML Use Case) and to ensure a good understanding of model terminology, we spoke with two Professors at UCLouvain, Manuel Kolp and Jean Vanderdonckt. Next, an analytical and comparative method based on the scientific literature is used to answer the research question.

1.5. Outline

This master's thesis is structured as follows: after the introductory chapter, the second chapter reviews the literature in order to provide a good overview of the models analysed in Chapter 3, with more details on the BPMN model. For this analysis, the three models chosen will be evaluated according to the following criteria: semantics, scalability, expressiveness and precision, abstraction level, flexibility and extensibility, ease of learning and use, and available modelling tools. Chapter 4 then examines the application of the three models to a common business case, namely the customer complaints management process, by comparing the Petri Nets and Use Case models with the BPMN model. The fifth and final chapter concludes this Master's thesis by summarising the findings, explaining the contributions and the future prospects of this study before giving a critical opinion on the limitations set out in Section 1.3 and suggesting ways forward for future research.

2. Literature review

This section provides an overview of the key concepts, history and application areas of BPMN, as well as the two models used for the comparative analysis in this paper, namely : UML Use Case and Petri Nets.

2.1. Business Process Modeling and Notation

BPMN is a business process modelling standard that provides a graphical notation for specifying business processes. It aims to be intuitive for business users while being able to represent complex processes with fairly simple semantics. BPMN supports the modelling of business processes for all stakeholders, including business analysts, technical developers and business managers, enabling the boundary between the design and implementation of business processes to be reduced (Athuraliya, 2023; von Rosing et al., 2015).

The beginning of BPMN was marked by the creation of the Business Process Management Initiative (BPMI) in the 1990s. The aim was to establish a new standard for business process modelling by bringing together software suppliers and consumers (Adel, 2023). The Object Management Group[3] (OMG) and BPMI merged in 2003. As a new standard for business process modelling, they jointly developed BPMN. BPMN uses a visual representation of business processes that is simple to read and understand even for non-technical users (Adel, 2023; von Rosing et al., 2015).

After BPMI had merged into OMG in February 2006, BPMN 1.0 was accepted as an OMG standard (Muehlen, 2009; von Rosing et al., 2015). A major update to this standard was introduced with the release of BPMN 2.0 in January 2011. Significant improvements were made with this version, which included non-interrupting events, event subprocesses, choreographies and discussion diagrams (Adel, 2023; von Rosing et al., 2015). The number of elements increased from 55 to 116. The flexibility and power of the notation were increased by applying several of these new features to the modelling of interactions between things and/or processes (von Rosing et al., 2015).

[3] "The Object Management Group® is an international, open membership, not-for-profit technology standards consortium. Founded in 1989, OMG standards are driven by vendors, end-users, academic institutions and government agencies. OMG Task Forces develop enterprise integration standards for a wide range of technologies and an even wider range of industries.", for more information, see: https://www.omg.org/about/.

BPMN has been widely adopted and is supported by a broad range of software tools, facilitating the creation, sharing, and modification of business process models. BPMN's objective has always been to give business users an intuitive semantics for process modelling and the tools to convert models into actual processes. Due to these two goals, BPMN has emerged as the industry standard for business process modelling and has gone through multiple updates to improve its functionality and suit industry demands (Adel, 2023; Muehlen, 2009; von Rosing et al., 2015).

Finally, BPMN is a universal model that can be used in many sectors, here is a brief overview. Each has unique characteristics and requirements that influence the applicability of BPMN. Firstly, BPMN is very suitable for Business Process Management (BPM) as it provides a standardised method of visualising processes. It is particularly useful in sectors where process efficiency and optimisation are important for achieving strategic objectives. Next, BPMNs are useful in financial markets, as these are often complex processes involving the transfer of money, etc. This model can help with modelling, guaranteeing transparency and compliance with financial regulations. BPMNs are also used in manufacturing, logistics and retail companies facing challenges in supply chain management, production planning and inventory control. Optimising these processes, results in improved operational efficiency. Finally, these diagrams are useful for the public administration and telecommunications sectors. These fields deal with a wide range of processes, from citizen services to the management of public networks. (Frye, 2024; Lucidchart, n. d.; Microsoft 365 Team, 2019).

2.2. BPMN sub-models

In BPMN modelling there are four types of sub-models (Von Rosing et al., 2015), namely:

- business process diagrams
- choreography diagrams
- collaboration diagrams
- conversations diagrams

They all have their specificities and therefore different application cases. A detailed explanation of each model would be out of scope of this paper, so a brief presentation will be made for all four but the complete analysis will focus only on one model.

The first type of diagram, and the one this paper will focus on, is the business process diagram. Some of these diagrams are called private while others are called public. The first one models private business processes, i.e. those internal to a specific organisation as it is shown in Appendix C. These processes have usually been called workflows, BPM processes or service orchestration. There are two types of private processes, executable and non-executable[4] (Von Rosing et al., 2015). Regarding public process, it represents interactions with another process or participant as it is illustrated by the appendix D. Only activities and events used to communicate with other participants are included in this process. As a result, the public process does not display the internal activities of a private business process (Von Rosing et al., 2015).

The second type of BPMN diagram is the choreography. These diagrams are intended to allow good interaction between cooperating organisations so that they meet specific objectives. To do this, these diagrams define the global constraints and specifications of a system in order to represent the interactions between its components while making abstractions of their internal structure (Corradini et al., 2020). A Choreography diagram looks like a private business process but is distinguished by activities representing the exchange of messages between several participants as it is shown in Appendix E. Furthermore, a choreography has no central controller, responsible entity or observer (Von Rosing et al., 2015).

Regarding collaboration diagrams, they describe the internal behaviour of components belonging to a distributed system. These diagrams illustrate how several components interact while providing details about internal actions (Corradini et al., 2020). These component interactions can be represented by public and/or private processes that communicate with each other (Von Rosing et al., 2015). The concept of collaboration is closely linked to that of choreography because it is crucial to maintain a conformity between the two in order to ensure a good collaboration between the components while respecting the communication constraints defined globally (Corradini et al., 2020). In a collaboration, any possible combinations of pools, processes and choreography are allowed (Von Rosing et al., 2015). An example of a collaboration diagram is depicted in Appendix F.

The last sub-model is the conversation diagram, which is derived from the collaboration diagram by taking an informal description of it. In this conversation diagram, no processes are

[4]An executable process is one that has been modelled to be executed in accordance with the stated BPMN execution semantics whereas a non-executable process is a private process that has been modelled for documenting process behaviour at a level of detail specified by the modeller (Von Rosing et al., 2015).

included in the pools but there are linked conversations between them to represent a separate business scenario. Each conversation, within the diagram, reflects a logical relationship of message exchange as represented in Appendix G (von Rosing et al., 2015). The conversation diagrams have semantic elements specific to them and these are included in Appendix H.

2.3. UML Use Case

The Use Case model, associated with the UML, has its roots before the creation of the UML. Its history began in 1986, when Ivar Jacobson, a pioneer in the field of software engineering, first formulated textual and visual modelling techniques for specifying what he called "use case". These techniques were intended to capture the functional requirements of software systems in a clear and structured way. In 1992, the publication of the book co-authored by Jacobson, "Object-Oriented Software Engineering - A Use Case Driven Approach", marked a turning point. This book helped popularise the use of Use Case to capture functional requirements and also laid the foundations for what would later become the UML (Lucidchart, n. d.).

With the formalisation of the UML in the 1990s by the OMG, Use Case was integrated as one of the central elements of this modelling language. The UML itself was designed to avoid the complexity and confusion of representing all objects, relationships and processes in a single diagram. To this end, the UML uses 14 different types of diagrams[5], divided into three main categories: structural diagrams, behavioural diagrams and interaction diagrams. Use Case is a subcategory of behavioural diagrams (IONOS Digital Guide, 2020).

Structural and behavioural diagrams play a crucial role in modelling software systems. Structural diagrams focus on representing all the elements of a system and the relationships between them. Behavioural diagrams, on the other hand, do not focus on static structures. Instead, they show the expected or actual flow of processes that can be expected when using the program or software. There are dynamic diagrams that represent behaviour in terms of process flows, interactions and state changes (IONOS Digital Guide, 2020).

Then, Use Case diagrams are designed to illustrate the interactions between users - known as actors - and the system. They highlight the functionality of the system in a way that is visible and useful to users. They play a crucial role in understanding the functional requirements of a

[5]for more information on different types of UML diagrams, see: https://creately.com/blog/diagrams/uml-diagram-types-examples/

system by representing the different ways in which stakeholders interact with it to achieve specific goals. A Use Case diagram captures the functionality of the system by integrating use cases, actors and their relationships. It models the tasks, services and functions required by a system or application subsystem. This representation helps to understand the functional requirements of the system from the user's point of view, facilitating communication between stakeholders and guiding the development process (UML Use Case Diagram - JavatPoint, n. d.).

Finally, there are many suitable domains for UML Use Case diagrams, particularly those involving interactions between users and the system. These diagrams are particularly useful for capturing the requirements of a system from the point of view of its users, helping stakeholders to visualise complex processes and interactions in a simple and intuitive way. Firstly, at the level of software applications, Use Case is essential, whether it's a customer relationship management system or an analytical platform (Wolfram, 2022). They make it easier to understand the flow of actions between the user and the system, which is essential for designing intuitive user interfaces. The same applies to systems integration, with Use Case helping to identify potential integration points, data flows and dependencies between systems (IONOS Digital Guide, 2020). Use Case is also used in e-commerce platforms, healthcare and education systems (Bacastow, n. d.; The Agile Modeling (AM) Method, 2023). For e-commerce platforms, these diagrams describe the entire shopping experience, ensuring that all user interactions are taken into account in the development of the platform. In the healthcare sector, Use Case is used to model processes such as patient registration, appointment scheduling and treatment planning. Finally, in education systems, diagrams are used to model the processes involved in registering students, recording courses and marking reports (The Agile Modeling (AM) Method, 2023). These diagrams therefore help to design optimised systems that improve both administrative efficiency and the customer experience (students, customers, purchasers, ...).

2.4. Petri Nets

Petri Nets are a well-known graphical tool for process analysis and modelling, invented by Carl Adam Petri. It all started in August 1939, when Petri illustrated how different elements would react to form a final molecule after undergoing some intermediate compounds. This idea evolved into his doctoral thesis “Communication with Automata,” at the Technical University of Darmstadt, Germany, in 1962, where he codified the net-like representations (Zhou & Wu, 2010).

A classical Petri Net (PN) is a directed bipartite[6] and weighted graph with two types of nodes: places and transitions connected by arcs. This structure makes Petri nets a powerful tool for modelling processes, allowing for both formal methodology with mathematical representation and schematic modelling techniques. This combination is instrumental in analysing workflows and identifying potential flaws early in the design process, enhancing reliability (Murata, 1989).

Over time, the classical PN has shown certain limitations, leading to the development of several extensions (Medina-Garcia et al., 2023). These new nets are referred to as high level Petri Nets (HLPN) and allow for the representation and study of complex business processes. Some of the most popular and widely used PN extensions are:

- Colored Petri Nets (CPNs)
- Timed Petri Nets (TPNs)
- Stochastic Petri Nets (SPNs)
- Hierarchical Petri Nets (HPNs)

These kinds of extensions are essential for using PNs to model intricate and sizable business processes. The HLPN carries over all of the benefits of the classical PN, including its exact and graphical structure, solid mathematical underpinnings, and plethora of analysis techniques (Van der Aalst & Hee, 1996).

As explained in Chapter 1, this paper focuses on CPN which is a graphical language used to create models of concurrent systems and analyse their characteristics. It is a discrete-event modelling language that combines high-level programming language functionality with that of Petri Nets (Jensen & Kristensen, 2009). Kurt Jensen's doctoral thesis, as published in 1979, served as the inspiration for the first iteration of CPNs, which have been developed by the CPN group at Aarhus University in Denmark since 1979 (Zhou & Wu, 2010). CPNs have been developed with the aim of creating an industrial strength modelling language, robustly theoretically supported and adaptable enough to be applied in practice to systems with the scale and complexity of normal industrial projects. Indeed, classical PNs offer the fundamental building blocks and graphical notation for modelling synchronisation, concurrency, and communication (Jensen & Kristensen, 2009). But the primitives for defining data types, specifying data manipulation, and building compact and parameterizable models are

[6] A bipartite graph has two disjoint sets of vertices, namely, the top and bottom vertices (Guillaume & Latapy, 2006). In Petri nets, the two disjoint sets of nodes are places and transitions.

provided by the CPN ML programming language, which is built on the foundation of the functional programming language Standard ML. CPN permits the usage of tokens that, unlike low-level PN tokens, which are typically depicted as black, “uncolored” dots, carry data values and can therefore be recognized from one another. Initially, only limited, ad hoc sets of colours were employed, such as when listing a predetermined set of procedures. The theory and methods could be generalised to allow for the use of arbitrarily complex data types as colour sets, as was later realised. Nowadays, it is not unusual at all to have tokens that represent complicated data values, such as a list of thousands of entries with numerous types of fields. The distinction between a type and a colour set, as well as between a token colour and a token value, is no longer present (Jensen, 1997).

CPNs, as well as being used to model and specify the behaviour of concurrent systems, have the characteristic of being a model that is both state-oriented and action-oriented. It explains the system's states as well as the events that may cause those states to change. The behaviour of the system can be analysed by running simulations of the CPN model, often with the aim of debugging and examining the design of the system. Simulations can be run interactively or automatically. During an interactive simulation, which can be associated with single-step debugging, the modeller chooses from the events activated in the current state to determine the next step. The graphical representation of the CPN model enables the effects of the various steps to be observed directly. CPNs can also incorporate a concept of time to capture the duration of events in the system. Therefore, CPNs can be used for simulating and validating real-time systems as well as for simulation-based performance analysis, which looks at performance measures such system delays, throughput, and queue lengths (Jensen & Kristensen, 2009).

According to Murata, Petri Nets apply to the following areas : modelling and analysis of distributed software systems, distributed database systems, concurrent and parallel programs, flexible manufacturing/industrial control systems, discrete event systems, multi processor memory, dataflow computer systems, fault tolerant systems, programmable logic, asynchronous circuits and structures, compilers and operating systems, office information systems, formal languages, and logic programs. The literature also examines human factors, digital filters, legal systems, neural networks, local area networks, and decision models as other interesting applications (Medina-Garcia et al., 2023). Therefore, PNs are used extensively in various fields, including economics, mechanics, computer science, logic, organisation, biology, and telecommunication protocols, thanks to their flexibility and versatility (Valavanis,

1990). The graphical depiction provided by PNs makes it possible to visualise dynamic behaviour and model concurrent systems, which is common in complex business processes involving multiple organisational departments (Medina-Garcia et al., 2023). PNs serve also as a potent communication tool for both practitioners and theorists. Theoreticians can guide practitioners in making their models more methodical, while practitioners can offer insights into making models more realistic. This two-way communication strengthens the practical application of Petri Nets in various domains (Murata, 1989).

3. Analysis

This chapter analyses the three models according to 7 criteria: the completeness of their semantics, their scalability, their expressiveness and precision, their level of abstraction, their flexibility and extensibility, the ease with which they can be learned and used, and finally their implementation. The three models will then be compared against each other according to these criteria. Finally, this chapter gives an overview of the criteria to be taken into account when choosing a model, depending on the modelling context: the modelling objective, the type of process to be modelled, the model's target audience, the cost-benefit ratio and the time available for modelling.

3.1. BPMN Overview

3.1.1. Semantic completeness

One of the main objectives of BPMN's creation was to provide a clear, easy-to-understand language for building Business Process models while also offering the semantics and underlying systems to manage the business processes' inherent complexity. The method used to address these two contradictory needs was to put the notation's graphical elements into distinct categories. The BPMN language[7] can be deconstructed and put into a layered model (Von Rosing et al., 2015).

The first layer is the flow objects, representing the main building blocks for any process diagram. There are three flow objects : event, activity and gateway. As illustrated in Appendix I, events are represented as circles with open centres so that internal markers can distinguish between various causes or outcomes. Based on when they impact the flow, events can be classified as start, intermediate, or end (White, 2004). Indeed, the event serves as both the catalyst for the start of a particular action and its representation as its outcome. An event is in fact the symbol of a potential outcome that may occur during the operational process. A start event initiates the business flow, while an end event ends it. Intermediate events might happen at any point during the flow (Ciaramella et al., 2009). Regarding activities, they represent the tasks that an organisation completes inside a process. An activity may be nonatomic (subprocess) or atomic (White, 2004). Tasks and processes are two examples of atomic and

[7] For an explanation of the full BPMN semantics, see the normative specification on BPMN 2.0.2 of the OMG : https://www.google.com/url?q=https://www.omg.org/spec/BPMN/2.0.2/PDF&sa=D&source=docs&ust=1716326897662338&usg=AOvVaw2rU4tvQWzTAw5xOZKqDBdf

compound business activities (Ciaramella et al., 2009). A process model's sub-process and task, which are rounded rectangle-shaped activities, as Appendix J shows, are examples of this type of activity (White, 2004). Both standard processes and choreographies involve activities. The last flow object is the gateways. In a process and in choreography, a gateway is utilised to regulate the divergence and convergence of sequence flows. As a result, it will decide when to branch, fork, merge, and join paths. The kind of behaviour control will be indicated by internal markers as it is shown in Appendix K (White, 2004) . The gateway task is responsible for guiding a process's evolution by determining the next action to take after assessing a condition. Alternatively or concurrently when the process evolution does not support parallel activities, the gateway splits the evolution into two parallel branches that need to be linked with the same element. Therefore, a gateway is a decision point used to regulate the flow of business activities (Carchiolo et al., 2019). They are symbolised by diamonds with a sign denoting potential flow exits in the centre (White, 2004).

To create the basic structure of a business process, the flow objects are linked together in a diagram using connecting objects, representing the second layer (White, 2004). There are four types of connectors: sequence flow, which, in a process or choreography diagram, illustrates the sequence of activities; message flow, which depicts the communication between two people in separate processes; association, which links flow objects to text and other elements and data association, which associates data with flow objects, showing the inputs and outputs of activities (Von Rosing et al., 2015). To see the depiction resolution for all the connecting objects, see Appendix L.

Then, pools and lanes represent the third layer. A pool is a process participant's representation. A participant may take the form of a more generic business position, such as a customer, seller, or manufacturer, or they may be a specific business entity, such as a company. As illustrated in Appendix M, pools are used as a container for all flow objects in the graphical representation. When modelling business-to-business scenarios, a pool can divide a process from other pools, even while a pool does not have to include any internal details, it can simply be a “black box” [8](Mpardis & Kotsilieris, 2010). A lane runs the length of the pool, vertically or horizontally, as shown in Appendix N, to represent a sub-division within

[8]A black box is an empty pool (Von Rosing et al., 2015).

the pool and they are used to represent the resource that is responsible for the execution of tasks in the process (White, 2004)[9].

Artefacts belong to the fourth layer which represents additional elements that can be used to clarify some of the elements in the diagram. Diagrams can have any number of artefacts added to them, depending on what makes sense given the context of the business processes they are modelling. For a graphical representation, see Appendix O. The last layer represents data objects showing how data is required or produced by activities, groups used for documentation or analysis and annotations to give the reader of a BPMN diagram more textual information (White, 2004). Appendix P shows the representation and description of data objects. In other words, the flow objects must be connected using the connectors. All those elements can be grouped in a pool to represent a process. Finally, we have the artefacts to add additional information to the model.

To conclude, BPMN has a very complete semantic, with about a hundred symbols. It enables all the nuances of a process to be modelled, including its components (IT, human resources, etc.), the temporal dimension, decisions, internal and external relations to the process, etc. All the elements are represented visually, distinctly and intuitively, providing a quick and unambiguous understanding of the process.

3.1.2. Scalability

From straightforward workflows to extremely intricate business processes, BPMN is scalable[10] and suitable for modelling processes of different complexity. It fulfils the requirements of both large and small enterprises, mainly through the use of sub-processes. With BPMN, subprocesses can be reused in other diagrams, and it is possible to break down large processes into smaller sub-processes to increase readability. When a part of the process needs more information, integrated sub-processes also assist in keeping the main diagram from becoming cluttered (Visual Paradigm, 2023). BPMN's ability to model processes in parallel, using gateways or parallel activities, also makes it easier to manage large-scale processes by splitting them. Besides that, Schäffer et al. (2021) emphasised the Process-Driven Approach, which uses BPMN to integrate and orchestrate different software tools and applications in

[9]For more information on the pools and lanes mechanism, see the work of White (2004) : 07-04_WP_Intro_to_BPMN_-_White-2.pdf (free.fr)

[10]The scalability of a business process model demonstrates a system's capacity to manage the volume of given business process activities. Scalability is a term used in business process management to describe how large a business process model is in comparison to other business process models (Yaqin et al., 2017).

order to improve scalability. Business operations may scale effectively because of this modular architecture, which reuses components and dynamically adjusts to changing circumstances (Schäffer et al., 2021).

3.1.3. Expressiveness and precision

BPMN's accuracy and expressiveness are key aspects contributing to its effectiveness in BPM. One of BPMN's main objectives is to ensure that models are comprehensible to end-users and analysts who need to communicate and use business processes in a standardised way (Zou and Pavlovski, 2008). Consequently, it is crucial that diagrams are clear to all stakeholders (Aldin & de Cesare, 2009). To achieve this, BPMN has a global standard that guarantees uniformity, thereby improving communication and understanding between the various process participants. BPMN incorporates a wide range of symbols and elements for accurately describing complex processes, ensuring detailed capture of every aspect of the process (Creately, 2023). By providing a common modelling language that can be understood by all stakeholders, BPMN encourages participation and facilitates faster agreement on current and future processes through unambiguous models. Notation facilitates operations analysis, enabling organisations to quickly agree on processes and create process flow libraries for training new employees (Education, 2022).

Secondly, BPMN is particularly expressive, capable of modelling complex process dynamics such as multiple paths, conditional logic and simultaneous activities. The notation is versatile, ensuring that BPMN serves as an effective communication tool between different roles and levels within an organisation (Creately, 2023). Compared with other process modelling languages, BPMN supports a wider range of workflow models, such as synchronisation, exclusive choice, simple merge, deferred choice, cancel activity, and so on. This includes more control flow structures, enhancing BPMN's expressiveness in describing process flow (Stein , 2010).

Finally, the standardised notation of BPMN, its vast set of symbols and its ability to manage complex processes give it a very high degree of precision. However, on the other hand, this very complete nomenclature has the effect of making the expressiveness of processes more complex. For the same process, there are different ways of modelling it, and the semantics are less formal and more extensive.

Another limiting aspect of this model is that it is not possible to add a notion that allows the dynamic state of the process to be represented. For example, in a standard BPMN model, process steps such as order receipt, stock check, preparation, order shipment and customer notification are represented sequentially with tasks and events. However, this model does not easily capture intermediate dynamic states, such as an order “awaiting stock check”, “being prepared” or “awaiting shipment”.

3.1.4. Abstraction level

A vast array of unique graphical symbols found in the BPMN semantics can be used to create process models at various levels of abstraction, from high-level value chain descriptions to intricate transactional processes intended for integration into applications (Recker et al., 2009). On the one hand, for modellers who want or need, for example, a low degree of precision to construct process models for communication and documentation needs, the fundamental components and connectors will facilitate the creation of understandable diagrams. On the other hand, for modellers who need a higher level of detail, more information can be added to the basic components and represented by internal markers in order to develop process models that will either be the subject of in-depth research or be controlled by a business process management system, for instance. A start event with a message element added could serve as a practical illustration. This provides indeed more information about the process's start by changing its meaning from a simple start to a message-triggered start. This flexibility allows process models to be modified to meet particular needs or circumstances (Eby, n. d. ; White, 2004). It should be noted, however, that this large number of precise semantics is rarely used in practice. An analysis of BPMN modelling in practice by Muehlen and Recker (2009) showed that less than ten of these symbols are typically shown in a BPMN diagram, and their frequency of use has a long-tailed distribution that is comparable to the use of words in natural language (Recker et al., 2009). In addition to these comprehensive semantics, which can be adapted to different levels of abstraction, an important feature of BPMN 2.0 is sub-processes and multi-instances. The sub-process is a component of the entire process model, which is typically called by a complex or call action and described independently. It is possible to deconstruct a subprocess into even more depth. The term “multi instance” refers to the fact that several instances react to an activity and the execution can be sequential or parallel. This refers to the necessity of either sequential or simultaneous execution of an activity or subprocess (Wang et al., 2015).

In summary, BPMN can provide a level of abstraction from moderate to high depending on the process to be modelled and the semantics used.

3.1.5. Flexibility and extensibility

In the case of the BPMN method, the characteristic of extensibility is closely linked to that of the level of abstraction. Indeed, BPMN is a methodical approach to model an organisation's processes that allows for the representation of extended models and the flexible modification or enhancement of processes without compromising the original model. BPMN, which was created to give modellers flexibility, allows them to add context unique to specific scenarios, such those in vertical sectors (banking, insurance, etc.), and to expand the core notation. Depending on the context of the business processes being simulated, an infinite number of artefacts may be included (Aldin & de Cesare, 2009; White, 2004). It increases BPMN's functionality without necessitating any unique specifications. Because there will not be a specification outlining how new flow elements will be related to existing flow elements, it is crucial when extending BPMN to avoid changing the overall layout of existing flow elements (like events, activities, and gateways) or adding new ones. Therefore, it is possible to expand BPMN's expressive capabilities without compromising the fundamental order, message flow, or compatibility with runtime languages like BPEL. Modelling authorisation constraints is a concrete example of the extensibility of BPMN semantics[11] (Wolter & Schaad, 2007).

Even while BPMN is generally intended for structured processes, components like events, gateways, and activities allow it to be applied to unstructured or semi-structured processes as well. This feature increases the model's versatility in handling the intricacies of the real world by enabling the inclusion of human interaction and decision points (Camunda, n. d.).

Moreover, being an open source standard that is managed by the OMG, BPMN is always changing and is available for anybody to modify or expand. This openness promotes innovation and the creation of new approaches and tools for BPMN, thus increasing its extensibility (Eby, n. d.). Numerous BPMN modelling tools provide a wide range of customisation and extension options for BPMN diagrams. Because of this interaction with software tools, BPMN's extensibility and flexibility may be used more practically, enabling

[11] For a detailed example of modelling authorisation constraints, see: Wolter, C., & Schaad, A. (2007). Modeling of Task-Based Authorization Constraints in BPMN. Dans Springer eBooks (p. 64‑79). https://doi.org/10.1007/978-3-540-75183-0_5

businesses to customise their process models to meet specific business requirements (Camunda, n. d.; Eby, n. d.).

Finally, BPMN is highly extensible and flexible thanks to its very precise semantics, which allow us to extend the diagram by adding elements and annotations. However, as there are no mathematical foundations in this model, these extensions can introduce ambiguities. For example, the way in which a new element should interact with existing elements is not always formally defined, which can lead to multiple and inconsistent interpretations.

3.1.6. Ease of learning and use

On the one hand, thanks to the standardisation of BPMN semantics, the learning process is simplified by providing a consistent approach to process modelling. What's more, the visual nature of the diagrams is fairly straightforward, making them easy to understand for people from different backgrounds. This clear visual aspect is particularly beneficial for non-technical stakeholders unfamiliar with programming languages (Mathenge, n. d.).

On the other hand, in terms of usability, BPMN was developed with one aim of being easy to use and easy to understand. It offers a wide range of flow types, making it well-defined and therefore easy to use for inexperienced stakeholders. Despite the fact that BPMN is a complex model due to some of its specialised notations, it is not necessary to know all these specialised notations to create a useful BPMN diagram (Aldin & de Cesare, 2009). Indeed, having interviewed several Headmind Partners consultants (see Appendix B) who work with BPMN, we can confirm that the majority of them do not use all the symbols available, as their primary objective is to be comprehensible to as many people as possible, even those unfamiliar with BPMN. They therefore generally restrict themselves to the simplest symbols, even if more sophisticated symbols are sometimes more appropriate. In this way, novices and experts alike can get by without having to look up the nomenclature of each symbol. Obviously, the level of simplification of their BPMN depends on the audience it is aimed at: the more people who read it, the more simplified the diagram will be. Based on these interviews, we also analysed the reason for using BPMNs in particular. Most of the time, consultants decide to use BPMN because they are familiar with it, because it is the easiest way to model processes, or because the customer is used to working with BPMN. In the majority of cases, the consultants had to model processes in order to standardise them, so that they could be understood by everyone in

the company, so that new employees could be familiar with the processes, so that the processes could be implemented in new software, etc.

To maximise the effectiveness of BPMN, it is advisable to follow certain best practices. For example, draw diagrams from left to right, use verb-object “names” for activities, or ensure that each process begins with a start event and ends with an end event. It is also advisable to avoid crossing lines so as not to impair the diagram's legibility. These few best practices make it easier to follow the process flow, improve readability and maintain the clarity and relevance of the process (Mathenge, n. d.).

We then asked the consultants if they would like to introduce BPMN training in the companies where they work, their responses were all positive. Indeed, if it adds value to the company, investment in BPMN training is crucial for organisations wishing to take advantage of this notation effectively. According to the consultants, training on BPMN modelling tools should also be included. Organisations that invest in BPMN training can expect better process documentation, improved collaboration and a more solid basis for process optimisation and automation.

To sum up, learning and using BPMN is relatively easy if you concentrate on the basic elements of its semantics. However, if you go into more detail and want to know and/or use all the existing objects, it becomes more complicated because of their large number and the sometimes narrow nuances between certain objects.

3.1.7. Implementation

BPMN modelling tools facilitate the creation, visualisation and analysis of business processes by offering a variety of features. Most modelling tools are compliant with the BPMN 2.0 specification, enabling compatibility across different platforms. Tools generally provide rich graphical interfaces for creating and editing BPMN diagrams. In addition, many tools offer collaboration capabilities, enabling teams to work together on process models in real time. Another interesting feature that some tools offer is a simulation engine that enables the behaviour of a process to be tested under different conditions. This helps to identify bottlenecks, optimise workflows and ensure process efficiency (Ivanov et al., n. d; Wicks, 2023)

There are a multitude of modelling tools available for BPMN that can be used to create, manage and optimise business processes. Here are just a few of them: Camunda, Aris, Merise, Intalio designer, Lucid Chart, SmartDraw, EdrawMax Online,... (Edrawsoft, n. d.)

3.2. UML Use Case

3.2.1. Semantic completeness

The semantics of Use Case include actors, use cases, system boundaries and relationships. These elements are shown in Appendix Q.

First of all, the actors in a Use Case diagram represent external entities that interact with the system being modelled. These entities can be represented in a variety of ways, including users, other software systems or hardware devices. The role of actors is essential, as they initiate use cases within the system and receive the results of these interactions. Each actor is identified by a name and plays a specific role in the business or system environment. Actors are similar to users, but they can play several roles within the system. For example, a teacher may act as both an instructor and a researcher, fulfilling different roles with distinct interactions with the system. One of the main functions of actors is to trigger the execution of use cases within the system. They have responsibilities to the system, providing inputs or stimuli that trigger specific functionality. At the same time, actors have expectations of the system, anticipating certain outcomes or responses to their interactions (GeeksforGeeks, 2024).

Secondly, use cases in UML Use Case represent the specific functionality or processes that a system offers its users to achieve defined goals. Each use case is like a scene in a play, describing a specific action or interaction that the system can perform. For example, in an e-commerce system, use cases might be “Place an order” or “Track delivery”. Each use case is represented by an oval shape and is named by a verb followed by a noun or a noun phrase. Each actor must be linked to at least one use case, but some use cases may not be linked to specific actors. By analysing each use case, we can understand its purpose, the actors involved, and the steps needed to accomplish that functionality, which allows us to effectively capture the functional requirements of the system (Bacastow, n. d.; GeeksforGeeks, 2024).

Thirdly, the system boundary in a diagram allows a clear distinction to be made between the internal components of the system and the external actors or entities interacting with the system. Typically represented by a rectangular box surrounding all the system's Use Case, the system boundary performs several functions: it defines the scope of the system, indicating

which components are internal and which are external. This makes it possible to focus on the system's essential functionality without including unnecessary details about external entities. In a complex system, each module may constitute a system boundary, enabling the system to focus on specific business functions. For example, in an Enterprise resource planning system, the personnel, payroll and accounting modules may each form a separate system boundary, focusing on specific business functions (Bacastow, n. d.; GeeksforGeeks, 2024).

In Use Case diagrams, relationships play a crucial role in capturing the interactions and dependencies between different elements such as actors and use cases. They provide an overview of the connections in the modelled system, facilitating communication and design decisions. A fundamental relationship is the Association Relationship, which represents communication between an actor and a use case. This relationship, which is represented by a simple line between the use case and the actor, underlines the actor's involvement in the functionalities described by the Use Case. In Appendix R, for instance, within an online banking system, the “Customer” actor is associated with the “Funds transfer” use case, showing that the customer participates in the process of transferring funds.. Another important relationship is the Include Relationship, which shows that a use case includes the functionality of another use case. This approach allows one use case to integrate the functionality of another. For example in Appendix S, in a scenario for publishing on social networks, the “Compose a message” use case includes the “Add an image” functionality, which means that composing a message also involves adding an image. The third Use Case relationship is the Extend Relationship. It indicates that, under certain conditions, a use case can be extended by another use case. As shown in Appendix T, this relationship is represented by a dotted arrow with the keyword “extend” at the end pointing to the base use case. For instance, as explained in the article of GeeksforGeeks, if you want to book a flight, the “Choose a seat” use case can extend the “Book a flight” use case when the user wants to select a specific seat. Even though this step is optional, this relationship is useful for handling non-mandatory behaviours. Finally, the last Use Case relationship is the Generalization Relationship. This creates an “is-one” connection between two use cases. In other words, it indicates that one use case is an improved version of another, like a parent-child relationship. As in Appendix U, an arrow pointing from the specialised use case to the generic use case represents this relationship. For example, in a car rental system, the use case “Rent a Car” and “Rent a Bike” are specialised versions of the general use case “Rent a vehicle” (Bacastow, n. d.; IONOS Digital Guide, 2020; GeeksforGeeks, 2024; Visual Paradigm, n. d.).

To sum up, the semantics of Use Case are fairly basic and do not include any technical specificities. It is therefore fairly complicated to model all the components of a complex process, such as a notion of time, a decision, a necessary resource or the way in which a task is executed. So that the reader fully understands the nuances of the different stages, it is important to complete the Use Case by writing additional information.

3.2.2. Scalability

By identifying actors and their tasks within the system, Use Case helps set boundaries and determine which features should be included in the system (Alam, 2024; GeeksforGeeks, 2024).

These diagrams provide a high-level overview of system functionality. This makes it possible to add actors and use cases as the system develops, while avoiding going into too much detail. As systems grow, use cases can be broken down into smaller elements, called sub-systems, to make them easier to manage. This modular approach facilitates scalability by breaking down complex systems, each addressing a different aspect of the system (IBM, 2023).

Secondly, Use Case is flexible and can be easily modified to reflect changes in system architecture, thanks to the different relationship styles. This adaptability allows the diagram to evolve along with the system without becoming too complex (IONOS Digital Guide, 2020).

Use Case is therefore designed to adapt to the growth of a system. Their high level, modularity, flexibility and integration capabilities make them a powerful tool for managing the complexity of large-scale projects. However, as the number of actors and use cases increases and since the semantics are quite limited, it becomes difficult to maintain clarity and readability of the diagram. In such cases, it is best to separate the process into sub-models, so as not to harm the clarity and readability of the main process (IBM, 2023.; IONOS Digital Guide, 2020).

3.2.3. Expressiveness and precision

UML Use Case diagrams play a crucial role in communication within software development teams. Designed to be intuitive, they enable all stakeholders, including those without technical expertise, to quickly understand how the system works and how users interact with it. One of the main advantages of Use Case diagrams is their flexibility. They can be applied at various stages of the development cycle and are compatible with a variety of methodological approaches, from agile to traditional. This adaptability makes them

particularly valuable in projects with changing requirements or in environments where collaboration between diverse teams is essential. However, despite their advantages in terms of communication, these diagrams have their limits in terms of precision. They are characterised by generalising and abstracting scenarios without providing details of the internal workings of the systems. Furthermore, they do not cover the specific features of the technologies used, leaving out the way in which data is handled or stored. There is therefore a risk of over-simplification, which can be detrimental to a full understanding of the complex interactions within a system. So, for an accurate and detailed representation, it is often necessary to complement Use Case diagrams with other types of UML diagrams, such as Class diagrams or Sequence diagrams, which offer more technical and detailed perspectives on the system (GeeksforGeeks, 2024).

3.2.4. Abstraction level

The level of abstraction of UML Use Case diagrams is characterised by their ability to provide a high-level view of system functionality, focusing on the interactions between actors (users or external systems) and the system under consideration to achieve specific objectives.

The optimal level of abstraction of a UML Use Case diagram is important to maintain the balance between clarity and necessary detail. This level should be high enough to present the interactions between the actors and the system in a general way, avoiding going into technical specifics or implementation details. Too much detail can distract attention from the main objectives, confusing stakeholders and complicating interpretation of the system. According to Professor Vanderdonckt (Appendix A.2), it is important to focus on the essential Use Case and maintain an appropriate level of abstraction. In this way, the diagrams remain clear and understandable and perform their function effectively by providing an overview of functionality and making it easier to understand the essential workflows and collaboration between the various parties involved without getting stuck in complexity.

As Use Case is designed to illustrate high-level objectives and interactions, it is well suited to the early stages of software development. These diagrams provide a representation of the system from the user's point of view, allowing stakeholders to understand the main objectives and interactions without being submerged in technical detail. However, as processes become more complete, more precise diagrams are required. Activity diagrams or UML Sequence

diagrams can be used to go deeper into the behaviour of the system and its components, allowing a more accurate visualisation of the system.

In summary, Use Case can provide a high level of abstraction which limits the modelling of precise and complex processes.

3.2.5. Flexibility and extensibility

The flexibility and extensibility of Use Case are key aspects that enable them to adapt to varying degrees of project complexity. Indeed, these diagrams are adaptable and can model many systems in different domains, from software engineering to business processes. This great adaptability is caused by the focus on interactions between actors and the system. Indeed, these interactions can be customised through stereotypes and labelled values to meet specific needs. Stereotypes are extensions to the UML language that enable the creation of new element types based on existing elements, and are used to add additional metadata and classifications to standard UML elements, such as classes or components. The ability to define custom stereotypes and apply constraints through tagged values adds a layer of flexibility, allowing diagrams to be customised to capture the specific nuances of a project or domain, increasing their relevance and usefulness (Visual Paradigm, 2023).

Use Case diagrams can be extended and integrated with other UML diagrams, such as activity diagrams for detailed representations of workflows, or sequence diagrams for message exchanges. This integration provides a global view of the system (IONOS Digital Guide, 2020).

To take full advantage of the flexibility and extensibility of Use Case, it is important to standardise the use of stereotypes, labelled values and constraints within a project. To do this, it is needed to choose meaningful names for the stereotypes and labelled values to improve the consistency and clarity of the project, making the diagrams more understandable. It is acceptable to use free text to specify constraints when precision is not crucial, but the use of Object Constraint Language (OCL) for more rigorous expressions is preferable. OCL is a formal language used to specify precise and unambiguous constraints in UML models. It allows conditions, invariants, preconditions and postconditions to be expressed in a structured and rigorous way (Visual Paradigm, 2023).

The flexibility of Use Case is therefore facilitated by their ability to adapt to various domains, the use of customisable elements. However, its extensibility is limited on the basis of the Use Case model itself, but possible if it is integrated with other UML diagrams.

3.2.6. Ease of learning and use

UML Use Case offers an intuitive and simplified approach to representing the interactions between actors and the system, so they are easy to understand and use (Visual Paradigm, n. d.). According to the consultants (see Appendix B) we interviewed at Headmind Partners, their versatility and simplicity mean they can be used to model many processes, while Professeur Vanderdonckt emphasises their ability to represent user preferences and user-system interactions.

In terms of ease of use, Use Case adopts an intuitive approach that focuses on user needs. UML Use Case does not provide a detailed examination of each element of the process, but rather offers an overview of the relationships between use cases, actors and the system. Their versatility means that a wide range of systems can be modelled. However, to model complex systems, it is necessary to effectively decompose these systems in order to correctly identify the actors and their interactions which complicate the modelling (Cacoo Staff, 2021).

3.2.7. Implementation

Use Case is a diagram that models without entering into detail, and has fairly basic semantics. In fact, the use of standard symbols (matchsticks for actors, etc.) facilitates the creation of Use Case diagrams. In view of its simplicity, there are many software tools available for modelling this model. These tools offer features that simplify the diagramming process, facilitate collaboration and improve overall efficiency.

There are many, but here are just a few: Lucidchart, Draw.io, Microsoft Visio, SmartDraw, PlantUML, Visual Paradigm and Cacoo. All these software packages offer similar Use Case modelling functionalities (GeeksforGeeks, 2024.; Gleek, n. d.; Visual Paradigm, n. d).

3.3. Colored Petri Nets

This analysis focuses on CPNs because it is the PN extension most often used in the scientific literature to model the complaint handling process. It is therefore important to note that if this analysis focused on other PN extensions, the values for each criterion could be different.

3.3.1. Semantic

First of all, there are four objects that can be used to map a PN, namely: places, transitions, arcs and tokens. Their graphical representation is shown in Appendix V.

Graphically, places are represented by circles or ellipses and represent conditions in modelling (Reisig, 2013). A place always represents a passive element; it can hold, accumulate, or show things (Reisig, 2013; Valavanis, 1990). Then, there are two types of places: the input place and the output place (Plà et al., 2012). To establish which type of place it is, there are two rules:

1. If there is an arc from place p to t, then place p is the input of a transition t.
2. If there is an arc from t to p, then place p is the output of a transition t.

In the formal definition of a PN $P = \{P_1, P_2, \dots, P_n\}$ is a finite set of places (Medina-Garcia et al., 2023).

Regarding transitions, they are represented thanks to straight line segments assembled in rectangles, squares or bars and they represent events in modelling. A transition t is always an active component that can produce things, consume, transport or change them (Reisig, 2013). A transition that lacks an output place is referred to as a sink transition, whereas a transition without an input place is considered a source transition, as shown in Appendix W (Murata, 1989).

When there are one or more tokens at each input place p of a transition t, the transition is enabled. A transition that is enabled can be fired. When it is fired, tokens are removed from the transition's input location and replaced at the transition's output location (Murata, 1989). Appendix X depicts graphically the firing phenomenon. Another phenomenon is “self-loops”, as illustrated in Appendix Y, but this time they do not have to appear in a diagram for it to be correct (Medina-Garcia et al., 2023).

Two transitions are causally independent, or in other words, concurrent, when one transition may fire before or after or in parallel with the other, as it is shown in Appendix Z (Murata, 1989).

In the formal definition of a PN, $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of transitions (Medina-Garcia et al., 2023).

Concerning arcs, they are modelled by an arrow. Directed arcs link places and transitions to one another (Reisig, 2013). An arc models an abstract or, sometimes, only notional relationship between components such as access rights, spatial proximities, logical connections or immediate linkings. It is customary to denote the sets of arcs with F and to think of arcs in pairs: $F \subseteq (P \times T) \cup (T \times P)$ (Reisig, 2013). It should be noted that, according to the principles of PN theory, arcs are not allowed to directly connect two nodes of the same type (Raedts et al., 2007). There are also different types of arcs, as shown in Appendix V.

The last objects are the tokens. In the graphical representation, tokens are drawn as black dots. They are movable entities that may be found in places. Tokens can go across transitions in the network to symbolise the flow of data or control within the system and may change in quantity and position during the PN execution (Plà et al., 2012). Tokens are essential for triggering transitions that cause state changes. Indeed, tokens are moved from input places to output places when a transition fires, simulating the dynamic behaviour of a system (Medina-Garcia et al., 2023; Murata, 1989).

When a token is found in a place, it is assumed that the location's associated conditions are true. Alternatively, the presence of k tokens in one location denotes the availability of k data items or resources. In Appendix AA, it is shown common interpretations of transitions and their inputs and output places (Murata, 1989). Moreover, tokens are frequently used to represent objects in a modelled system, such as documents, resources, goods, and people. As a result, we frequently wish to depict these things' properties (van Der Aalst & Hee, 1996). However, The only way to distinguish between the tokens using the classical model is to make duplicates of the PN and place each token in a different net, which significantly increases the size and complexity of Petri Nets in real-world issues. Therefore, to eliminate confusion between tokens, CPNs give each token a category or identity. Colored Petri Nets offer the benefit of increased adaptability and flexibility while simulating business processes (Plà et al., 2012).

To sum up, classical PN and CPNs have a fairly simple semantic, made up of four distinct elements. This makes the visual representation of the process rather unintuitive and therefore difficult to understand for someone with little knowledge of this model and/or the process itself. Explanatory text is needed to accompany the diagram if we want to understand the process in detail, for example, to know how a task is executed (manually, using software, etc). However, one point of semantics on which the CPN is strong is the visual representation of the dynamics of different objects within the same process thanks to the coloured tokens.

3.3.2. Scalability

Petri Nets are a helpful tool for system modelling, but when used on systems with moderate to high complexity, they can be difficult to analyse. The complexity of the PN structure tends to expand with the complexity of the processes described, resulting in a notable rise in the number of places and transitions. This can result in a state space explosion problem, which makes analysis and verification more difficult as the number of alternative system states grows exponentially (Desrochers et al., 2005). To solve these problems of complexity and size associated with real systems, many authors have proposed extensions to the basic PN model, in particular “colour”, “time” and “hierarchy” (Van Der Aalst & Hee, 1996). Also, stochastic CPNs and HLPNs, which add exponentially dispersed transition rates to HLPNs and CPNs, shrink the size of the physical network by utilising the system's symmetries. However, this comes at the cost of increased analytical complexity. Furthermore, although its expansions produce accurate and straightforward models, they preclude the study of structural features (Desrochers et al., 2005).

At the same time, it is difficult to model time-dependent behaviour with conventional Petri Nets, which leads to a lack of precision and can therefore also have an impact on the scalability of a PN. Although this is a limitation of the classical PN and CPN, it is important to note that a solution is offered by the SPN. Indeed, this extension allows transition times to be exponentially distributed and therefore offers additional flexibility and adaptability, thus improving the scalability of networks (Kabir & Papadopoulos, 2019).

Finally, when a state explosion problem arises, or the underlying stochastic models are not amenable for tractable mathematical analysis, timed Petri Nets serve as a ready simulation model. Simulation may be conducted for the analysis of both logical and quantitative properties. The discrete event simulation can be driven from the model, sometimes using complex algorithmic strategies responding to real-time scheduling and control policies for DEDS's (Discrete event dynamic systems) (Jensen & Kristensen, 2009).

3.3.3. Expressiveness and precision

In Petri Nets, expressiveness is the capacity to depict a broad spectrum of system dynamics, behaviours, and structures. Accurately capturing the subtleties and specifics of these behaviours is what is meant by precision.

According to Tadao Murata, Petri Nets are a useful tool for characterising and analysing information processing systems that are described as being concurrent, distributed, asynchronous, nondeterministic, and/or stochastic.

Concurrency, also known as parallelism, takes place when operations related to independent entities in a system don't need to be synchronised. Additionally, it reflects a philosophy of time that holds that, from a logical standpoint, the only significant characteristic of time is its ability to define a partial ordering of the occurrence of events. Real-world events need varying lengths of time; the PN model accounts for this variety by not relying on the concept of time to dictate how events happen in order (Valavanis, 1990).

Then, the nondeterminism in the PN execution is reflected in the representation of conflicts . This occurs because the PN, like the system it represents, is understood as a series of discrete events whose order of occurrence is one among several that the basic structure permits to be conceivable. Any of the enabled transitions may occur if more than one transition is enabled at any given time. Although this nondeterminism has modelling benefits, it may significantly increase the complexity of the PN analysis. As a result, there is a widely acknowledged restriction that states that a transition's firing is immediate (Valavanis, 1990).

Additionally, Petri Nets' accuracy and solid mathematical foundation have led to the development of a plethora of tools and analytic techniques. Petri Nets are well-founded and have formal semantics available. Indeed, a number of enhancements (colour, time, hierarchy) and the semantics of the traditional PN have been officially described and this has several advantages. Firstly, when a workflow technique is described using a PN, there is no opportunity for more than one interpretation; each construction's meaning is clear. In this manner, endless debates concerning the exact meaning of a workflow procedure specification can be avoided. Then, a contract between sub departments can be modelled after a PN workflow. Conflicts on how common workflow procedures should be interpreted can be settled using formal semantics. Thirdly, a Petri-Nets-based workflow procedure's interpretation is tool-independent; it remains unchanged upon the introduction of a new WFMS version. Finally, property reasoning for a given work flow technique is made possible by the formal semantics. Dynamic features like livelock and deadlock can be demonstrated not to exist (van der Aalst & Hee, 1996).

It should be noted that certain characteristics of Petri Nets restrict the model's ability to capture the full complexity of the system’s behaviour and interactions.

The first limitation is the fact that to represent the conditions that occur in the system, only one class of places is used. This leads to an inadequate depiction of the system's true operation. Although Beck and Krogh's suggestion to use operation places and resource state locations enhances the representation[12], it does not account for every scenario. The complexity of the system's states and processes may not be fully captured if only one class of locations is used. This constraint limits the range of system behaviours and structures that may be accurately described, which has an impact on the PN model's expressiveness. PNs offer a strong basis for modelling concurrent and distributed systems because of their fundamental components : places, transitions, tokens, and arcs. Then, in classical PNs, the only existing class of tokens used to represent a piece of data or a control that moves across the system as the PN runs is another limitation. It is indeed not possible to depict the movement of components, resources, information, and control through the system at the same time. It may lead to oversimplification or abstraction of certain aspects of the system, potentially overlooking critical details or nuances in its operation. CPNs aim to overcome this drawback and improve the expressiveness of the model. Additionally, there is uncertainty regarding the composition and functionality of the system under model. There is indeed no indication of the precise pattern of token flow from the input places to the output places through the transitions at any transition with multiple inputs and numerous outputs (Valavanis, 1990). It becomes difficult to adequately represent the dynamics and behaviour of the system without exact knowledge of how tokens travel across transitions. Because crucial information regarding the dependencies and sequencing of token flows is missing, this uncertainty may result in a loss of integrity in the system's representation. Moreover, if the token flow is unclear, the complex connections and relationships that exist between the many components of the system may not be sufficiently recorded (van Der Aalst, 1996; Murata, 1989; Reisig, 2013).

Then, the asynchronous character of Petri Nets represents another restrictive characteristic of expressiveness. It reflects the absence of an intrinsic way to quantify time or the passage of time (Valavanis, 1990). In the meantime, it is challenging to model time-dependent behaviour using conventional PNs. Nevertheless, as mentioned in Section 3.3.2 of this paper, the SPN is an extension of NPs that allows transition times to be exponentially distributed. Moreover, in

[12] For more information on Beck and Krogh's suggestion, see: Beck, C., & Krogh, B.H. (1986). Models for simulation and discrete control of manufacturing systems. Proceedings. 1986 IEEE International Conference on Robotics and Automation, 3, 305-310.

order to allow instantaneous and temporal transitions in the same PN model, SPNs have been developed in GSPN (Desel & Esparza, 1995; Kabir & Papadopoulos, 2019).

It should also be noted that the unambiguity of the CPN diagram depends on the precision with which the tasks are described. Indeed, some processes, such as social or medical processes, may include implicit ambiguities in their tasks. For example, the task of "giving medication to the patient according to the patient's own needs" and not at a specific time and in a specific quantity, in the modelling of the care pathway for hospitalised patients.

In conclusion, despite these few limitations, when tasks are accurately and unambiguously described, CPNs are very accurate and expressive. This is due to their formal semantics and their ability to accurately model concurrency, the dynamic state of a system and its transitions.

3.3.4. Abstraction level

Petri Nets have the capacity to model a system hierarchically. It stems from their ability to replace a whole net with a single place or transition to allow for more abstract modelling or to replace places and transitions with subnets to allow for more detailed modelling (Farwer & Mistra, 2003). PNs provide a strong modelling framework that may be used to depict intricate systems. But the more complex the modelled system gets, the more complicated the PN model needs to be in order to effectively reflect it. There is frequently a trade-off between the model's overall abstraction level and the level of detail it contains when modelling complex or large-scale systems (Desrochers et al., 2005).

Petri Nets have an additional advantage over other process modelling methods in that they make it possible to explicitly model the state of a case. Petri Nets offer a state-based approach, while models like data flow diagrams and process algebras concentrate on event-based modelling, where transitions are explicit and states between transitions are implicit. Appendix AB shows the diagram corresponding to an event-based modelling. Places represent intermediate states in PNs, and transitions constitute tasks.

Thanks to this explicit representation, a clearer distinction can be made between task activation and execution. The same applies to the consideration of states between the execution of subsequent tasks (Aalst, 1998). Appendix AC shows the same graph as in Appendix AB converted in a state-based approach.

In summary, CPNs can provide a level of abstraction from low to moderate.

3.3.5. Flexibility and extensibility

Petri Nets may represent a wide range of systems, from simple to complicated, thanks to their modular nature. They can be modified to reflect many aspects, like concurrency, choice, and synchronisation, and expanded to simulate systems with an infinite number of locations and transitions. Furthermore, PNs have a wide range of applications, including process modelling, system design verification, planning and scheduling, proving their adaptability to many scenarios and capacity to change in response to particular requirements (van der Aalst & Van Hee, 1996).

So CPNs are highly flexible and extensible, thanks in particular to the range of possible token types, variables, the ability to create hierarchical models and the integration of temporal and stochastic elements. Extensibility in CPN is structured and formal thanks to the formal mathematical foundations of this model. Additional functionalities or extensions are therefore integrated with a precise formal definition of their semantics, enabling a clear understanding of their impact on the model as a whole, ensuring overall consistency and facilitating formal analyses.

3.3.6. Ease of learning and use

Although Petri Nets have undergone many improvements to address their shortcomings, experienced stakeholders still find them difficult to use for business process management, as they are still considered a non-user-oriented technique, especially due to a non-intuitive nomenclature. Indeed, a deep understanding of PN theory is required to correctly interpret the process and its dynamic state (Aldin & de Cesare, 2009; Jensen & Kristensen, 2009).

3.3.7. Implementation

Many PN tools have been developed because the PN can graphically characterise a large number of systems. With those tools, users can, on the one hand, examine the system's performance and depict their system in detail using PN tools. On the other hand, users can also use PN tools as a code generator and graphical editor. Additionally, the system can be simulated and model checked using a few PN tools (Thong & Ameedeen, 2014).

Listing all the existing tools is not in the scope of this work, but here are some of them: CPN Tool, ePNK, GPenSIM, HIQPN-Tool, IOPT-Tools, WoPeD, Wolfgang, ITS Tools, TAPAAL,

SYROCO, SNAKES, SimHPN, WINSIM, WebSPN, COSMOS, PNetLab, SIPN - Editor[13]. Some of them are specific or recommended for specific types of Petri Nets. For example, a wide range of CPN tasks, such as manipulation, simulation, and analysis, are commonly performed with CPNTools. Regarding SNAKES, the main objective is to function as a global PN library that can create and run most PN models and provide researchers with a tool to quickly prototype new ideas (Kumbhar & Chavan, 2023).

3.4. Comparison

Table 1 summarises the above analysis by assigning a score from one to five for each model according to each criterion, with one being the lowest score and five the best one. It summarises all the information we have given above by comparing the three models.

Regarding the abstraction level, based on the various interviews we conducted, we consider that a high level of it is rather negative because it does not provide a detailed view of the process, whereas a low level of abstraction is positive. Indeed, it emerged that the main objective of process modelling was to analyse the process in order to improve or document it. It is therefore interesting to use a model that has a low level of abstraction in order to be as precise as possible and to enable a good understanding of the process. This is why we associate a low level of abstraction with a good score of five and a high level of abstraction with a low score of one. Obviously, the desired level of abstraction depends on the modelling objective, so this classification may vary depending on the situation.

Based on our analysis we see that BPMN stands out with its comprehensive semantics and scalable behaviour. It is also flexible, extensible and easy to learn and use. Compared with Use Case and Petri Nets, it scores well for all the features we have mentioned. In fact, the two features for which it gets the lowest marks are expressiveness and precision. Nevertheless, it is still better than Use Case and Petri Nets for these two criteria. This makes it a fairly complete model and justifies its recurrent use by Headmind Partners consultants.

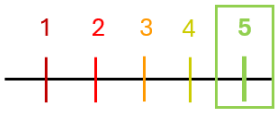
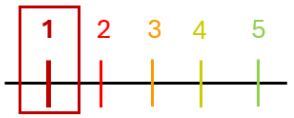
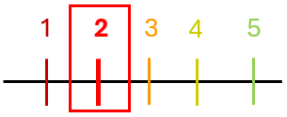
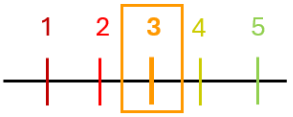
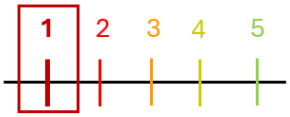
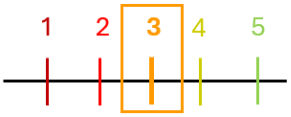
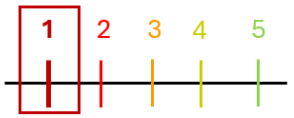
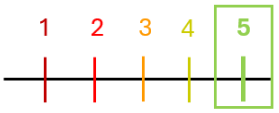
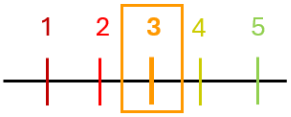
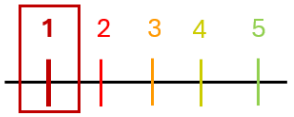
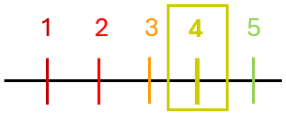
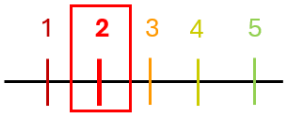
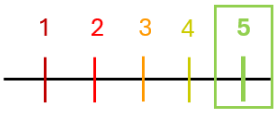
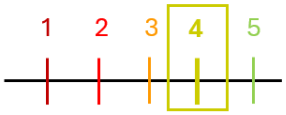
As far as the UML Use Case is concerned, it is clear that its strongest point is its ease of learning and use, which goes hand in hand with its fairly limited semantics, which is,

[13] For a comparison of the different Petri net tools, see: Kumbhar, V. B., & Chavan, M. S. (2023). A Review of Petri Net Tools and Recommendations. *Advances In Computer Science Research*, 710-721. https://doi.org/10.2991/978-94-6463-136-4_61 , or : Thong, W. J., & Ameedeen, M. A. (2014). A Survey of Petri Net Tools. In *Lecture notes in electrical engineering* (p. 537-551). https://doi.org/10.1007/978-3-319-07674-4_51

conversely, one of its weak points. This limited semantics also explains the other two weak points of the model, namely expressiveness and flexibility and the level of abstraction. We can quickly see that UML Use Case is behind BPMN and Petri Nets on almost all criteria, apart from scalability.

Finally, as far as PN is concerned, we can see from Table 1 that the two points on which it stands out positively compared with the other two models are its precision and expressiveness and its low level of abstraction. After that, it is much weaker in terms of learning and use, and scalability. In fact, it is quite hard to learn and use, especially for high-level processes. It gets both good and poor marks, which puts it somewhere between BPMN and Use Case.

Table 1: Comparative table of BPMN, Use Case and Petri Nets based on six criteria

Criteria	BPMN	UML Use Case	Petri Nets
Semantic completeness			
Scalability			
Expressiveness and precision			
Abstraction level			
Flexibility and extensibility			
Ease of learning and use			

3.5. Model choice context

In addition to the characteristics mentioned above, there are other criteria that assist the consultant in choosing the most appropriate model for modelling a given process. In fact, there are an infinite number of possible ways and diagrams for modelling the same process. When a consultant has to model a process using the method of his choice, various criteria come into play to help him choose the model he will use. Here is a description of the 5 elements most frequently cited by the people we interviewed, whether the consultants from Headmind Partners or the Professors from UCL Jean Vanderdonckt and Manuel Kolp (see Appendix A).

3.5.1. Modelling objectives

First of all, the primary criterion is the modelling objective, which will depend on the level of abstraction of the information required. For example, if the objective is to communicate the evolution of a process in an informative way to the whole company, then Petri Nets will not be the model of choice because some people in the company may not be familiar with the semantics of this model. A model with a higher level of abstraction and a more intuitive representation, such as BPMN, will be preferred. BPMN diagrams are commonly used to illustrate situations where process flow needs to be clearly communicated at various levels within an organisation. Moreover, because it is formalised in XML, BPMN is versatile, allowing both easy reading by humans and automated processing. As for them, PN networks are also often used to find a bug in a system or precisely to prove that there is no bug. Then, if the modelling is aimed at a company's customer experience department and enables the user's path to creating a subscription to be analysed, then Use Case will be the most appropriate model. These diagrams focus on the interactions between users and the system. On the other hand, if the objective is to automate a process, then the level of abstraction will have to be low, as it is important to clearly specify each stage of the process. In this case, models such as Petri nets or BPMN will be used. The modelling objective is therefore an important criterion in the choice of the model to be used.

3.5.2. Process type to model

Based on the criteria discussed earlier in this paper, it appears that some processes will be more easily and/or accurately modelled using a particular method. The type of process is therefore an additional criterion to be taken into account when choosing the modelling method.

In fact, for modelling simultaneous and parallel processes such as a manufacturing system in which several machines operate simultaneously to produce goods (for example, a car assembly line); for systems with synchronisation and resource sharing such as a traffic control system where traffic lights synchronise and share resources efficiently, for example; in the case of real-time systems such as a process for real-time monitoring and control of a power grid system; or even for distributed systems such as a distributed computing system where tasks are executed across multiple interconnected nodes; PN is the most suitable model due to

its ability to accurately capture and analyse dynamic behaviour, concurrency, synchronisation and resource sharing within systems.

Regarding BPMN, its ability to provide a clear, standardised and easily understandable graphical representation of business processes makes it the most appropriate method for modelling processes such as simple linear business processes such as a straightforward approval process where a request goes through a sequence of approval steps, like an employee onboarding process for example; for multi-step workflows such as a comprehensive project management process involving multiple stages, for example a procurement process; for conditional and event-driven processes like a process for handling customer orders that triggers different actions based on order status changes: or even for inter-organizational workflows, for instance a supply chain management process that involves interactions between multiple organisations.

Finally, concerning Use Case diagrams, due to their ability to capture and represent functional requirements, user interactions, and roles in a straightforward and comprehensible manner, represent the most suitable method for functional requirements modelling, for example in the case of an e-commerce platform to capture user interactions (browsing, purchasing, account management, etc); for modelling user-system interaction scenarios, for instance to illustrate how users interact with the system on a banking application; for high-level process or even for systems where user roles and goals are crucial, a concrete example could be modelling a Use Case diagram for a healthcare management system to depict interactions between doctors, patients and administrative staff. In general, Use Case diagrams are suitable for the early stages of a software development for gathering requirements and ensuring that the system fulfils user expectations.

3.5.3. Target audience of the model

The target audience is also a criterion which will help the consultant to define the model on the basis of which he will produce his diagrams. The more specialised the audience is in modelling languages, the more complex the model will be.

For people who are comfortable with business processes, BPMN notation is often the most appropriate. Because BPMN semantics are fairly simple and the symbols are intuitive and standardised, the diagrams are easy to interpret. Thanks to this visual simplicity, individuals

with no technical training can understand the processes modelled, making communication between the various stakeholders more effective.

On the other hand, PN nets are better suited to a technical audience with a solid understanding of the concepts. Petri Nets are powerful for modelling complex processes. However, their less intuitive graphical representation makes them less accessible to a non-technical audience. If the process is to be communicated to non-technical stakeholders, Petri Nets would not be the optimal choice because of the difficulty of mastering them.

Finally, UML Use Case is often used to capture the interactions between users and a system. On the one hand, this model is suitable for developers, systems analysts and software designers. On the other hand, the Use Case is relatively simple and, as such, largely understandable by people with no technical knowledge. However, to ensure full understanding by these non-technical stakeholders, it may be necessary to provide additional explanations.

3.5.4. Cost benefit ratio

The fourth decisive criterion for choosing a model is the cost/benefit ratio. In considering this ratio, companies need to find a balance between the resources invested (time, money, skills) and the results obtained.

As Wil van der Aalst and Kees van Hee (2002) explain in their book “Workflow Management”, achieving good external performance also requires considerable internal investment[14], such as overtime or additional resources to speed up processes. It should not be forgotten that the aim of any organisation is to minimise its total costs. To do this, it must carefully weigh the costs of poor external performance (such as lost sales) against the costs of the internal effort required. As can be seen from the graph in Appendix AD, it is essential to find the right balance to maximise efficiency while controlling expenses.

The principle is the same when it comes to choosing the model that offers the best return on investment, depending on the company's constraints.

[14] For more information about External performance indicators and Internal performance indicators, see page 100: van der Aalst, W., & Hee, K.M. (2002). Workflow Management: Models, Methods, and Systems. In *The MIT Press eBooks*. https://doi.org/10.7551/mitpress/7301.001.0001

First of all, BPMN is often associated with high costs, because staff have to be trained and licences for some modelling tools have to be obtained. However, the benefits of BPMN are also considerable. They are useful in identifying inefficiencies in order to improve the organisation's processes. On the other hand, for companies with smaller budgets, Use Cases are a less expensive alternative. These diagrams are easier to create and understand, requiring less specialist knowledge. Finally, Petri Nets also represent an investment in terms of training and modelling time. The costs involved are high, but are justified by the results in terms of accuracy and robustness (van der Aalst & Van Hee, 2002).

3.5.5. Available time for modelling

The last criterion that consultants mentioned the most was the time required to produce the diagram. For the same process, the three models do not require the same modelling time. Use Case is faster to diagram than BPMN, because it focuses on the interactions between players and systems rather than on operational details. Then, the semantics of the model are fairly basic. Secondly, the time required to create a BPMN model is generally longer than for a Use Case model, due to the complexity of the notation and the need to represent business processes accurately. Finally, for the same process, PN is the model with the longest modelling time. This is due to the formal and detailed nature of its notation. Modelling with Petri Nets often requires specialist expertise, which can affect the time available for modelling.

4. Comparative example applications

This section examines a customer complaints handling process, modelled using the three models to provide a concrete illustration of the criteria discussed in Section 3 of this document. After discussions with various Headmind Partners consultants (see Appendix B), it became apparent that this type of process was often encountered in their assignments.

4.1. BPMN application

LUTAAYA et al (2019) model this process as a BPMN in Figure 1.

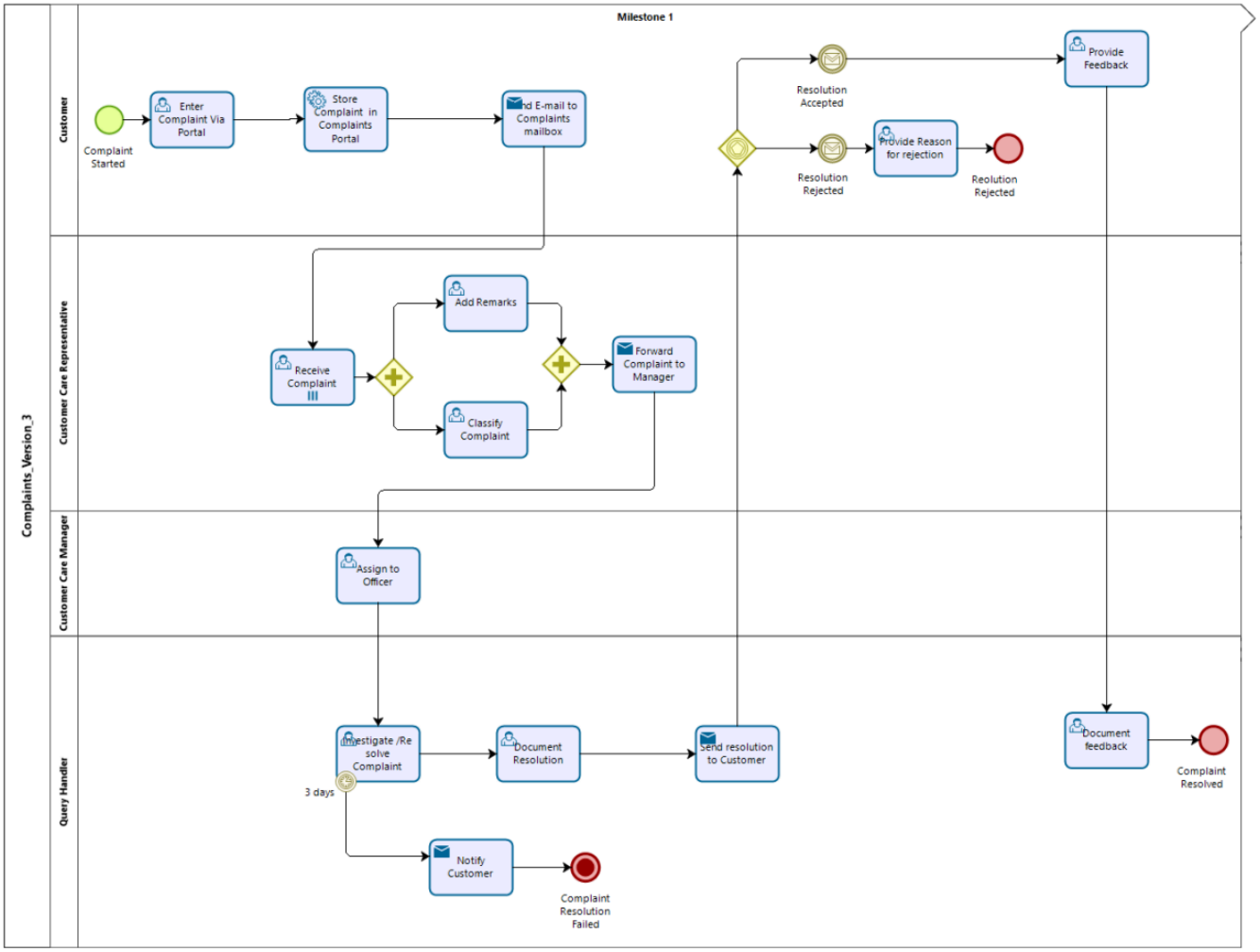


Figure 1: Example of a BPMN describing the customer complaint handling process

Source: Lutaaya et al., 2019

According to what is explained by the authors and what we see on Figure 1, the process can be defined with the following steps. First, a complaint is submitted by the user on the portal,

which initiates the process. The complaint is then held in the complaints portal. To notify the appropriate staff members about the new complaint, an email notification is sent to the complaints mailbox. The customer service representative therefore receives notification of the complaint(s) by email. Once the customer service representative has received the complaint, he files it and adds any comments at the same time. These are therefore parallel activities. The complaint is then forwarded to the Customer Services Manager who assigns the complaint to a request manager who then tries to find a solution by investigating the complaint. Two situations are then possible: In the first case, the request manager has found a solution within the required time (three days), the solution is therefore documented and a notification is sent to the customer. On the other hand, if no solution has been found within three days, the procedure is terminated. A notification is then sent to the customer to inform them that the resolution has failed. In the first case, a solution has been found, so the process continues. So, once the solution has been documented, there are still two possible situations: either the customer accepts the resolution or he rejects it. In the first case, the customer is then asked to provide feedback, which is documented by the request manager and the process then ends. In the second case, the customer can provide a reason for rejection and then the process ends.

4.2. Use Case application

Figure 2 represented by Markopoulos, P., and Marijnissen, P. J. (2000), also represents a customer complaint management process. The model focuses on user activities rather than system functionality. The purpose of the Use Case is not to describe the business process, but rather a single user task that is integrated into that process.

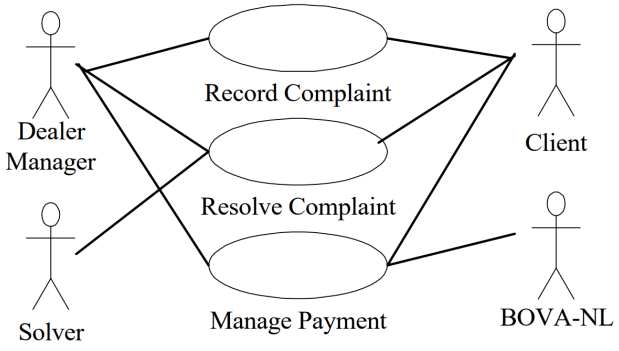


Figure 2: Example of a Use Case describing the customer complaint handling process

Source: Markopoulos, P., & Marijnissen, P. J., 2000

In the article, informative text accompanied the diagram to help the reader understand all the steps in the process.

In the context of complaint handling represented in Figure 2, the current situation within the BOVA organisation, the dealer plays the primary role. BOVA is a Dutch luxury coach manufacturing company. Other actors involved include clients, BOVA-NL, and solvers, which are contracted garages. The main event path of the complaint handling process consists of three key steps which are the use case : first, recording the complaint, then resolving the complaint, and finally, managing the payment. Once these steps are completed, the complaint is considered resolved and paid. Information about the complaint is then archived for later processing, ensuring follow-up and future analysis of incidents.

The complaint handling process within the dealer establishment begins with the “Record Complaint” use case. When a client telephones or mails a complaint, the dealer, who is the primary actor, receives the complaint from the client. The dealer then notes the complaint details on paper and subsequently types them into a standard request form. However, if the submitted data is incomplete, the dealer may need to call the customer to obtain the missing information or ask for the necessary details from the company's central database. This marks the completion of the recording phase of the complaint.

Following this, the “Resolve Complaint” use case is initiated. At this stage, the complaint details have already been recorded. The process involves both the dealer and the solver establishment. The primary actor remains the dealer, while the client and the solver (a contracted repair garage) are also involved. The dealer archives the claim, monitors the solver's progress in addressing the issue, and identifies any recurring problems. In some cases, the solver might deal directly with the client to expedite the resolution process. This ensures that the complaint is thoroughly investigated and resolved.

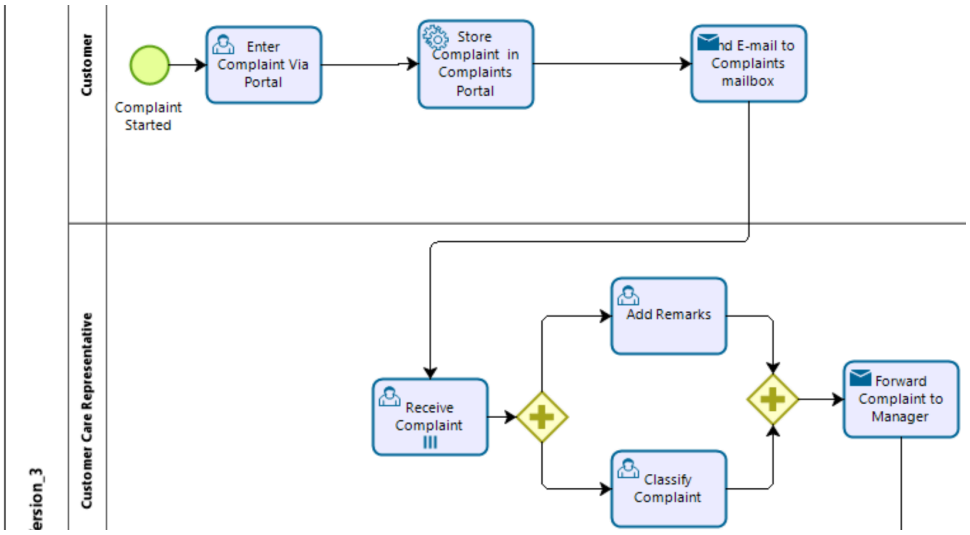
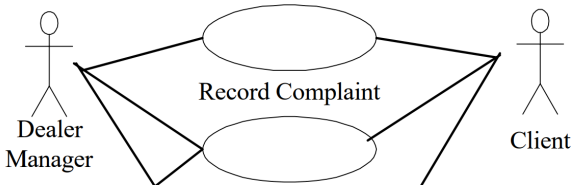
Once the complaint has been resolved, the “Manage Payment” use case comes into play. This occurs within the scope of the BOVA organisation. The primary actor is again the dealer, with BOVA-NL and the client as other actors. The process begins when BOVA-NL sends a judgement concerning warranty coverage. Following this, the dealer negotiates the price with the client. Additionally, the client may use the invoices from the complaint resolution to negotiate a discount for the purchase of a new coach. This finalises the payment aspect of the complaint resolution.

The Use Case presented above does not present all the existing facets of the Use Case nomenclature. Here are a few elements that could have been added to make it more complete. Firstly, only association relationships are represented. Association relationships are the relationships that exist between use cases and actors. Other types of relationships such as Extend Relationship, Include Relationship and Generalization Relationship are not used in the diagram. The authors must surely have found these relationships unnecessary for understanding the Use Case as they have represented it. Furthermore, in the above model, there is no boundary system. The author could have included one, as this is a fairly important element in Use Case, acting as a visual boundary that separates the system's internal functionality from the external players who interact with it. It would have taken the form of a rectangle framing the three Use Case. Based on our interviews and research, we can state that there is no single best way to model a process in Use Case format. In fact, the same process can be modelled in different ways, for example in this case with a system boundary as well.

4.3. BPMN and Use Case comparison

We can see that the BPMN and Use Case presented above both represent a customer complaint management process, but that the two graphs are not at all similar. First of all, the main difference between these two models is that the BPMN is much more detailed than the Use Case and therefore provides a better understanding of the process. In fact, it is clear that the Use Case does not provide many details on how to manage a customer complaint. For the same process, the Use Case represents the process in the form of just three steps, whereas the BPMN represents around twenty steps.

Table 2: Level of detail in task representation, BPMN and Use Case models

Comparison point	Representation of the task "receiving and recording complaints"
BPMN	
Use Case	

As we can see in the table above, the two models use very different methods to represent the creation of the complaint by the customer, and the sending, receipt and recording of the complaint by the "seller". We can see that for all these stages, the BPMN uses 7 different tasks, two gateways and one start event, whereas the Use Case only uses two actors and one use. As we said earlier, Use Case is not precise and is best understood at a high level and by a large number of people, whereas BPMNs give much more detail about the different stages, the choices to be made, etc.

Next, we can see that the two models do not read in the same direction. On the one hand, the BPMNs are read from left to right and top to bottom, just as you read a book. On the other hand, Use Case is read from the outside of the sheet towards the inside, which is less intuitive. In addition to this, BPMNs have start and end events which identify the beginning and end of the process, whereas Use Case does not. In a Use Case, it is sometimes difficult to understand where the process starts and where it ends.

Finally, a common point between the two models is that it is clear which stakeholder is responsible for which task in the process. One of the objectives of Use Case is to emphasise the role of each person in the process. Use Case focuses on the actors, who are represented by string characters, and not on the stages of the process itself. BPMNs, on the other hand, focus more on the stages of the process itself. However, thanks to the lanes and pools, the actors and users involved in the process are also clearly identified. In fact, depending on the pool/lane in which the process step is located, we know directly who is responsible for it. This gives model readers a clear understanding of who is responsible for which process step.

4.4. Colored Petri Nets application

A customer complaints process designed in CPN is modelled by van der Aalst & Van Hee (2002). The process represented in Figure 3 is described as follows. First, a customer complaint is registered. The client who filed the complaint and the department concerned by the issue are then contacted. The customer is then asked for more information about the complaint. The department concerned is informed of the complaint and may be questioned about its first response. These last two tasks can be carried out simultaneously or in any order, in other words, in parallel. After combining the data, a choice is made. Then either compensation is paid, or a letter is sent, depending on the decision. Finally, the complaint is filed.

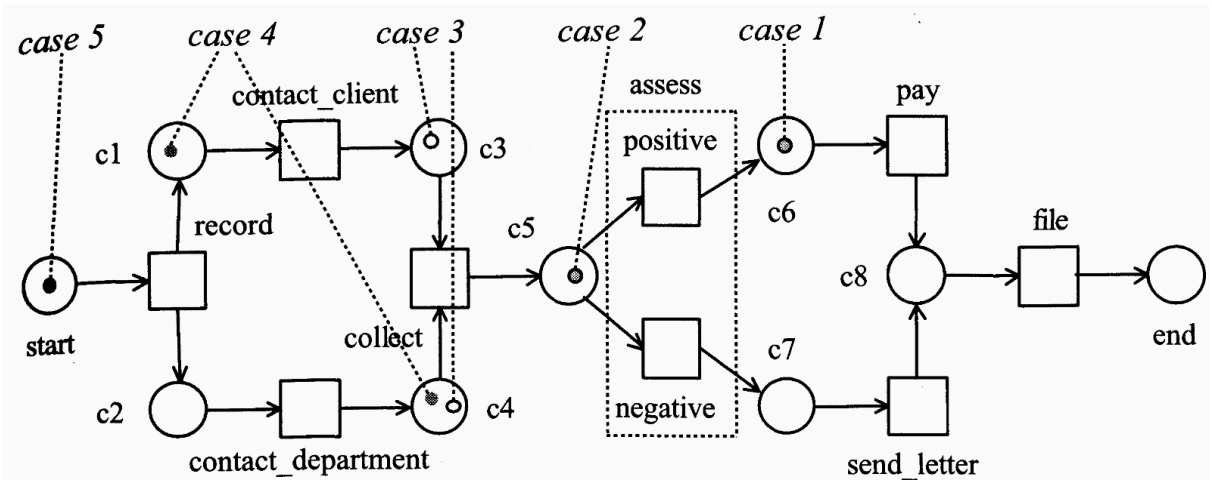


Figure 3: Example of a CPN describing the customer complaint handling process

Source: Van Der Aalst, W., & Van Hee, K. M., 2002

In this CPN[15], each task (such as registration, customer_contact, service_contact, payment and file) is modelled by a transition. Two transitions are used to model the assessment of a complaint. The first is a positive transition corresponding to a positive decision. The second is a negative transition, corresponding to a negative decision. Places other than “start” and “end” correspond to conditions that are, or are not, fulfilled by each case in progress. Conditions serve two crucial purposes: first, they guarantee that actions are completed in the correct order; second, they allow the case's status to be determined. Point c8, for example, ensures that a complaint can only be filed once it has been fully processed. This point also represents the state that exists between a complaint being fully processed and filed.

Regarding the dynamic of the process, the token in the place “start” illustrates the presence of a case. Then, once “record” has fired, there are two tokens in the diagram. Indeed, one is at c1 and a second one is at c2 but both represent the same case. This exemplified the fact that the number of tokens may fluctuate as a case is being handled. Indeed, a particular case is represented by a number of tokens which is equal to the number of its conditions that have been met. It should be noted that the conditions met do not in themselves determine the state of a case. In fact, using CPNs, a case can have one or more attributes. One aspect of CPN that is perhaps not clearly visible on the diagram is that the tokens corresponding to particular cases are separated so that they do not influence each other. It is in fact possible to identify the case to which the token refers thanks to the value initially given to each token. In this diagram, we can see that case 1 is practically completed, while case 5 is in the “start” state. To prevent tokens from various cases from becoming “mixed up” , every transition has a precondition stating that only tokens from the same case can be consumed at a single firing. For instance, this precondition will guarantee that the two white tokens for case 3 are used up if the transition “collect” in the diagram now triggers.

Beside that, each process must meet two conditions in order to guarantee correct termination of each case starting at the “start” position. The first condition is that it must be possible at any time to reach a state in which there is a token at the end, by executing a number of tasks. The second stipulates that all tokens must have disappeared when the “end” place contains one.

---

[15] W. Van der Aalst & Van Hee (2002) also modelled this process with a classical PN and an HPN, see: van der Aalst, W., & Hee, K.M. (2002). Workflow Management: Models, Methods, and Systems. In *The MIT Press eBooks*. https://doi.org/10.7551/mitpress/7301.001.0001

4.5. BPMN and Colored Petri Nets comparison

Even if the BPMN diagram and the CPN diagram do not come from the same authors and have therefore not been drawn on the basis of the same statement, the basic process remains the same and a parallel can therefore be drawn between the two models:

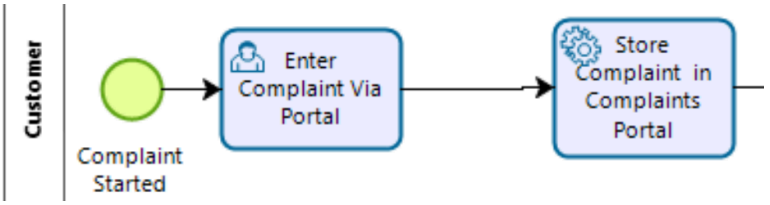
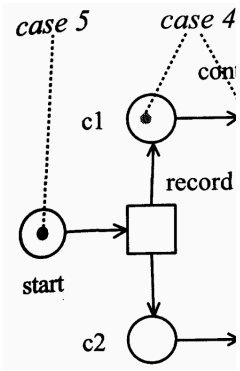
First, the complaint is registered. Then, the customer and the department concerned are contacted and an employee investigates the complaint. After that, a decision is made (in the BPMN statement, the customer accepts or rejects the proposed solution; in the CPN statement, the decision is based on the assessment of the complaint). The complaint is then filed and the process ends.

The first thing to note in the comparison between the two models, is that the diagrams read in the same direction, i.e. from left to right, which facilitates the understanding of the workflow.

Regarding the differences between the two diagrams, the most obvious visual difference between a BPMN and a CPN is probably the pool and lane structure present in a BPMN, which organises activities by roles or departments, a feature absent from a CPN. This structure clearly shows who performs each task making BPMN easier to read and understand than CPN. To find out who carries out the various stages, documentation to accompany the CPN diagram is needed, which does not allow the process to be read as smoothly as in BPMN.

Then, if we look at the modelling of tasks, we immediately see that they are visually explicit thanks to BPMN semantics. Whereas transitions are simply named in the CPN, tasks in the BPMN diagram are illustrated by an icon in the top left-hand corner of the rectangle, providing information about how the activity is going to be carried out. Table 3 clearly illustrates this difference in the level of detail when modelling a task.

Table 3: Level of detail in task representation , BPMN and CPN models

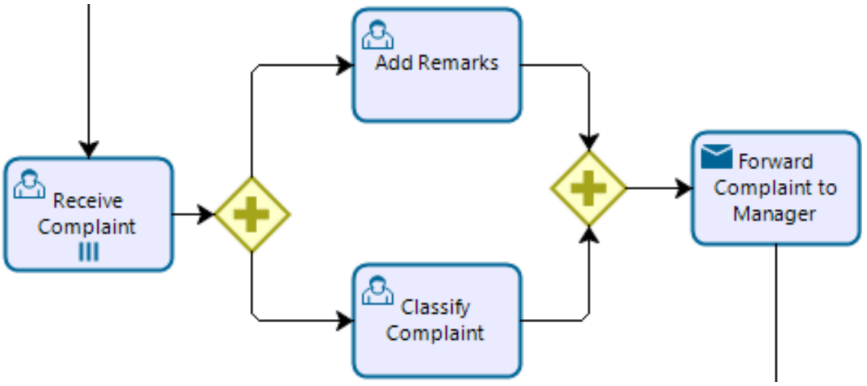
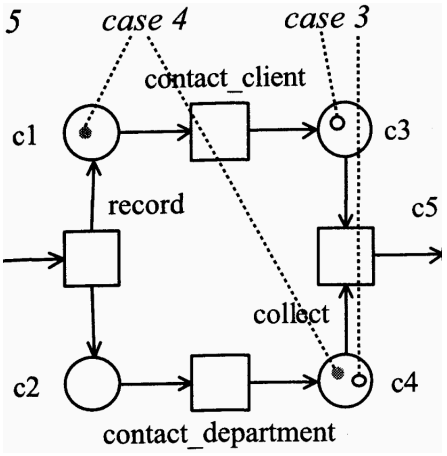
Comparison point	Representation of the task “registration and storage of the complaint”
BPMN	
CPN	

In this table it is shown that the BPMN diagram separates the registration of the complaint in the portal from its storage in the portal. A user task is used for the first one as a human performer carries out the task with the help of a software application. For instance this task can be programmed via a task list manager. This type of task can be found several times in the diagram, because the customer complaint management process requires human decision-making at several levels, such as classifying the complaint and deciding whether or not to pass it on to a manager. A service task is used for the second one because some sort of service is required. In this case it is the complaints portal. Regarding the CPN diagram, it uses only one transition to illustrate this stage. However, we can see that the token of case 4 is in place c1, which corresponds to the state of the system just after the complaint has been registered. We can therefore say that once it reaches c1 and c2, the complaint is stored. However, without accompanying documentation, tasks and information about their completion are more complicated to understand when looking at a CPN diagram than a BPMN. It would have been possible to add a place and a transition just after the “record” transition to illustrate this “storage” stage and separate it from the “record” stage. This would have been more complete but also more cumbersome visually. Unlike the BPMN diagram, which is more readable thanks to the intuitive semantic, the CPN diagram is more condensed

and quickly becomes complicated to read when there are a lot of places and transitions. This partly illustrates the scalability problem. Nevertheless, this issue is already improved by the use of a CPN. Indeed, with a traditional PN, a case-by-case diagram would have been necessary to differentiate between the various cases, which would have made the process more difficult to read and understand.

Still regarding activities, some can be performed simultaneously and Table 4 shows that this scenario is modelled similarly in both models.

Table 4: Parallel activities, BPMN and CPN models

Comparison point	Representation of parallel activities
BPMN	
CPN	

In BPMN, parallel tasks like “Add Remarks” and “Classify Complaint” are visually clear with multiple outgoing sequence flows from a gateway. The first parallel gateway is used to allow both user tasks to occur simultaneously while the parallel gateway after these tasks is used to join the parallel paths into a single one. This ensures that the process waits for the “Adding

remarks" and "Classify complaints" tasks to be completed before moving on to the next step. In the CPN diagram, the "record" transition leads to places c1 (contact customer) and c2 (contact department), indicating that these tasks can run in parallel. Each place can therefore contain tokens representing the progress of each task. Next, the "collect" transition combines tokens from places c3 and c4, indicating that the "contact_customer" and "contact_department" tasks must be completed before moving on to the next step. We can see here that the "record" and "collect" transitions act like parallel BPMN gateways. The only difference to note for the representation of concurrent activities is that synchronisation is explicitly handled by the parallel gateway in the BPMN model, whereas in the case of CPN it is implicit in the requirement that all necessary tokens must be present in the places connected to a transition before it can trigger. This implicit requirement makes the CPN diagram less intuitive visually, because by relying solely on token flows to model concurrent activities, a deeper understanding of PN theory is required to correctly interpret the process. However this is also a strong point of the CPN as these token flows are a crucial tool for process analysis and simulation.

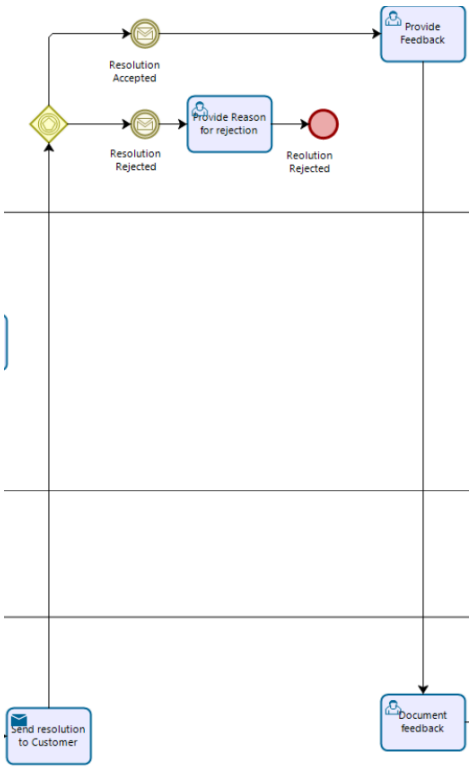
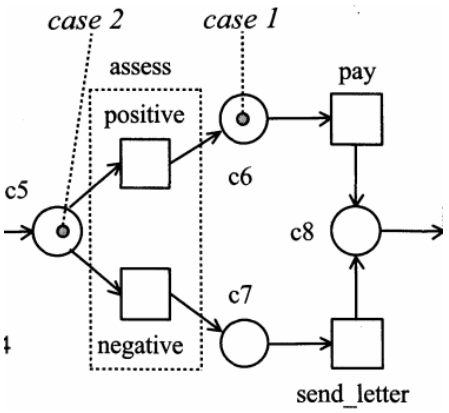
In addition to modelling concurrency, Table 4 shows a link between CPN tokens and parallel multi-instance markers in BPMN semantics. These markers indicate that multiple complaints can be handled simultaneously just as the different token colours in the CPN diagram show. However, BPMN semantics does not model the dynamic state of the system as CPN does through token flows.

Continuing with the order of the process, we see that a time element is integrated into the BPMN diagram. In fact, a condition, explicitly modelled by a boundary timer event, stipulates that a sub-process is triggered if no solution has been found by the query handler within three days. This condition is not represented in the CPN because it was not part of the statement on which the diagram is based but this could also have been modelled. To do this, we could have used tokens with timestamp attributes to model delays. Assuming that the CPN diagram includes exactly the same steps and tasks as the BPMN, to model this time condition, we would have had to add a place and a transition, which we could have called, for example, "timer" and "time expired" respectively. The tokens then included in "Investigate/Resolve Complaint" would be time-stamped, recording the moment when the tokens enter this place, and if the 3-day time limit is exceeded, the token triggers the "Time Expired" transition, allowing you to move on to "Notify Customer". However, while time event modelling is

possible in CPN, it is not as straightforward as in BPMN and explicit management of timestamps and transition conditions is required for both implementation and understanding.

Then, in the complaint management process, decisions must be made and it can be seen that both BPMN and CPN effectively model decision points but use different approaches. Table 5 compares these two models of modelling decision points. For ease of reading, the table is shown the other way round. An illustration of this entire table can be found in Figure 1.

Table 5: Decision point, BPMN and CPN models

Comparison point	**BPMN**	**CPN**
Decision point		

On the one hand, in the statement corresponding to the BPMN diagram, the decision to be made is the acceptance or not of the resolution by the customer. As this is a decision made by a person outside the company, the acceptance and refusal of the resolution are modelled by intermediate events of the “reception of messages” type. An event gateway is therefore used just before these events to represent the waiting stage until a response is received. The process will continue towards the event that occurs first. The gateway thus divides the flow into two mutually exclusive paths, clearly showing the logic of decision. On the other hand, the

decision point in the CPN is related to the assessment of the complaint. Namely, if the company considers that the complaint is founded, then financial compensation will be paid, in the opposite case, a letter will be sent. Decision points are modelled here by transitions that are triggered based on the presence of tokens and the evaluation of the conditions associated with these tokens. Decision logic is included in the conditions associated with transitions. This makes the decision points less visually explicit in a CPN than in a BPMN diagram and again requires a good understanding of the token flow and transition conditions.

Finally, the last observable difference on the diagrams is the number of possible endings. In this BPMN diagram, the process can end in three different ways, represented by end events: the first is an "end event" and the other two: "no end event" because the process fails in these two cases as the customer is not getting a suitable solution. On the other hand, we see that only one end is possible in the CPN diagram. This simply means that the process is not allowed to end in any other way than by sending a response (positive or negative) from the company as it is the only final result expected. A single end point provides a validation point where all the final conditions can be verified before formally completing the process. This de-clutters the diagram and makes it easier for stakeholders to follow the flow. The BPMN diagram could also have had only one end event[16], called for example "complaint filed" which would gather the three possible final situations by saying that whatever the outcome of the complaint, it must be classified for the process to end. Conversely, the CPN could also have had several end events by allowing additional scenarios. The only difference is that there is no visual distinction between the different end types in a CPN which would have complicated the visual understanding of the process. To sum up, BPMN offers an easy-to-understand high-level perspective of the complete process through its visual and structured parts. Although the CPN model provides a complete, sequential depiction of process dynamics, it necessitates a comprehension of tokens and transitions in order to accurately capture them.

4.6. Analysing the BPMN correctness thanks to Petri Nets

The ability of PNs to represent dynamic behaviour enables accuracy analysis of BPMN. Indeed, although BPMN is widely used to model business processes, it can lead to potential

[16] For a concrete example of a BPMN representing a complaint handling process with a single end-event, see: Ouyang, C., Dumas, M., Ter Hofstede, A., & Van der Aalst, W. (2006). *From BPMN process models to BPEL web services.* https://doi.org/10.1109/icws.2006.67

semantic errors such as livelocks and deadlocks due to its lack of unambiguous definition. This means that BPMN models can sometimes contain errors that are difficult to detect and correct (Li & Dai, 2018). Incorrect use of BPMN elements such as OR/XOR-join, AND-join or an event that does not allow more than one outgoing arc, leads to syntax errors. The validity of models, for example invalid flows or constructs, can be checked syntactically quickly using tools such as Intalio2 and BizAgi1 as explained by Mutarraf et al. (2018). On the other hand, structural errors are often detected at runtime, unlike syntactic errors, which are identified at the design stage using modelling tools. Structural errors, such as the incorrect use of splits and joins or the incorrect combination of elements, are difficult to identify at runtime due to the lack of formal semantics in BPMN models. The ability to statically analyse that model has become a desirable feature of BPMN process modelling tools. It has been found that semantic errors in a BPMN model, occasionally created by a user, can be detected using well-known verification techniques. To conduct a formal verification of a BPMN graph, a formal language like Petri Nets must be utilised (Mutarraf et al., 2018). PNs, thanks to their strong mathematical concepts, offer a reliable analysis of BPMN models, making it possible to guarantee their accuracy and check aspects such as semantic errors, the absence of deadlock[17], invariance properties and the termination of procedures (van der Aalst & Hee, 1996; Li & Dai, 2018; Mutarraf et al., 2018). Due to their effectiveness in analysing discrete event systems and the availability of analysis tools, PNs are considered the best language for these types of analyses (Mutarraf et al., 2018).

Specific mapping rules, (Appendix AE), are required to convert BPMN to PNs. These rules can be implemented in languages like C++ and XSLT. By following these guidelines, the final PNs are guaranteed to retain the original intended behaviour of BPMN models. For example, Appendix AF shows time-out exceptions and other specific behaviours that are mapped accurately to preserve the semantics of the model. Appendix AG illustrates the different instruments and modifications applied to PN analysis and conversion, such as Yasper, Woflan, INA and LoLA (Raedts et al., 2007). Other tools can obviously be used effectively, as shown by Dijkman et al. (2007) who performed a static analysis of a PN representation of BPMN using the ProM tool to check for dead tasks[18] and the absence of improper completion[19].

[17] A set of processes is deadlocked if every process in the set is waiting for an event that can only be brought about by another process in the set (Kowalski & Goetz, 2005). Deadlocks represent dead marking where the sink place is unmarked (Dijkman et al., 2007).

[18] The absence of dead tasks means that there are no tasks in a model that can never be executed. This can be confirmed by looking for dead transitions in the corresponding network (Dijkman et al., 2007).

[19] The absence of inappropriate completion means that, in a process, every instance is eventually completed correctly by reaching the end event and no more tasks are activated.(Dijkman et al., 2007).

5. Conclusion

This final chapter concludes the dissertation by explaining its various contributions (theoretical, managerial and CSR), criticising the limitations set out at the beginning of the paper and suggesting avenues for future research.

5.1. Theoretical contribution

In terms of its theoretical contribution, this master's thesis is distinguished by its ability to critically synthesise existing knowledge. Its main objective was not to make new theoretical contributions to the fields of BPMN, Use Case and Petri Nets. Given the abundance of theoretical literature already available on these process modelling models, our work focuses on the comparative and critical analysis of these models, providing an easily accessible overview of the strengths and weaknesses of each model. Nevertheless, thanks to our interviews, we have been able to analyse the application of the different models in concrete cases. In fact, we have established typical processes for applying modelling methods in client missions.

5.2. Managerial contribution

This master's thesis firstly highlights the importance of adopting a critical and constructive approach when choosing a modelling model. It shows that a good understanding of modelling models enables them to be used appropriately by assessing their suitability for the process to be modelled. This allows the company to gain in efficiency by taking full advantage of the model's benefits and enabling the most appropriate visual representation possible according to the objective sought. Indeed, when comparing BPMN, Use Case and Petri Nets, it is quite clear that no tool is superior to another; each has advantages and limitations depending on its application context.

Firstly, the BPMN model is particularly useful for business process management thanks to its ability to represent complex operations visually and intuitively. Its flexibility and extensibility mean that it can be used to describe processes of different sizes and complexity. However, although it is excellent for representing operational processes, it can be less accurate than Petri Nets for some technical aspects.

Secondly, Use Case stands out because it focuses on how users interact with the system, providing a behavioural perspective on functional needs that stakeholders will easily understand. Their simplicity and accessibility make them ideal for modellers and/or a target audience with no particular modelling knowledge. These features also make it a good option for modelling systems of various sizes, although it lacks the semantic depth to describe complex internal processes, an area where BPMN is more robust.

Secondly, Petri nets are appreciated for their accuracy, which exceeds that of the other two models, and their ability to analyse and simulate complex systems. However, this makes it more expensive than the other two, and negatively affects its ease of use by a non-technical audience and/or one with no knowledge of Petri net theory.

Besides that, it is clear that if the modeller has a good knowledge of the model used, there is less risk of error in the diagram, resulting in a better understanding and application of the process by the company.

Then, by showing the importance of choosing the model according to the type of process you want to model, this master thesis supports the need for a consultant to be familiar with various models in order to be versatile and best prepared to meet the different needs of clients.

Moreover, sufficient knowledge of several process modelling models also enables them to be combined to cover several levels of complexity and achieve a more holistic approach to process modelling.

Finally, as our various interviews (see Appendix B) have shown, few people in companies (Headmind Partners or client companies) are familiar with the BPMN model and even fewer are familiar with other models. If we assume that similar observations could be made in other consultancy companies, a good knowledge of various models by a consultancy company would enable it to stand out from the others by giving it a competitive advantage in the market.

In conclusion, this master's thesis shows that consultancy firms gain in efficiency by mastering several business process modelling models, avoiding wasted time due to misunderstandings of the diagrams or errors in setting up the process. That is why it is important for consultants to continually learn about different modelling tools in order to create and maintain a competitive advantage.

5.3. Contribution to CSR

Although the primary objective of this document is not to contribute directly to a company's CSR performance, it does contribute in a number of indirect ways. Better modelling of processes leads firstly to a better understanding of them by the stakeholders, which in turn facilitates communication and collaboration within the company concerning these processes.

This better understanding of processes also enables them to be applied more efficiently, and therefore to make better use of time and resources. This allows workers to focus their time on the most important tasks and reduces waste of all kinds (energy, raw materials, etc.). Finally, better process modelling also strengthens the transparency of operations within the company and with external stakeholders. All this can lead to a more sustainable and socially responsible business environment.

5.4. Critics of limits

In terms of the limits we set at the beginning of this thesis (see Section 1.3), we can say that they were well established. In the majority of cases, they did not limit us in our research and, on the contrary, allowed us to deepen our research within the framework we had set ourselves.

Firstly, with regard to the choice of models to compare with BPMN, we could have chosen models other than those we have used. We could have chosen an intermediate model between Use Case and BPMN, such as the Activity diagram. However, there would have been a risk that the models would have been too similar to each other and that the comparative analysis we carried out would not have been relevant. Secondly, we did well to concentrate on the Colored Petri Nets extension, as this enabled us to delve deeper into our research into this model without getting lost in all the extensions available. The scope we have set is relevant to our study.

In addition, the diagrams we used in Chapter 4 come from different sources and were not produced by the same authors or on the basis of the same complaint analysis statements. It would have been interesting to compare a BPMN, a Use Case and a CPN produced by the same author and based on the same statement. In this context, we do not have a standardised real case, so we cannot compare and measure the differences accurately. It would have been more relevant to have a real company and to be able to compare, for a real case, different models produced by the consultants we were following. We could have obtained different

analysis results that would have been more precise and more realistic. This approach could have added a theoretical contribution to our thesis by providing information that is realistic and even different from the ideological information found in scientific literature. The literature usually publishes the results of successful applications. In this way, we could have demonstrated the effectiveness or otherwise of the models in a real case and added an objective critique and a temporal and monetary dimension of modelling to our analysis.

Finally, with regard to the consultants interviewed, we concentrated on Headmind Partners employees. We could perhaps have interviewed consultants from other companies. This would have enabled us to obtain different opinions on the models studied, to draw more general conclusions about the consultant's world and possibly to have access to process models. However, this was not a limiting factor for the completion of our thesis.

In the end, the limits we had set ourselves were not restrictive for the completion of this thesis and for the most part did not distort the results obtained.

5.5. Future prospects

Emerging trends have an impact on different areas such as process-oriented case management, etc. Models such as Gartner's hyper cycles[20] (see appendix AH) are frequently used to illustrate this, showing the evolution of a technology from its inception to its maturity (von Rosing et al., 2015). These trends reflect changing business environments, driven by digital transformation and the pursuit of operational excellence.

Robotics Process Automation (RPA) is an emerging trend in process modelling. With the advent of low- and no-code tools, RPA is becoming accessible to non-technical users, enabling greater adoption and use of this technology within organisations. In addition, the consolidation of RPA with other technologies such as BPM and Intelligent Document Processing (IDP) into a single platform model is an important trend. The aim is to provide a more effective approach to optimising business processes and customer interactions, offering innovative ways of automating repetitive tasks, improving operational efficiency and contributing to the digital transformation of businesses (Veenendaal, 2023).

In addition, the integration of AI and machine learning into process modelling is revolutionising the way companies approach process optimisation. These technologies make

[20] This hyper model is used to provide an objective perspective on process trends, taking into account adoption, performance and maturity lifecycle (von Rosing et al., 2015).

it possible to automate complex tasks, make accurate predictions and make decisions based on historical data, enabling problems to be anticipated and performance to be improved. They can propose improvements or automations, enhancing the customer experience through personalised services and recommendations (Cejas, 2023).

Next, We can also focus on trends over the short, medium and long term.

First of all, in recent years, companies have focused more on the customer experience in response to changing expectations and competitive constraints. This means adapting products, services and internal operations to customers' preferences. Secondly, there is currently a trend towards faster, more agile processes, aided by technological advances. The aim is to react quickly to market dynamics and customer feedback (Wyatt, 2019).

Then, as regards medium-term trends, i.e. between two and five years, we expect technological integration. In fact, there are plans to deepen the collaboration and integration of processes with IT departments in the coming years. It is essential to ensure that the digital tools and systems match the company's objectives, in particular in areas such as mortgage lending, financial transactions and employee induction. This can significantly reduce processing times and improve customer satisfaction (Wyatt, 2019).

Finally, for the next 5 years, on the one hand, with the growing importance of sustainability, companies will have to take environmental aspects into account in their process models. The main objective is to improve the efficiency of resource use, reduce waste and create supply chains capable of coping with future shocks. On the other hand, artificial intelligence and machine learning will expand. This will offer the possibility of making predictions, taking decisions and automatically optimising processes (von Rosing et al., 2015).

Collectively, these trends signify a move towards more intelligent and adaptive process modelling, reflecting the wider move towards digital transformation and operational excellence in today's rapidly changing business environment.

5.6. Proposals for future research

As this critical review of BPMN with a comparison of the Use Case and PN comes to a conclusion, it is interesting to suggest a number of possible approaches for future research. The following suggestions could enrich our knowledge of these models and meet the needs of companies by incorporating technological advances.

A first idea is to use automation methods for existing modelling tools. For example, it could be useful to create extensions to facilitate conversion between BPMN, Use Case and Petri Net. It would be possible to automate the conversion from one model to another using these tools to reduce manual errors.

Another automation idea could concern model validation. It would be possible to use machine learning algorithms to identify inconsistencies in BPMN models to improve the accuracy and speed of model validation.

Thirdly, another idea would be to propose theoretical improvements to the semantics of the models in order to increase their expressiveness and reduce any ambiguities. Advanced semantics can lead to more precise and comprehensible models.

Finally, it would be interesting to do a case study on the implementation of models within companies. The aim is to assess the effectiveness of the models and their return on investment. This could be an interesting idea for consultants, to see whether the diagrams they have produced have been useful to the company. From the reader's point of view, another interesting research proposal is to conduct controlled experiments to test the understanding of the models by novices and experts. These experiments could help identify the elements of the models that require improvement and create appropriate training for users.

These proposals aim to deepen our knowledge of the BPMN, Use Case and PN models.

Bibliography

Adel, A. (2023, March 19). *The history of BPMN - Abbas Adel - medium*. Medium. Retrieved April 28, 2024, from https://abbasadel.medium.com/the-history-of-bpmn-6b539272ce10

Aguilar-Savén, R. S. (2004). Business process modelling : Review and framework. *International Journal Of Production Economics*, *90*(2), 129‑149. https://doi.org/10.1016/s0925-5273(03)00102-6

Aldin, L. & de Cesare, S. (2009) *A COMPARATIVE ANALYSIS OF BUSINESS PROCESS MODELLING TECHNIQUES,* [Department of Information Systems and Computing Brunel University Uxbridge, U.K.], Microsoft Word - UKAIS 2009 1.0 final .doc (brunel.ac.uk)

Alotaibi, Y. (2014). Business process modelling challenges and solutions : a literature review. *Journal Of Intelligent Manufacturing*, *27*(4), 701‑723. https://doi.org/10.1007/s10845-014-0917-4

Athuraliya, A. (2023, Decembre 12). *What is BPMN ? The Easy Guide to Business Process Modeling Notation*. Creately. Retrieved March 10, 2024, from https://creately.com/guides/what-is-bpmn/

Bacastow, T. (n. d.). *Elements of a Use Case Diagram*. GEOG 468 - Geographic Information Systems Design And Evaluation. Retrieved March 31, 2024, from https://www.e-education.psu.edu/geog468/l8_p4.html

Balbo, G. (2007). Introduction to Generalized Stochastic Petri Nets. In *Formal Methods for Performance Evaluation* (Vol. 4486, p. 83‑131). Springer. https://doi.org/10.1007/978-3-540-72522-0_3

Bause, F., & Kritzinger, P. S. (1998). Stochastic Petri Nets : An introduction to the Theory. *ACM SIGMETRICS Performance Evaluation Review/Performance Evaluation Review*, *26*(2), 2‑3. https://doi.org/10.1145/288197.581194

Best use case tool. (n. d.). Visual Paradigm. Retrieved April 15, 2024, from https://www.visual-paradigm.com/solution/usecase/use-case-tool/

BPMN - A Comprehensive Guide. (2023, September 12). Visual Paradigm. Retrieved March 15, 2024, from https://guides.visual-paradigm.com/1595-2/#BPMN_A_Larger_Example

BPMN : définition, utilisation et exemples. (n. d.). Lucidchart. Retrieved April 15, 2024, from https://www.lucidchart.com/pages/fr/bpmn

Building flexibility into BPMN models | Camunda 8 Docs. (n. d.-c). Camunda. https://docs.camunda.io/docs/components/best-practices/modeling/building-flexibility-into-bpmn-models/

Cacoo Staff. (2021, December 23). *How a UML use case diagram can benefit any process*. Nulab. Retrieved April 5, 2024, from https://nulab.com/learn/software-development/how-a-uml-use-case-diagram-can-benefit-any-process/

Callou, G., Maciel, P., Tutsch, D., Arajo, J., Ferreira, J., & Souz, R. (2012). A Petri Net-Based Approach to the Quantification of Data Center Dependability. In *InTech eBooks*. https://doi.org/10.5772/47829

Carchiolo, V., Catalano, G., Malgeri, M., Pellegrino, C., Platania, G., & Trapani, N. (2019). BPM Tools for Asset Management in Renewable Energy Power Plants. *Annals Of Computer Science And Information Systems*. https://doi.org/10.15439/2019f110

Cejas, R. (2023, Octobre 3). Emerging Trends in Process Modeling and Design. *Flokzu*. Retrieved Mai 5, 2024, from https://flokzu.com/bpm/emerging-trends-in-process-modeling-and-design/

Ciaramella, A., Cimino, M.G., Lazzerini, B., & Marcelloni, F. (2009). Using BPMN and Tracing for Rapid Business Process Prototyping Environments. *International Conference on Enterprise Information Systems*. https://doi.org/10.5220/0002005002060212

Corradini, F., Morichetta, A., Polini, A., Re, B., & Tiezzi, F. (2020). Collaboration vs. choreography conformance in BPMN. *Logical Methods in Computer Science*, *16*.

Desel, J., & Esparza, J. (1995). *Free choice Petri Nets*. Cambridge University Press. https://doi.org/10.1017/cbo9780511526558

Desrochers, A., Deal, T., & Fanti, M. (2005). Complex-Valued token Petri Nets. *IEEE Transactions On Automation Science And Engineering*, *2*(4), 309-318. https://doi.org/10.1109/tase.2005.853393

Diagram maker for developers. (n. d.). Gleek. Retrieved April 25, 2024, from https://www.gleek.io/blog/best-uml-tools

Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information And Software Technology*, *50*(12), 1281-1294. https://doi.org/10.1016/j.infsof.2008.02.006

Eby, K. (2016, Novembre 28). Beginners Guide to Business Process Modeling and Notation (BPMN). *Smartsheet*. Retrieved March 31, 2024, from https://www.smartsheet.com/beginners-guide-business-process-modeling-and-notation-bpmn

Education, I. C. (2022, January 6). *The Basics of Business Process Modeling and Notation (BPMN) - IBM Blog*. IBM Blog. Retrieved Mai 10, 2024, from https://www.ibm.com/blog/bpmn/

Elavarasan, D., Vincent, D. R., Sharma, V., Zomaya, A. Y., & Srinivasan, K. (2018). Forecasting yield by integrating agrarian factors and machine learning models : A survey. *Computers And Electronics In Agriculture*, *155*, 257‑282. https://doi.org/10.1016/j.compag.2018.10.024

Endert, H., Hirsch, B., Küster, T., & Albayrak, S. (2007). Towards a Mapping from BPMN to Agents. In *Service-Oriented Computing : Agents, Semantics, and Engineering* (Vol. 4504, p. 92‑106). Springer. https://doi.org/10.1007/978-3-540-72619-7_7

Farwer, B., & Misra, K. (2003). Modelling with hierarchical object Petri nets. *Fundamentae Informaticae*, 55(2),129‑147.

Frye, M. K. (2024). *La gestion des processus métiers : qu'est-ce que c'est ?* MuleSoft. Retrieved May 5, 2024, from https://www.mulesoft.com/fr/resources/esb/what-is-business-process-management

Goetz, J., & Kowalski, K. (2005). Using Petri Nets to Introduce Deadlock Concepts. *Computer Science And Information Systems*, 351‑366. https://www.researchgate.net/publication/329504074_Using_Petri_Nets_to_Introduce_Deadlock_Concepts

Guillaume, J., & Latapy, M. (2006). Bipartite graphs as models of complex networks. *Physica. A*, *371*(2), 795‑813. https://doi.org/10.1016/j.physa.2006.04.047

Heiner, M. (1992). *Petri net based software validation (prospects and limits), Technical report No. TR-92-022.* International Computer Science Institute.

Heiner, M. (2004). *Petri net based software validation (prospects and limits), Technical report No. TR-92-022.* International Computer Science Institute.

Houhou, S., Baarir, S., Poizat, P., & Quéinnec, P. (2019). A First-Order Logic semantics for Communication-Parametric BPMN collaborations. In *Business Process Management. BPM 2019. Lecture notes in computer science* (Vol. 11675, pp. 52–68). Springer, Cham. https://doi.org/10.1007/978-3-030-26619-6_6

How to Develop As-Is and To-Be Business Process ? (n. d.). Visual Paradigm. Retrieved April 16, 2024, from https://www.visual-paradigm.com/tutorials/as-is-to-be-business-process.jsp#:~:text=Developing%20and%20populating%20As%2DIs,process%2C%20culture%2C%20and%20capabilities.

Ivanov, S., Kalenkova, A., & Van Der Aalst, W. (2015). BPMNDiffViz : A Tool for BPMN Models Comparison. In *Proceedings of the BPM Demo Session 2015 Co-located with the 13th International Conference on*

Business Process Management (BPM 2015), Innsbruck, Austria, September 2, 2015 (Vol. 1418, p. 35-39). Florian Daniel and Stefan Zugal. http://ceur-ws.org/Vol-1418/paper8.pdf

Jensen, K. (1997). A brief introduction to coloured Petri Nets. In *Lecture notes in computer science* (p. 203-208). https://doi.org/10.1007/bfb0035389

Jensen, K., & Kristensen, L. M. (2009). Coloured Petri Nets : Modelling and Validation of Concurrent Systems. In *Springer eBooks* (1st ed.). Springer Berlin, Heidelberg. https://doi.org/10.1007/b95112

Kabir, S., & Papadopoulos, Y. (2019). Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments : A review. *Safety Science*, *115*, 154-175. https://doi.org/10.1016/j.ssci.2019.02.009

Kumbhar, V. B., & Chavan, M. S. (2023). A Review of Petri Net Tools and Recommendations. *Advances In Computer Science Research*, 710-721. https://doi.org/10.2991/978-94-6463-136-4_61

Li, L., & Dai, F. (2018). Transformation and Visualization of BPMN Models to Petri Nets. In *IOP conference series. Earth and environmental science* (Vol. 186, p. 012047). https://doi.org/10.1088/1755-1315/186/5/012047

Lorbeer, J. U. (2018). *Hierarchical reconfigurable Petri Nets* [Master's thesis, Hochschule für Angewandte Wissenschaften Hamburg]. http://hdl.handle.net/20.500.12738/8414

LUTAAYA, S., KIBUUKAMUSOKE, F., ODONG, S. P., & APIO, E. (2019). *Business Process Modelling and Analysis : Automating the business Processes of Frisco Cola Company December 2019*. Retrieved May 10, 2024, from https://doi.org/10.13140/RG.2.2.20377.98406

Manning, K. (2021, March 25). *A Complete Guide to Business Process Documentation with Examples*. ProcessMaker. Retrieved April 5, 2024, from https://www.processmaker.com/blog/business-process-documentation/

Markopoulos, P., & Marijnissen, P. J. (2000). UML as a representation for Interaction Design. In C. Paris, N. Ozkan, S. Howard, & S. Lu (Eds.), *Proceedings OZCHI 2000 Interfacing Reality in the new millenium, CSIRO* (pp. 240-249). CSIRO.

Mathenge, J. (2020, July 13). *Business Process Modeling Notation (BPMN) : Getting Started with Visualization*. BMC Blogs. Retrieved March 20, 2024, from https://www.bmc.com/blogs/bpmn-business-process-modeling-notation/

Medina-Garcia, S., Medina-Marin, J., Montaño-Arango, O., Gonzalez-Hernandez, M., & Hernandez-Gress, E. S. (2023). A Petri Net Approach for Business Process Modeling and Simulation. *Applied Sciences*, *13*(20), 11192. https://doi.org/10.3390/app132011192

Microsoft 365 Team. (2019, Octobre 1). *De beginnershandleiding voor het gebruik van BPMN bij je dagelijkse werkzaamheden*. Microsoft. Rettrieved April 23, 2024, from https://www.microsoft.com/nl-be/microsoft-365/business-insights-ideas/resources/the-guide-to-using-bpmn-in-your-business

Molloy, N. (1982). Performance analysis using stochastic Petri nets. *I.E.E.E. Transactions On Computers/IEEE Transactions On Computers*, *C-31*(9), 913-917. https://doi.org/10.1109/tc.1982.1676110

Monsalve, C., April, A., & Abran, A. (2015). *Business Process Modeling with Levels of Abstraction : A Survey of Practitioners with Experience*. IEEE COLCOM. https://doi.org/10.1109/colcomcon.2015.7152080

Mpardis, G., & Kotsilieris, T. (2010). *Bank Loan Processes Modelling Using BPMN*. https://doi.org/10.1109/dese.2010.45

Murata, T. (1989). Petri nets : Properties, analysis and applications. *Proceedings Of The IEEE*, *77*(4), 541-580. https://doi.org/10.1109/5.24143

Mutarraf, U., Barkaoui, K., Li, Z., Wu, N., & Qu, T. (2018). Transformation of Business Process Model and Notation models onto Petri nets and their analysis. *Advances In Mechanical Engineering/Advances In Mechanical Engineering*, *10*(12), 168781401880817. https://doi.org/10.1177/1687814018808170

Object Management Group. (2013). Business Process Model and Notation (BPMN) : Version 2.0.2. In *OMG*. https://www.omg.org/spec/BPMN/2.0.2/PDF

Ouyang, C., Dumas, M., Ter Hofstede, A., & van der Aalst, W. (2006). From BPMN Process Models to BPEL Web Services. In Proceedings of the 4th International Conference on Web Services (ICWS). IEEE Computer Society. https://doi.org/10.1109/icws.2006.67

Ouyang, C., Dumas, M., van der Aalst, W. M. P., Ter Hofstede, A. H. M., & Mendling, J. (2009). From business process models to process-oriented software systems. *ACM Transactions On Software Engineering And Methodology*, *19*(1), 1-37. https://doi.org/10.1145/1555392.1555395

Pla, A., Gay, P., Meléndez, J., & López, B. (2012). Petri net-based process monitoring : a workflow management system for process modelling and monitoring. *Journal Of Intelligent Manufacturing*, *25*(3), 539-554. https://doi.org/10.1007/s10845-012-0704-z

Raedts, I. G. J., Petkovic, M., Usenko, Y. S., Werf, van der, J. M. E. M., Groote, J. F., & Somers, L. J. A. M. (2007). Transformation of BPMN models for behaviour analysis. In J. C. Augusto, J. Barjis, & U. Ultes-Nitsche (Eds.), *Proceedings of the 5th International Workshop in conjunction with the 9th*

International Conference on Enterprise Information Systems (MSVVEIS 2007) 12-13 June 2007, Funchal, Madeira, Portugal (pp. 126-137). INSTICC Press.https://doi.org/10.5220/0002428801260137

Rational Software Architect 9.7.0. (2023, Septembre 21). IBM. Retrieved April 25, 2024, from https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-case

Raza, M. (2020, May 5). *Introduction to the Gartner Hype Cycle*. BMC Blogs. Retrieved April 27, 2024, from https://www.bmc.com/blogs/gartner-hype-cycle/

Recker, J., Muehlen, M. Z., Siau, K., Erickson, J., & Indulska, M. (2009). Measuring method complexity: UML versus BPMN. In 15th Americas Conference on Information Systems 2009, AMCIS 2009 (pp. 4444-4452). (15th Americas Conference on Information Systems 2009, AMCIS 2009; Vol. 7). https://www.academia.edu/2753843/Measuring_method_complexity_UML_versus_BPMN?email_work_card=view-paper

Reisig, W. (2013). The basic concepts. In *Understanding Petri Nets* (p. 13-24). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33278-4_2

Schäffer, E., Stiehl, V., Schwab, P. K., Mayr, A., Lierhammer, J., & Franke, J. (2021). Process-Driven Approach within the Engineering Domain by Combining Business Process Model and Notation (BPMN) with Process Engines. *Procedia CIRP*, *96*, 207-212. https://doi.org/10.1016/j.procir.2021.01.076

Sinha, Y. (2024, March 26). Future Trends in Process Modeling and Simulation. *Medium*. Retrieved April 28, 2024, from https://medium.com/@land_rover__parts/future-trends-in-process-modeling-and-simulation-ffcd1b754fbc

Stein, S. (2010, April 15). *EPC vs. BPMN - the perfect flamewar*. ARIS BPM Community. Retrieved April 5, 2024, from https://ariscommunity.com/users/sstein/2010-04-15-epc-vs-bpmn-perfect-flamewar

Team Kissflow. (2024, March 20). *BPMN - An Ultimate Guide to Business Process Model and Notion for 2024*. Kissflow. Retrieved April 3, 2024, from https://kissflow.com/workflow/bpm/what-is-bpmn/

The Agile Modeling (AM) Method. (2023, November 23). *UML use case diagrams: Diagramming Guidelines*. The Agile Modeling (AM) Method - Effective Strategies for Modeling and Documentation. Retrieved March 15, 2024, from https://agilemodeling.com/style/usecasediagram.htm

The Basics of documenting and analyzing your AS-IS Process. (2019, October 21). Lucidchart. Retrieved March 31, 2024, from https://www.lucidchart.com/blog/as-is-process-analysis

The UML Extensibility mechanism explained. (2023, September 13). Visual Paradigm. Retrieved March 31, 2024, from https://guides.visual-paradigm.com/the-uml-extensibility-mechanism-explained/

The use case diagram in Unified Modeling Language. (2020, July 24). IONOS Digital Guide. Retrieved March 15, 2024, from https://www.ionos.ca/digitalguide/websites/web-development/use-case-diagram/

Thong, W. J., & Ameedeen, M. A. (2014). A Survey of Petri Net Tools. In *Lecture notes in electrical engineering* (p. 537-551). https://doi.org/10.1007/978-3-319-07674-4_51

Top 6 online BPMN diagram Tools. (n. d.). Edrawsoft. Retrieved March 31, 2024, from https://www.edrawsoft.com/article/6-online-bpmn-diagram-tool.html

UML Use case Diagram - JavatPoint. (n. d.). Javatpoint. Retrieved April 5, 2024, from https://www.javatpoint.com/uml-use-case-diagram

Use case diagram tutorial. (n. d.). Visual Paradigm. Retrieved April 12, 2024, from https://online.visual-paradigm.com/diagrams/tutorials/use-case-diagram-tutorial/

Use case diagrams Unified Modeling Language (UML). (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

Vaisman, A. (2013). An Introduction to Business Process Modeling. In *Lecture notes in business information processing* (p. 29-61). https://doi.org/10.1007/978-3-642-36318-4_2

Valavanis, K. (1990). On the hierarchical modeling analysis and simulation of flexible manufacturing systems with extended Petri nets. *IEEE Transactions On Systems, Man, And Cybernetics*, *20*(1), 94-110. https://doi.org/10.1109/21.47812

van der Aalst, W. (1998). Three Good Reasons for Using a Petri-Net-Based Workflow Management System. In *Information and Process Integration in Enterprises. The Springer International Series in Engineering and Computer Science* (vol 428, p. 161-182). Springer, Boston. https://doi.org/10.1007/978-1-4615-5499-8_10

van der Aalst, W. , & Hee, K. (1996). Business process redesign : A Petri-net-based approach. *Computers In Industry*, *29*(1-2), 15-26. https://doi.org/10.1016/0166-3615(95)00051-8

van der Aalst, W., & Hee, K.M. (2002). Workflow Management: Models, Methods, and Systems. In *The MIT Press eBooks*. https://doi.org/10.7551/mitpress/7301.001.0001

van der Aalst, W., La Rosa, M., & Santoro, F. M. (2016). Business Process Management. *Business & Information Systems Engineering*, *58*(1), 1-6. https://doi.org/10.1007/s12599-015-0409-x

VanZandt, P. (2024, April 11). *What is the use case Diagram ? Definition, uses, examples, and UML use case diagram*. IdeaScale. Retrieved March 31, 2024, from https://ideascale.com/blog/what-is-the-use-case-diagram/

Veenendaal, A. (2023, Decembre 20). *The Future of RPA : 2024 Predictions*. SS&C Blue Prism. Retrieved May 6, 2024, from https://www.blueprism.com/resources/blog/future-of-rpa-trends-predictions/

Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Von Rosing, M., Scheer, A., Von Scheel, H., Svendsen, A. D., Kokkonen, A., Ross, A. M., Bøgebjerg, A. F., Olsen, A., Dicks, A., Gill, A. Q., Bach, B., Storms, B. J., Smit, C., Clemmensen, C., Swierczynski, C. K., Utschig-Utschig, C., Moorcroft, D., Jones, D. T., Coloma, D., . . . Dumond, R. (2015). Business Process trends. In *The Complete Business Process Handbook* (Vol. 1, pp. 187–216). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00011-2

Wang, Y., Wen, L., Yan, Z., Sun, B., & Wang, J. (2015). Discovering BPMN Models with Sub-processes and Multi-instance Markers. In *On the Move to Meaningful Internet Systems : OTM 2015 Conferences* (Vol. 9415, p. 185-201). Springer, Cham. https://doi.org/10.1007/978-3-319-26148-5_11

What is BPMN ? (n. d.) Visual paradigm.Retrieved March 31, 2024, from https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/

What is Business Process Modeling Notation?. (n. d.). Lucidchart. Retrieved April 15, 2024, from https://www.lucidchart.com/pages/bpmn#:~:text=Business%20Process%20Modeling%20Notation%2

What is Use Case Diagram ?. (n. d.) Visual Paradigm. Retrieved April 15, 2024, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

White, S. A. (2004). Introduction to BPMN. *BPTrends*. http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf

White, S. A. & Miers, M.(2008). *BPMN Modeling and Reference Guide: Understanding and Using BPMN* (Vol. 1). Future Strategies Inc, Lighthouse Point, Florida, U.S.A. https://books.google.be/books/about/BPMN_Modeling_and_Reference_Guide.html?id=0Z2Td3bCYW8C&redir_esc=y

Wicks, R. (2023, August 24). *Comparing different BPMN modeling tools*. Neilvw. Retrieved April 26, 2024, from https://www.neilvw.training/single-post/comparing-different-bpmn-modeling-tools

Widya, P. (2016). UML use case diagrams. *Pradani Widya - Academia.edu*. https://www.academia.edu/24342204/UML_Use_Case_Diagrams

Wolfram, G. (2022, May 4). *Formalize Your Domain Model With a Use Case Diagram*. OpenClassrooms. Retrieved March 29, 2024, from

https://openclassrooms.com/en/courses/6121861-apply-a-domain-driven-design-approach-to-development/6368261-formalize-your-domain-model-with-a-use-case-diagram

Wolter, C., & Schaad, A. (2007). Modeling of Task-Based authorization constraints in BPMN. In *Business Process Management. BPM 2007. Lecture Notes in Computer Science* (Vol. 4714, pp. 64–79). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75183-0_5

Wyatt, F. J. (2019, July 11). The Future of Process Excellence - On Business Process Management and Workflow Automation - Medium. *Medium*. Retrieved May 15, 2024, from https://medium.com/business-process-management-software-comparisons/the-future-of-process-excellence-3bb10e145139

Yaqin, M. A., Sarno, R., & Fauzan, A. C. (2017). Scalability Measurement of business process model using business processes similarity and complexity. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE. https://doi.org/10.1109/eecsi.2017.8239129

Yu, R., Huang, Z., Wang, L., & Zhang, H. (2014). Analyzing BPMN with Extended Object Petri Net. *Journal Of Software Engineering*, *8*(2), 58-74. https://doi.org/10.3923/jse.2014.58.74

Zbigniew, & Zbigniew. (2020, July 30). *Free BPMN modelling tools – 2020 edition*. BPM Tips. Retrieved May 3, 2024, from https://bpmtips.com/free-bpmn-modelling-tools-2020-edition/

Zhou, M., & Wu, N. (2010). System Modeling and Control with Resource-Oriented Petri Nets. In *CRC Press eBooks* (1st ed.). https://doi.org/10.1201/9781439808856

Zuberek, W. (1991). Timed Petri nets definitions, properties, and applications. *Microelectronics Reliability/Microelectronics And Reliability*, *31*(4), 627-644. https://doi.org/10.1016/0026-2714(91)90007-t

Zou, J., & Pavlovski, C. J. (2008). Control case approach to record and model non-functional requirements. *Information Systems and E-business Management*, *6*(1), 49–67. https://doi.org/10.1007/s10257-007-0057-x

Appendices

Appendix A: *Notes and transcript of the interviews with two UCL professors teaching modelling models.*

Appendix A.1: *Emmanuel Kolp, 18/04/2024, Teams, note-taking of the meeting.*

1. **Pouvez-vous nous parler des 4 modèles suivants (BPMN, PN, Use Case et Flowchart) et nous donner leurs principaux avantages/ inconvénients?**

Attention, UML, Petri Nets et BPMN sont des MODELES, pas des METHODES (quand on parle de méthode on parle plutôt de méthodes de gestion de projet informatique, par exemple: scrum et kanban)

PETRI NETS:

- Modèle très formel et fort basé sur la théorie.
- Utilisé en supply chain, chaîne de MARKOV.

BPMN:

- Ce modèle est arrivé dans les années 2000.
- Défauts: trop répandu, très peu formel.
- Avantages: développé par OMG (document officiel sur OMG.ORG/BPMN), lié à aucune méthode, on peut utiliser avec SCRUM, simple à apprendre.
- Modélise des flux.

Flow chart:

- Modèle du même niveau que le modèle BPMN mais il est moins riche que BPMN.
- Utile pour les database design, année 70, analysent comment les systèmes “communiquent” entre eux.
- BPMN ET FLOWCHART SONT ORIENTÉS WORKFLOW, analyse de business, par rapport à des comportements séquentiels, iteration, alternatives, parallels, tout ça ce sont des portes logiques.

UML:

- USE CASE est une autre famille (comme KAOS, HIGHSTAR,...), il existe pleins de modèles différents. On utilise ce modèle pour représenter les fonctionnalités (et souvent une fois à l'intérieur alors on utilise un diagramme entité).
- Le modèle BPMN est hors du modèle UML mais il s'agit du même organisme (OMG) qui a émis le modèle.
- Il y a une conjonction entre UML et BPMN (voir les versions). Ces modèles viennent des machines de ALAN TURING, qui sont des automates abstraits. Dans ces machines on a défini des cartes comportements de base, qui ont défini les comportements algorithmiques dans les années 30.
- Process avec étapes, par exemple des conditions pour un branchement (alternative) ou itération ou parallélisme.
- A partir de TURING, dans les années 50, il y a eu les STATES chart puis une flopée de diagrammes (petri net, flowchart,...). Ils sont tous assez semblables mais il y a une grosse différence: le niveau de formalité.
- Les state machines sont vraiment pour les informaticiens et développeurs.
- UML ne modélise pas des flux (flux c'est BPMN, flow chart).
- Modèle assez informel.
- Avec UML on fait une cartographie: chaque bulle représente une fonctionnalité qu'on va devoir informatiser.
- Il faut faire attention que ce sont des familles de modèles différents.
- UML est un standard ISO, donc très important d'être sur des standards quand on développe pour une entreprise.
- Se renseigner sur :
    - ISTAR: model plus poussé que UML qui possède des extensions
    - La méthode de gestion de projet RUP. Il y a une extension sur les use case. Il s'agit d'une méthode plus simple car elle ne comprend que des acteurs.

- Si on veut connaître TOUT UML, ça va demander beaucoup de temps car il y a beaucoup de modèle cela est moins accessible à tout le monde car certains sont complexes.
- Ne prendre que use case (et diagramme d'activité d'activité) dans notre mémoire, en faisant attention de bien prendre une optique business process modeling.

Comment modéliser un processus ? :

Toujours à l'aide du modèle BPMN. Il y a pleins d'éléments mais nous ne sommes pas obligés de tous les utiliser. On peut seulement en représenter certains: séquence, itération, concurrence, alternative. Tout programme, toute chaîne de production fonctionne comme ça.

SI on représente un système, BPMN et flowchart ne représentent pas la structure des data mais seulement des processus.

Comment choisir un modèle? :

Le choix dépend du niveau de formalité:

- Petri net est très formel alors que BPMN et flowchart ne le sont pas
- Les informaticiens préfèrent STATE MACHINE aux diagrammes d'activité car c'est plus formel. Par rapport à des perspectives productions, algorithmiques, c'est pas le top. C'est mieux quelque chose avec plus de formalisme.

Pour des consultant dans le développement de systèmes IT: UML avec state machine à l'intérieur.

Pour des consultant Odoo, SAP: BPMN

Critères pour choisir le modèle:

- Formalité
- Standardisation
- Situation:
    - Analyse de cahier de charge: BPMN
    - Suivre un processus de logiciel: toute l'enveloppe UML (use case, diagramme de classe, entité,...)

- intégration de gestion de projet

Appendix A.2: *Jean Vanderdonckt, 10/04/2024, Teams, transcription of the recording.*

- **Jean Vanderdonckt** : Le BPMN comme son nom l'indique il a plus pour objectif de représenter et de modéliser les processus d'affaires tandis que les cas d'utilisation en anglais, les use cases de UML, eux ont plutôt pour objectif d'exprimer l'interaction avec entre l'utilisateur final et le système ou bien aussi quelles sont les préférences, les souhaits, ce qu'on appelle aussi les requis de la part de l'utilisateur. Donc le BPMN lui n'exprime pas ça.

- **Luci** : Et vous avez aussi travaillé avec Petri Nets ou pas ?

- **Jean Vanderdonckt** : Oui oui alors je prends imaginons que vous êtes étudiantes et vous devez vous inscrire à l'université. Le BPMN, il va dire que premièrement il faut remplir un formulaire, ce formulaire sera soumis au secrétariat qui l'a validé ou pas, si le résultat est positif vous êtes notifié de la réponse et si le résultat est négative il sera transféré à une autre personne responsable pour traitement etc. Donc ça ça décrit quelle est en fait la gestion du flux dans le processus en fait depuis le début jusqu'à la fin mais il ne dit pas comment ça va être réalisé.

- **Luci** : C'est ça, ok.

- **Jean Vanderdonckt** : Alors que dans le UML use case là on va dire que l'utilisateur va d'abord recevoir un formulaire, ce formulaire va d'abord être découpé en morceaux, il va d'abord traiter ceci puis il va traiter ça puis cette partie-là sera comme ça. Donc ce sont des choses un peu différentes mais ils sont donc complémentaires.

- **Luci** : Je voulais avoir votre avis, j'ai vu pas mal de fois dans la littérature scientifique que par exemple Petri Net était utilisé pour vérifier l'exactitude d'un BPMN par exemple vérifiez si le fonctionnement…. s'il y avait pas de deadlock etc.

- **Jean Vanderdonckt** : Alors oui, un réseau de Petri est beaucoup plus précis qu'un modèle BPMN. Je sais pas si vous êtes familière avec la notation BPMN ? Une référence que je donne c'est souvent une université allemande qui a fait un très très bon poster là-dessus, qui résume toute la notation. Je sais pas si vous l'avez, je vais le chercher, un instant si vous le permettez. Je pense que si on tape BPMN version 2.0….Et c'est un poster très visuel avec un exemple. Oui mais voilà mais c'est de Berlin. Je vous mets le lien dans la zone de conversation, c'est bien ça oui c'est ça ce poster est évidemment très bien parce qu'il a…. attention c'est compliqué sur ce lien-là, vous allez télécharger un un poster au format PDF qui donne un résumé visuel total de la notation BPMN. Donc c'est l'université de Berlin qui fait ça et qui maintient ça à jour, donc vous avez tous les éléments de la notation, les détails, les tâches, les transactions et puis tous les types d'événements. Oui vous voyez qu'il y en a un paquet hein quand même.

- **Chloé et Luci :** Oui Oui

- **Jean Vanderdonckt** : Alors en majeure partie évidement ça dépend qui l'utilise. Quand c'est quelqu'un qui est formé au BPMN oui il peut être très précis. Vous avez un exemple ici au milieu. Mais la plupart du temps, lorsque ce sont des gens qui ne sont peut-être pas habitués ou pas experts ou qui ne le souhaitent pas, ils gardent des diagrammes BPMN avec les objets les plus simples ou les plus courant parce que c'est plus simple à communiquer avec les utilisateurs finaux. Mais alors évidement c'est une notation donc on ne vérifie rien quant à l'exactitude, la correction, la complétude, il y a plusieurs critères de qualités, donc on ne vérifie rien quant aux schémas. Pour ça il faut faire appel à des outils logiciels, il y en a qui permettent de vérifier que tout est correcte dans la notation même, il y en a d'autres qui disent qu'il faut transformer en Petri Net, c'est pas nécessaire mais ça existe aussi.

- **Luci :** Ok. Et alors est-ce que peut-être vous saurez euh nous éclairer là-dessus : Pourquoi si on dit que le pétri peut nous aider à corriger les erreurs BPMNs ou vérifier son exactitude pourquoi ne pas directement passer par un réseau Petri net pour modéliser ?

- **Jean Vanderdonckt** : Alors il existe des outils logiciels qui permettent de vérifier, donc y a 3 facteurs hein, il y a la correction, la complétude et la cohérence. Ce sont les 3 grands critères de qualité, il y en a d'autres mais ce sont les principaux. Alors il existe des outils qui vérifient ces propriétés là sur le modèle BPMN directement. Mais alors il faut

introduire votre BPMN dans cet environnement-là, c'est pas juste le dessin parce que le dessin je peux faire n'importe quoi à la limite. Si je prends un outil de dessin je peux faire n'importe quoi, y compris des choses qui sont incorrectes incomplètes ou incohérentes. Alors, dans certains outils logiciels on peut vérifier ça sans devoir transformer dans un autre format comme les réseaux de Pétri. Mais on ne peut pas tout vérifier parce qu'en effet dans la notation BPMN c'est une notation qui ne précise pas tout. Au contraire dans un réseau de Petri c'est extrêmement précis mais c'est aussi très complexe. Don le niveau de complexité est baeucoup plus grand pour un réseau de Pétri que pour un modèle BPMN. Ça dépend ce qu'on veut en faire, si, moi par exemple quand on veut discuter avec des gens eans des entreprises, des organisations sans rentrer trop dans les détails je préfère montrer diagramme de flux ou un modèle. Ça ils comprennent assez vite. Il faut que ça soit communicable, il faut que ça soit compréhensible sans trop rentrer dedans. Et alors après, une fois que ça a été validé ou communiqué avec les gens qui travaillent effectivement dans ces parties-là après on peut les préciser, les corriger, les rendre plus précis etc. Mais alors de prime abord je ne montre jamais un réseau de pétri à un utilisateur final. C'est trop complexe c'est trop détaillé. Mais par contre voilà l'avantage c'est qu'une fois qu'on a la première partie bah on peut passer à une partie plus heu plus formelle, plus détaillée.

- **Luci :** Ok, et vous heu parliez des diagrammes de flux là juste avant en fait on cherche une 3e donc une 4e méthode plutôt, heu à comparer, on pensait au diagramme de séquence…. Je ne sais pas, vous qui connaissez pas mal heu…

- **Jean Vanderdonckt** : Oui, oui c'est aussi un des 8 modèles de UML hein dans uml ou unified modeling notation c'est en fait une heu une série, en fait une suite de 8 modèles différents qui expriment des choses différentes. Certaines sont liées, d'autres sont tout à fait indépendantes. Alors dans le activity training, là vous pouvez aussi en fait exprimer aussi un enchaînement de tâches, d'actions, alors en général on le fait du point de vue du système et on peut faire du point de vue de l'utilisateur.

- **Luci** : OK.

- **Jean Vanderdonckt** : Alors celui-là en général , de nouveau hein ça dépend l'objectif, mais quand c'est pour heu travailler, communiquer avec les futurs utilisateurs du sytème, je ne montre pas celui-là parce qu'il est de nouveau déjà plus orienté vue system. Donc je

préfère montrer soit un BPMN ou alors un diagramme des flux. En ne disant alors je en garde que quelques heu que quelques symboles, je peux vous en montrer un. Vous avez suivi mon cours de BAC 3 ?

- **Luci :** Oui.

- **Jean Vanderdonckt** : Donc voilà, si vous vous souvenez le quatrième et dernier modèle, le diagramme des flux est un modèle qui est facile à comprendre par les gens hein il y a essentiellement, vous avez des traitements hein un block, ça peut être une activité, une tâche, comme dans BPMN et puis bah vous avez des messages qui sont en fait des sont échangés entre les différentes unités. Don ce diagramme là en général les gens comprennent assez vite. Et peuvent surtout réagir, donc ce qui est important c'est de valider le schéma pour dire oui c'est bien comme ça que ça se passe. Mais il est moins précis, le diagramme des flux est plus facile que le BPMN mais il est moins précis que le BPMN. Le BPMN peut exprimer plus de choses. Hein par exemple, lorsqu'il y a des branchements si on va par là, par là. Tandis que dans le diagramme des flux, on peut aller là, là, ou là mais on dit pas pourquoi. En fait ce que vous pourriez faire, cest faire un schéma sur 2 axes hein, sur l'axe des x vous avez par exemple le niveau d'expressivité et sur l'axe des y vous avez le niveau de complexité. Donc vous avez une courbe comme ça, plus c'est expressif ben plus c'est complexe évidemment mais moins c'est communicable et compréhensible par quelqu'un qui n'est pas formé. Don sur l'axe des x si vous mettiez l'expressivité ben ceux qui sont les moins expressifs ce serait par exemple le diagramme des flux et le plus exresssif ou le plus détaillé de l'autre côté ce serait le réseau de Pétri. Mais la complexité d'un côté est très basse et d'un autre côté est très élevée. Et vous auriez BPMN qui est un peu au milieu.

- **Luci** : Ok, est-ce que vous sauriez comme ça, vu que vous êtes heu que vous les connaissez un peu tous et que c'est très rare, vous êtes la première personne à qui on parle qui connaît toutes ces méthodes. Heu, si vous deviez citer en tout cas les avantages et inconvénients principaux de ces méthodes ?

- **Jean Vanderdonckt** : Alors l'avantage mais qui est aussi, c'est vraiment les deux faces d'une pièce, l'avantage c'est qu'on exprime de manière abstraite certains aspects d'un système informatique avant même de e développer. Donc ça veut dire qu'on le représente, et alors c'est visuel donc tout le monde peut y accéder, peut regarder, peut corriger, peut

commenter, donc ça c'est un avantage mais ça oblige, comme c'est un niveau d'abstraction, ça oblige toujours à changer sa perspective et d'arriver à un niveau d'abstraction un peu plus élevé que travailler dans le concret. Je prends un exemple si je dis heu voilà vous devez vous inscrire comme étudiante heu à l'université bah pour ça il y a un processus à faire, il y a des formulaires à remplir. Si on est très concret, on dirait : bah vous avez juste ce formulaire là à remplir et c'est tout. Par contre ces modèles là on va dire il faut extraire les données, il faut représenter les données, comment est-ce qu'elles seront structurées, quelles données vont être échangées quelles sont-elles qui vont être traitées. Et donc ces modèles-là ont l'avantage d'être indépendants de toute implémentation hein lorsque vous êtes inscrite, je sais pas si vous êtes inscrites en ligne directement, vous ne vous occupez pas du fait que le formulaire est en html, en java script, ou en tout ce que l'on veut, hein l'utilisateur final il s'en fou de savoir quelle base de donnée derrière il y a hein est-ce que c'est le format x, y ou z, on s'en fou. Et donc ces modèles là ont pour objectif de s'abstraire de l'implémentation. Ah c'est pour ça que par exemple si je dis à quelqu'un, je pourrais très bien demander à un développeur : implémentez moi ça avec telle base de données, tel langage ou tel environnement, il pourrait le faire mais alors ça sera très spécifique. Par contre si je dis avant, bah j'essaye de faire mon BPMN, bah alors là je valide ça avec la personne concernée et puis après on peut commencer l'implémentation mais c'est indépendant de l'implémentation. Donc après si il le fait avec le système A, le B ou le C, ce sera toujours le même modèle. Le modèle ne change pas.

- **Luci** : ok et est-ce que vous savez donné un point fort et un point faible de BPMN, UML et Petri Nets par exemple ? Les points qui les caractérisent vraiment.

- **Jean Vanderdonckt** : Oui alors heu on peut évidement, comment dirais-je, on peut les comparer suivant différents critères je vous avais dit. Bon le premoier critères c'est d'abord l'objectif, c'est qu'est-ce qu'ils font ? Le BPMN ça sert surtout pour modéliser les processus d'affaire et les workflows. Tandis que UML c'est plutôt pour les requis de l'utilisateur, l'intéraction, heu, les préférences de l'utilisateur. Tous les deux ils ont toujours un plus haut niveau d'attraction puisqu'ils sont indépendants de l'implémentation mais BPMN est plus haut en abstraction que les cas d'utilisation. Tous les 2 utilisent des éléments évidemment ils ont de notation visuelle avec des éléments qui sont définis des symboles, des règles. Heu, les use cases dans UML bah il y en a beaucoup moins, hein ou

bien dans le diagramme des flux si vous vous souvenez bah il y a juste essentiellement 2 grands concepts ou avec des variations hein. On peut dire si un traitement est automatique, interactif, manuel ou autres. Mais du coup comme la notation est moins compliquée c'est plus facile, c'est plus accessible pour quelqu'un qui n'est pas habitué. Et puis aussi bah bon aujourd'hui le support en termes d'outils logiciel est à peu près le même dans tous les cas mais de nouveau leur expressivité n'est pas la même. Donc voilà l'avantage c'est que ça reste indépendant de l'implémentation mais l'inconvénient c'est que ça demande toujours un effort de comprendre la notation, de maitriser l'abstraction et de la manipuler. Donc voilà il y en a qui ont plus de difficulté que d'autres. C'est pour ça que moi j'enseigne en UML le diagramme de flux parce que ça reste indépendant de toute implémentation donc si c'est par exemple voilà vous êtes gestionnaire, si un jour vous devez travailler, c'est un exemple mais je connais quelqu'un qui est sorti avec un diplôme de la lsm en tant que ouais, avec un master en gestion, et donc cette collègue, qui est une amie personnelle, elle n'a jamais développée de sa vie, par contre elle a a travaillé avec ce genre de modèle pour dire : bah voilà ça c'est le contrat, une fois que ça c'est entre le l'utilisateur et la firme, une fois que les modèles sont exprimés et validés on dit ok c'est bon, après on les envoie aux développeurs, au développement qui va développer et qui va prendre ça comme base. Zt ça c'est toujours le même. Donc si un jour je donne mon BPMN en Belgique, en Europe, aux États-Unis, en Asie ou en Afrique peu importe, toute personne qui, puisque c'est un langage international, toute personne qui comprend les BPMNs va pouvoir faire la même chose. Par contre si moi j'utilise ma propre notation, mon propre langage, on peut le faire aussi, on en a eu, bah y a que moi qui vais el comprendre, évidement un peu limité. C'est toujours, l'avantage c'est comme c'est abstrait par rapport à l'implémentation, c'est indépendant et dès que c'est abstrait bah il faut rentrer dans le langage, dans la notation, dans les éléments et donc savoir ce que ça exprime, ce que ça n'exprime pas. Donc ça demande de la pratique.

- **Luci :** Ok et donc ça pour vous uml est plus simple à comprendre que BPMN vous avez dit ?

- **Jean Vanderdonckt** : Alors il y a 8 modèles mais le modèles des cas d'utilisation, en anglais use case dans UML, celui-là est plus simple. Donc il est effectivement plus simple que BPMN.

- **Luci** : Et le use case pour vous c'est celui qui se rapproche le plus de BPMN enfin de l'utilisation du BPMN ? Qu'on pourrait lieux comparer ?

- **Jean Vanderdonckt** : Oui, en termes d'objectif oui.

- **Luci :** ok ok. Chloé je crois que t'avais des question non pour heu ce que tu avais lu ?

- **Chloé :** Oui donc pour comparer, on cherche une quatrième méthode, vous nous conseillez plus de comparer avec les diagrammes de séquence ou les diagrammes de flux ?

- **Jean Vanderdonckt** : moi je vous propose le diagramme des flux parce que c'est de nouveau, ça reste simple et donc comme c'est simple c'est plus largement utilisable. Heu les diagrammes d'activités sont déjà plus expressifs, un peu plus compliqué mais ça reste encore relativement bien utilisable dans les organisaitons hein mais heu la pratique montre que plus c'est simple, plus c'est communicable et donc plus les gens peuvent réagir. C'est pour ça apr exemple que les réseaux de Pétri c'est extrêmement précis, extrêmement détaillé donc ça c'est très bien en termes voilà hein, en terme de qualité. Mais avant que quelqu'un qui n'est pas formé à ça rentre dedans et puisse me dire : ah ça c'est juste, ça c'est pas juste, ça il faut faire comme ça ou comme ça, ça va prendre énormément de temps.

- **Luci et Chloé :** Ok.

- **Jean Vanderdonckt** : Donc voilà si on veut si on veut avoir un retour rapide du client, on ne montre jamais un réseau de Pétri, parce que de prime abord c'est complexe. Bon on peut les simplifier hein dans un autre cours, je vais peut être ttendez, je vais vous donner le lien sur Moodle. Attendez deux secondes, je cherche le lien. Si vous le souhaitez vous pouvez aussi, l'accès est évidement ouvert. J'ai un autre cours dans lequel j'enseigne les workflows et donc là j'utilise les réseaux de Pétri mais seulement une partie.

- **Luci :** On pourrait avoir accès à celui là ?

- **Jean Vanderdonckt** : Oui oui attendez je cherche le lien et que comme ça vous pourrez télécharger les notes heu de cours. C'est le cours LSINF2382, ah le voilà. Donc je vous mets le lien dans la zone de conversation.

- **Chloé** : Merci.

- **Jean Vanderdonckt** : Donc dans ce cours là enfait j'utilise les réseaxu de Pétri pour modéliser les workflows. Mais ça c'est un cours en informatique.

- **Chloé** : Oui oui.

- **Jean Vanderdonckt :** Donc vous pouvez regarder dedans comment ça fonctionne, il y a plein d'exemples. Donc c'est très précis, mais c'est manipulé seulement par des personnes dont la formation est en informatique. Je donne pas ça à un utilisateur final par exemple sinon il va dire : ouhlala. Qu'est ce que c'est ?

- **Chloé :** Ok super, est-ce que selon vous il y a vraiment des thèmes que nous devons absolument aborder dans notre mémoire ?

- **Jean Vanderdonckt :** C'est toujours, quel est le gain, d'abord quel est l'objectif hein de ses méthodes ou modèle ça hein qu'est ce qu'on veut faire d'abord donc premièrement. Et puis deuxièmement, c'est toujours un rapport entre le coût et le bénéfice. Un diagramme des flux aura un bénéfice très bas mais un bénéfice modéré. Par contre, si je prends un modèle BPMN, le cout sera beaucoup plus important mais le bénéfice sera plus élevé aussi. Donc on peut aussi évoluer, enfin ça dépend de ce qu'on veut. Si je n'ai que quelques jours pour faire, pour avoir un retour chez le client, bah en fonction du nombre de jour je vais prendre tel ou tel modèle. Evidemment ma préférence ira pour un BPMN parce que ça c'est plus précis quand même, c'est un bon milieu entre le diagramme des flux qui est très simple mais peu expressif et le réseau de Pétri qui est très détaillé mais aussi très précis donc complexe, au milieu le BPMN est un bon compromis. Il exprime quand même déjà pas mal de chose. Et puis comme vous avez vu dans le poster que je vous ai montré, bah je ne suis pas obligé non plus d'utiliser tous les symboles.

- **Luci :** Oui ça on a remarqué que les consultants le faisaient surtout à leur manière.

- **Jean Vanderdonckt** : Oui alors ça c'est l'avantage et l'inconvénient. L'avantage c'est qu'il est relativement flexible mais l'inconvénient c'est que un modèle BPMN produit par quelqu'un n'a pas toujours toutes les qualités requises comme la complétude, la correction, la cohérence ou d'autres. Pour ça il existe des règles mais si la personne qui a rédigé le schéma, le modèle, ne suis pas les règles bah ça on peut rien n'y faire évidement.

- **Luci** : est-ce que du coup Uml, Petri net, bon bah du coup les diagrammes je suppose que oui, mais ce que les autres méthodes sont du coup selon vous plus facile à respecter leur sémantique et qu'elle soit du coup vraiment standardisé ?

- **Jean Vanderdonckt** : Non, c'est la même chose. Je prends l'exemple des diagrammes de flux. Oui au cours si vous vous souvenez j'ai vu qu'il y avait des règles de complétude, de cohérence etc mais si quelqu'un dessine juste un diagramme des flux, il peut très bien avoir un diagramme des flux qui ne respecte pas ces règles là, c'est pas grave mais enfin ça exprime quand même des choses. Et j'ai vu des gens qui le faisaient assez vite pour avoir au moins un retour. Mais si après le schéma, le modèle comporte des erreurs ça c'est une autre affaire. Parfois l'objectif n'est pas juste d'être extrêmement précis, c'est d'avoir un moyen de communication entre l'utilisateur final, fin entre vous, la personne qui veut réaliser la spécification, la description et l'utilisateur final. Qui est le ou la promotrice de votre mémoire ?

- **Luci** : Monsieur Per Agrell. Chloé je sais pas si t'avais encore des questions ?

- **Chloé** : Juste peut-être est-ce que vous avez également des exemples de modélisation UML et BPMN ?

- **Jean Vanderdonckt** : Heu oui ça j'ai beaucoup d'exemple mais ça dépend de nouveau qu'est-ce que vous souhaitez en faire ?

- **Chloé :** C'est plutôt les analyser pour montrer dans tel cas on utilise plus souvent les BPMNs et dans tel cas plus souvent UML. Et analyser à quel point ça va être précis dans l'utilisation des symboles.

- **Luci :** Par exemple, un cas qu'on a beauocup trouver, en tout cas on a des exemples en BPMN et en Pétri net c'est les gestion de plaintes de clients par exemple. Je ne sais pas si ça pourrait se faire en UML par exemple ?

- **Jean Vanderondckt** : Oui si un même problème peut être exprimé par ces différents modèles mais de nouveau ils exprimeront sur des aspects différents soit semblable mais avec plus ou moins de précision.

- **Luci** : Mais du coup c'est ça qu'on aimerait trouver, des situations types dans les 3 modélisations.

- **Jean Vanderdonckt** : Ah oui donc idéalement ce qu'il faudrait c'est un même problème mais qu'il soit représenté dans les différents modèles. Oui donc ça je n'ai pas mais je devrais peut-être pouvoir trouver attendez. Si on regarde heu… Vous avez déjà recherché sur internet pour voir s'il n'y a pas ça ?

- **Luci :** Oui oui on est en plein dans nos fouilles.

- **Jean Vanderdonckt** : Oui comme ça je n'en trouve pas, un même exemple qui traiterait enfin non, différents modèles mais qui traiteraient du même prolème avec les modèles respectifs. C'est possible de trouver certainement mais il faut chercher. Si vous tapez BPMN versus UML Use Case ou bien UML Activity diagramme je pense que vous trouverez certainement. Ou bien BPMN versus block diagrams. Block diagram c'est diagramme des flux en anaglais. Ou dataflow diagrams mais bon parfois, il y a déjà des noms différents parfois, sauf pour UML, BPMN où là c'est standardisé, mais parfois pour un même types de schéma il y a des noms différents en anglais. Parfois il y en a qui utilisent block diagram, parfois il y en a qui utilisent data diagram, d'autres disent data flow diagram.

- **Jean Vanderdonckt** : Ok génial, merci beaucoup. Sinon n'hésitez pas aussi à interroger mon collègue et voisin le Professeur Manuel Kolp, si ce n'est pas déjà fait.

- **Luci** : Oui on l'a déjà contacté.

- **Jean Vanderdonckt** : Il en utilise beaucoup aussi. Petri nets je ne sais pas mais BPMN et UML oui en tout cas.

- **Luci** : Ok on va voir avec lui alors. Merci beaucoup !

- **Jean Vanderdonckt :** Je vous en prie, bonne suite pour votre mémoire alors.

- **Chloé :** Merci beaucoup, au revoir.

- **Luci** : Au revoir.

- **Jean Vanderdonckt** : Au revoir.

Appendix B: *Notes taken during the interviews with Headmind Partner Brussels consultants familiar with business process modelling.*

Appendix B.1: *Mike Staudt, 25/02/2024, Teams, note-taking of the meeting.*

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission 1:

Développer une application sur base d'un processus opérationnel.

Client:

KEYTRADE

But et démarche:

Développer une nouvelle application (MENDIX) pour remplacer les processus opérationnels et les automatiser dans le but qu'un employé administratif puisse avoir accès à l'application.

Par exemple: un client veut ouvrir un compte, les employés doivent faire différentes choses pour créer ce compte. Ils utilisent Excel mais ce n'est pas facile de bosser à plusieurs via Excel. Il y a donc plusieurs problématiques car ce n'est pas standardisé!

Étape 1: Regarder tous les processus:

- Exemple de processus: Enquête judiciaire

  AS IS: processus par mail.

  TO BE: processus automatisé dans une application .

  Pour y arriver: Tout un processus est mis en place, création de BPMN pour voir quel est le processus exact dans cette situation. Sur base de ces workflows on les standardise pour que ce soit + facile à installer dans une application.

Mission 2:

Projet de restructuration.

Client:

BNP

But et démarche:

Pas de modèle BPMN tel quel, mais quelque chose qui y ressemble.

Différents départements sont imbriqués les uns dans les autres —> Cross départements (5-6 départements).

Appendix B.2: *Nicolas De Hertogh, 04/03/2024, Teams, note-taking of the meeting.*

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission:

Business analyst pour la création d'ERP pour les filiales de Proximus

Client:

PROXIMUS

But et démarche:

Différents pôles interviennent: achat, stockage, vente

Exemple de processus: Procure to Pay Process

1. identification du flow, quelles sont les étapes quand une filiale de Proximus veut acheter du matériel (achat, stock, vente), facturer des heures de consultation (achat, vente), payer un loyer (achat, vente),…

2. création d'un flow/BPMN qui ne suit pas la sémantique classique, BPMN simplifié car tout le monde n'a pas les notions exactes de BPMN et donc faut que ça soit compréhensibles pour tous, BPMN très visuel

Démarche:

1. Parler avec les stakeholders pour voir le rôle de chacun dans le processus : ce qu'ils faisaient, pourquoi et comment ils le faisaient.

2. Définir la/les conditions, ex: si order>15 000, alors c'est Mr X qui va s'en occuper,…
3. Ensuite, il utilise dynamics 365 comme ERP et implémente les flow qu'il a identifiés.

Objectif de cette démarche: pouvoir expliquer à toute l'entreprise le rôle de chacun.

Ça a pris des mois pour comprendre tous les processus/ personnes impliquées. En plus, on attendait de lui de comprendre tout ça et de le représenter donc BPMN était le début mais aussi le cœur de sa mission.

Le modèle BPMN a donc permis de construire la base pour avoir une idée claire en tête de comment le processus doit être dans l'application mais aussi pour s'assurer que toutes les autres parties soient d'accord avec le fonctionnement de l'entreprise tel qu'il l'a analysé.

Pourquoi l'utilisation du modèle BPMN?

1. Demande de Proximus pour construire l'ERP.
2. Le modèle BPMN a aidé Nicolas à comprendre le business process de Proximus.

Limites de la modélisation BPMN:

- Modélisation très coûteuse en temps si on fait sur un ordi mais à la main c'est rapide.
- Difficulté de la compréhension (certains sont en frein à apprendre un nouveau langage).
- Le client veut que ça soit rapidement fait, peu importe les réelles significations des symboles (alors que les business analysts y font peut être plus attention).
- Pas rigoureux, sémantique pas toujours respectée car l'objectif est d'être le plus clair et le plus accessible à un grand nombre de personnes

Avantages de la modélisation BPMN:

- Rapide
- Facile à faire et à comprendre
- Aide à comprendre le fonctionnement d'une entreprise de manière claire

Appendix B.3: *Sophie Van der Sanden, 30/04/2024, Teams, note-taking of the meeting*

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission:

Business analyst chez Telenet afin de comprendre comment gérer les IT business and services (elle a aussi travaillé chez proximus, il s'agissait de 2 missions dans le même domaine).

Client :

TELENET - PROXIMUS

But et démarche:

Comprendre comment gérer les IT business and services.

AS IS: Si un customer a un problème, il téléphone et il va recevoir un ticket (CATALYS: création du ticket, assignation du ticket,… le ticket passe à travers un processus).

TO BE: Implémenter ce processus dans un tool.

Pour y arriver: Pour "schématiser" le processus, elle utilise BPMN. Telenet veut certifier cet outil, pour cela il doit créer un processus en BPMN avec les risques etc.

Pourquoi l'utilisation du modèle BPMN?

Selon elle, dès qu'on parle de business process il faut utiliser BPMN. UPar exemple, UML, différentes manières de l'utiliser sont possible mais c'est plus informatif que business process tel quel, c'est plus formel et pour représenter les users,…

Elle utilise BPMN quand elle doit consulter les processus des compagnies. Parfois elle utilise d'elle même mais elle n'est pas process analyst donc ce n'est pas pour l'entreprise.

Il y a 3 différents outils pour modéliser BPMN car c'est un langage, donc plusieurs langages sont disponibles dont ARIS, celui qu'elle a utilisé dans ces deux missions.

Limites de l'utilisation du BPMN:

BPMN n'est pas utilisé de la bonne manière par les autres consultants parce que c'est easy to use mais il y a des advanced symbols en BPMN qui aide à mieux comprendre. Mais les consultants n'utilisent que les symboles de base car ils ne veulent pas chercher à faire compliqué pour que ça soit simple et compréhensible par tout le monde.

Cependant, elle a toujours su se faire comprendre au travers de ses modélisations.

Appendix B.4: *Léo Collet, 15/03/2024, Teams, note-taking of the meeting.*

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission:

Faire évoluer un processus dans un outil CRM.

Client:

ENGIE

But et démarche:

Chez Engie dans le domaine du service client, le but est de recourir à la modélisation pour faire évoluer un processus dans un outil CRM situation client. Développement d'un outil de téléphone de service client, accompagnement du client lors de la navigation sur le site. Il n'y a pas d'automatisation mais il faut normaliser le processus pour que tout le monde comprennent.

Les diagrammes BPMN ont été utilisés pour avoir un support IT afin d'expliquer les processus et les changements aux métiers.

Pourquoi l'utilisation du modèle BPMN?

- BPMN est la SEULE méthode de modélisation qu'il connaît. Il utilisait les icônes basiques mais pas de façon totalement rigoureuse, sur base de ce qu'il a étudié car pas non plus des choses très compliquées à modéliser, seulement des choix et actions
- BPMN est souvent utilisé en service client car souvent ceux externes ont un délai de traitement et de réponse et donc besoin d'être rapide et efficace grâce à une démarche standardisée et comme externe ils connaissent pas tous et assez arborescent.
- Il a aussi utilisé les diagrammes de séquences.

Limites de l'utilisation du modèle BPMN:

- Si ça avait été des objets plus complexes il aurait simplifié les schémas (pas utilisés toutes les nuances qui existent dans les icônes) comme Sofie disait car tout le monde n'aurait pas compris (surtout côté métier).
- Les gens (consultants et clients) manquent de connaissance dans les BPMNs alors que la demande est importante car de plus en plus de clients sont dans l'optique d'optimisation des coûts et donc des processus.

Solutions aux limites proposée par Nicolas:

Il pense qu'une formation devrait être donnée aux clients et consultants. Les consultants devraient être calés bien plus là dessus et pouvoir expliquer au client de façon un peu plus complète ce qu'est le BPMN pour qu'ils puissent mieux comprendre la complexité de leurs processus (grâce au schéma) et de voir là où on peut simplifier et comment (montrer que telle boucle dans le schéma est trop complexe et doit être transformer en deux branches par exemple afin d'optimiser leur processus, et donc expliquer ce que ce changement implique et comment y arriver, sauf que client va rien comprendre à cette explication si pas de formation en BPMN). Pas besoin de forcer la formation pour tous mais lorsque le projet en a besoin faire venir un senior 2h chez le client pour expliquer tout ça. Ou, si besoin, proposer un forfait sur une semaine ou on forme le client avant le début de la mission. Cela permettrait au client d'être plus investi dans le projet et de mieux comprendre la nécessité des changements.

Personnes à contacter selon lui :

- Alexandre Beltran (déjà contacté) et Dimitri Choux bossent ensemble sur business performance en business process management donc connaissent surement d'autres méthodes et/ou d'autres consultants et missions

- Aurélia Nilly senior associate digital france : Mission chez total avec bcp de process donc utilisé peut être d'autres méthodes
- Nicolas Martineau: Orange Belgique projet DETOX, diminuer le nombre d'appels au service client qui auraient pu être des demandes avec réponses self care, FAQ. Checker ce qui a été fait dans cette mission car possible qu'il y ai du processing modeling. Il connaît la mission et ceux qui ont bossé dessus.

Appendix B.5: *Ugo Watelet, 18/03//2024, Teams, note-taking of the meeting*.

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission:

Gestion des rejets de domiciliation.

Client:

TOTAL

But et démarche:

Il a fait un BPMN qui était au niveau de la gestion des rejets de domiciliation. Quand on facture des clients, il y a un envoi de fichier qui dit de combien on doit débiter les clients. Cela est automatique. Pour les clients où on ne sait pas prélever, il y a plusieurs raisons (il a dit analyser les raisons, et envoyer des mails automatiques)

Objectif: schématiser les options disponibles, regarder les familles de rejets (même famille, même problème) et voir les réponses automatiques à ces rejets que le client va recevoir.

Description du processus:

- Point de départ du processus: reçu de la banque
- Reçu spécifique, 4 familles de rejets (ex: client décédé, pb technique, solde insuffisant,…)

- Ensuite pour chaque famille de rejet au sein de Total, ils utilisent SAP. Le SAP dit si le client est bloqué à la relance, donc tout va être bloqué quand le client ne paye pas et activer un processus qui est le rejet de domiciliation.
- Si data incorrecte: annuler la domiciliation et passer le client en virement. On clôture le mandat avec la banque et on envoie au client un email pour activer sa domiciliation en proposant un lien pour payer la facture. Cet email met aussi en garde d'un frais de rappel s' il ne paye pas à nouveau.
- En cas de problème client, solde insuffisant,… : d'abord on avertit le client 1 fois, on le bloque pendant 5 jours dans tous les autres process. Une fois qu 'il a un mail d'avertissement, il doit payer dans les 5 jours. Si de nouveau il ne paye pas dans les 5 jours alors il passe en virement (voir ci-dessus).
- Enfin on envoie soit un mail, soit un sms en fonction de la réponse du client.

<u>Pourquoi l'utilisation du modèle BPMN?:</u>

- Parce que c'est le standard dans la boîte (TOTAL) d'utiliser ça.
- Ugo ne voit pas quel autre modèle utiliser pour ce genre de processus (word, powerpoint?)
- BPMN n'est pas compliqué à utiliser et à faire.
- TOTAL utilise tout le temps ce modèle pour schématiser des b2b, de base il avait utilisé Powerpoint mais plus dur et c'est plus resumé,…
- Il a aussi utilisé FIGMA (utiliser pour des processus quand tu te balades sur internet, sortes de Powerpoints qui sont connectés entre eux). C'est quelque chose de plus visuel et moins schématique.

<u>Connaissance du BPMN par le consultant et le client:</u>

- Chez Total ils utilisent tous ce modèle et le comprennent tous.
- Total propose peut-être des formations mais Ugo a dû voir par lui-même en revoyant notamment ses rappels de l'université pour être sûr.

Appendix B.6: *Quentin Halsberghe, 18/03//2024, Teams, note-taking of the meeting*.

1. **Parlez-nous des missions au cours desquelles vous modélisez des processus, dans quel but, quel modèle?**

Mission:

Description des processus internes d'une petite entreprise.

Client:

FIDEA (société d'assurances qui n'existe plus car elle a été rachetée)

But et démarche:

FIDEA est une petite compagnie qui était basée à Anvers et n'avait aucun processus décrit. Donc ils ont fait un gros track pour décrire tous les process internes. L'idée était de modéliser pour améliorer les processus. Quentin est arrivé vers la moitié du projet, il a dû prendre certains processus et les modéliser. Il a donc dû regarder à la modélisation et l'amélioration des process.

Démarche:

- D'abord il allait dans le département et voyait avec les gens qui exécutent le processus en quoi ça consistait.
- Ensuite il faisait des schémas, cartes, BPMN pour définir les chemins, analyser les pb,..
- Après, il retournait dans le département pour montrer l'analyse qu'il avait faite (ex: checker timer,..) et voir si c'était comme ça que ça se passait.
- Ensuite, grâce à ces modélisations, il a pu voir ce qu'il devait améliorer, comment le faire etc. Une fois les schémas faits, il a identifié les étapes ou on pouvait prendre action. Soit il refait tout le processus ou soit juste une partie, pas toujours une amélioration de TOUT le processus, ça dépendait.

Il a surtout étudié le processus 'incident management' car FIDEA était en lien avec un tiers qui gérait leur résolution d'incident mais le processus n'était pas hyper bon.

Pourquoi l'utilisation du modèle BPMN?:

- Dans ses autres missions il a toujours été confronté à des modèles déjà faits, n'a plus dû en créer lui-même. Les modèles qui étaient déjà faits étaient toujours des BPMNs, il n'a jamais vu une autre méthode.
- Selon lui toutes les entreprises modélisent quasi que avec des BPMN (voir tous les software qui sont créés pour modéliser des processus, ils utilisent toujours BPMN).
- L'entreprise n'a pas demandé le BPMN en particulier, il a juste utilisé la méthode la plus simple.
- Méthode standard des process.
- BPMN le plus abordable au début et le plus complet.
- UML: plus utilisé pour software, ne s'applique pas dans des processus d'une entreprise mais ça permet de modéliser le processus dans un développement de software. Il s'agit plus d'une méthodologie qu' un processus.
- DFD: diagramme de flux de données, si un client veut acheter un abonnement, comment ses données et sa demande vont voyager au sein de l'entreprise.
- Si objectif est de modéliser le process d'une entreprise : BPMN est LE standard aujourd'hui. Les autres méthodes répondent à d'autres objectifs + précis, + spécifiques mais si on reste dans cadre du processus d'entreprise (échange entre département, …) alors le mieux est BPMN.

Limites de l'utilisation du modèle BPMN:

Il a utilisé une méthode simplifiée car tout le monde devait comprendre. Ils n'ont pas respecté toute la nomenclature car ça aurait été plus compliqué pour tout le monde.

Personnes à contacter selon lui :

- Natasha Abindagene: envoyer un message pour avoir plus d'infos sur OSSAD.

Appendix B.7: *Natacha Abindagene, 18/03/2024, Teams, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

Projet de transformation pour le client Scarlet : il fallait avoir des processus bien documentés à échanger avec le vendor sélectionné afin de remplacer le BSS de Scarlet (=Business support system : solution de gestion de commande, billing, customer care … dans les telecom).

Pour cela il fallait s'assurer que leur solution couvrait bien les process internes business, et donc leur compréhension devait se faire par différent parties sans ambiguïté

Comme ils allaient transformer la structure il fallait documenter les processus avec des BPMNs afin d'avoir documentation de process interne pour un projet de transformation

Process pour avoir un projet de transformation:

- Contacte ceux qui peuvent présenter le processus répondre (les diagrammes ont été utilisés pour répondre à la solution)
- Quand le vendor a été utilisé, alors le business manager donne les requirements
- En interne: personnes responsables des process doivent faire évoluer les documentations (ex: si changement dans un timer, si dans le business on décide que c'est 2 jours et plus 3, l'owner doit modifier le process pour que ce soit bien à jour)

2. **Connaissez vous d'autres méthodes de modélisation de processus que BPMN?**

- UML: A connu UML théoriquement car l'a étudié mais ne l'a jamais appliqué dans un cadre professionnel. Selon elle, UML est plus facile pour des personnes dans L'IT mais au niveau des process c'est plutôt BPMN. Elle n'a jamais vu personne utiliser UML dans le monde professionnel.
- BPMN: aspect technique plus accessible.
- FLOWCHART: à la base ils faisaient ça mais sans aucune règle, avec pleins de commentaires donc la personne qui le faisait comprenait mais les autres lecteurs ne comprenaient pas toutes les notes, c'était trop brouillon. Ils sont passés de flowchart à BPMN pour être sur que tout le monde pouvait comprendre tout. Ils sont passé de

flowchart à BPMN pour un souci de standardisation, lisibilité, upgrade des process décrits.

3. **Pourquoi BPMN et pas une autre méthode?**

- Celle qui était la plus connue chez le client, plus facile pour le client et pour l'utilisateur (externe), méthode la plus lisible, donc au moins une personne qui comprenait.
- langage simple, standard, belle étendue, modélisation visuelle.
- Elle a installé VISIO (pas coûteux comme outil car que sur 2 ordis). Modélisation dans Visio : simple
- Dans cette mission elle travaillait avec des personnes situé aux Royaumes Unis, en Belgique et n Inde, donc équipe diversifiée. Tout le monde se rencontrait sur BPMN même sans la certification, lisible pour beaucoup de personnes.
- Utilisation aussi de Flowchart (voir au-dessus)
- Simplicité du langage
- Compréhension par toutes les parties prenantes, même celles pas formées au BPMN

4. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

<u>BUT premier:</u>

- Avoir des processus qu'on pourra utiliser, langage commun.
- Documenter les processus utilisés par les employés pour qu'ils les comprennent bien.
- Entretenir la documentation, si quelqu'un partait ou arrivait, tout le monde pouvait comprendre ce qui était en cours/ déjà fait.

<u>3 besoins dans la mission :</u>

- Faisabilité: chacun devait savoir comment gérer tel processus donc important que le document soit compris sans doute.
- Documentation, visibilité.

- Business requirements: entretenir le documentation, si queuql'un partait ou arrivait, tout le monde pouvait comprendre ce qui était en cours/ déjà fait.

5. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Oui mais avec une entente entre les deux responsables (1 en interne et 1 certifiée (ANIKA) sur ce que chaque outil voulait signifier,...)

Processus de validation:

Les analyst valide les process surtout ceux techniques : Un business owner il détermine comment se déroule ces étapes là mais l'analyse valide (ex: les communication, les flows) (responsable du processus du business)

Level de précision: 0 , 1, 2, 3

Ils ont utilisés les 3 en fonction des tâches ex ordering level 2 et CRM level 0

"Nous avons utilisé plusieurs niveau de détails et de levels: 0 – 3, 0 étant le flow entier, level 1 : process, level 2 : subprocess, ... ce qui permet d'aller selon l'audience, plus loin et avoir plus de détails (timing, exceptions, etc) ;

En revanche, le choix de notation était laissé libre au Business Process Managers (certifiés en BPMN), validé auprès du owner du process (experts), et revue en équipe par les analystes (responsables techniques)"

6. **Si oui, pensez-vous qu'il serait intéressant de créer une formation pour les consultants et pour les clients afin de les familiariser avec les standards BPMN ?**

- Certification de ceux qui s'alignent sur les simplifications pour comprendre jusqu'où aller et ce qui sera compris de tous.

- Une formation existe, mais les certifications sont données en externe pour faire valoir la connaissance vis à vis du client, certif nécessaire pour gagner le contrat

- Par rapport au client, leur besoin dépend du contexte de chacun, entreprise avec contexte challenge de documentation des process ou contexte de digitalisation, acquisition, standardisation (entreprise belgo vers entreprise françaises,...). Dans ce type de contexte, c'est très important la certification. La nécessité d'une certification dépend du challenge de l'entreprise.

"La formation BPMN existe au sein de HMP, et je la trouve importante dans les entreprises avec des challenges de documentation, de digitalisation ou dans un contexte de croissance et internationalisation."

7. **Avez- vous utilisé une nomenclature particulière?**

Nomenclature adaptée, elle n'est pas certifiée BPMN mais pour satisfaire Headmind, ils ont proposé 2 personnes certifiées. Le business owner en interne était certifiée, donc 2 personnes certifiées de chaque côté, et elles ont à 2 déterminer une "structure", un "level".

8. **Avez vous un exemple de diagramme à nous montrer ?**

BUT:

Business process management : augmenté d'un cran en maturité pour gérer les process chez Scarlett.

Process Level 0 : CRM (jusque level2)

CAS 1: les disputes : high level car pas tellement de détails

CAS2 : Ordering dans un abonnement mobile (toutes les étapes en détail: quand on commence, E-ticket, temps d'attente etc)

AMBITION: documenter pour déterminer s' ils allaient pouvoir couvrir tous ces processus, voir ce qui allait devoir être établi, comment ils s'organisent, où ça coince

Le niveau et la justesse ont été choisis par les deux femmes à la tête de ces schémas, elles ont fait des choix et se sont mises d'accord pour que tout le monde comprennent et s'accordent sur la conception.

Appendix B.8: *Terence Bricout, 20/03/2024, office HMP, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

Problématique actuelle chez son client (EUROCLEAR):

- Quelle modélisation est la plus adéquate?

Il utilise BPMN dans des processus de data, ETL = extract, transform, load (data pipeline, process sur la durée de vie des data base, aspect data) Il existe une schématisation avec des requêtes SQL, power BI,... tout ça fait partie de la data pipeline et le but est de standardiser.

- Pourquoi BPMN avec tout ça?

Schématiquement parlant, ça lui permet de visualiser les étapes, transformer ETL en BPMN. Pour chaque étape du ETL il décrit comment les tables réagissent entre elles, (ex: trigger qui présente une base de données etc, différents chemins possible,....)

Cf cours d'informatique de Jean Vanderdonck à la LSM pour créer des tables à combiner avec le BPMN car permet d'avoir différentes étapes pour transformer le BPMN et décrire comment les tables interagissent entre elles.

Partie ETL à combiner avec BPMN.

2. **Connaissez vous d'autres méthodes de modélisation de processus que BPMN?**

UML mais trop technique sinon ne connaît pas d'autres méthodes (n'a pas trouvé). Son client avait parlé de UML et BPMN et il a choisi de bosser sur BPMN.

UML est à utiliser pour faire des diagrammes entre les bases de données, par exemple: si un schéma des bases de données, c'est quelque chose de plus technique. Ou bien, autre exemple, une base de données qui se connectent entre elles (1 to many relationships), alors c'est mieux d'utiliser UML car c'est plus technique. Lorsqu'on a des diagrammes techniques, schématisation de tous les attributs d'une base de données alors on utilise UML.

BPMN c'est pour schématiser les étapes d'un processus.

Plusieurs niveaux (voir piscines):

- Si plusieurs acteurs business alors on utilise UML.
- Si besoin d'une description de business alors on utilise BPMN.

Chez son client, ils ont un PDF dans l'entreprise ou ils regroupent tous les BPMN et les tables associées, comme ça ils peuvent les utiliser pour des procédures standards, pour montrer à des nouveaux employés de l'entreprise, etc.

3. **Pourquoi BPMN et pas une autre méthode?**

UMLest trop technique, il a hésité entre BPMN et UML mais pour lui UML est plus technique et trop compliqué pour l'aspect relationnel. BPMN est plus user friendly, parle à tout le monde

4. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

Un peu pour automatiser mais surtout standardiser dans le cas de 2 approches: maintenance et si quelqu'un doit refaire tout le processus que lui a dû faire (Si il n'est plus là, quelqu'un de son équipe peut reprendre son travail facilement, BPMN classique).

5. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Nomenclature simplifiée au maximum. Il existe par exemple différentes flèches mais il a essayé de simplifier : pleine pour changer de process, trigger juste un rond, si il respecte le case alors utilise un triangle avec X.

Plusieurs options sont possibles donc il a simplifié pour que ce soit compréhensible par tout le monde. Il regarde les BPMNs créés par le client (avec couleurs,..) et s'inspire de leur manière de faire pour qu'ils comprennent le sien.

Pour chaque étape, il détaille dans une table tout ce qu'il fait et pourquoi, comme ça c'est très compréhensible.
Par exemple, pour l'étape 2 il explique en bas dans sa table en décrivant le processus très précisément.

Il utilise l'outil ADONIS pour dessiner les BPMNs car quand il reçoit des infos du client, c'est via ADONIS. Il peut donc directement attacher des pièces jointes à son BPMN. ADONIS est un outil très précis et bien explicite, il y a moyen de cliquer sur chaque processus et une page s'ouvre où on peut mettre des infos pour bien expliquer le processus, il y a aussi moyen de créer un sous processus dans chaque process. C'est un outil très complet. Permet d'intégrer des pièces jointes types SOP.

6. **Si oui, pensez-vous qu'il serait intéressant de créer une formation pour les consultants et pour les clients afin de les familiariser avec les standards BPMN ?**

Oui totalement. Il y a déjà une formation BPMN chez Headmind. Quand on lui a demandé de faire des BPMN au début de sa mission (1 mois), il aurait trouvé ça intéressant de faire une

formation sur les outils BPMN (ADONIS ici). Une formation serait une grosse plus value pour les consultants et les clients même si Euroclear est déjà calé et va former pour Adonis, et pousser un peu loin.

Appendix B.9: *Guillaume Cornette, 20/03/2024, office HMP, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

Il est project manager chez Orange (consultant en Cyber) , il gère plusieurs projets différents en utilisant BPMN dans différents projets. Pour 1 des projets, il doit migrer les solutions vers une solution microsoft, donc il met en place un EDR: ça scan la visualisation d'un pc et ca détecte les choses potentiellement dangereuses, ça donne aussi des réponses automatiques (par exemple: qqn télécharge du contenu dangereux, alors ordi bloqué tout seul), ces alertes sont regroupées puis analysées (si pas grave, on gère, si grave on envoie en Pologne), modélisation des processus.

Tout est freestyle chez orange, pas de certification.

Sur un autre projet il a juste "dessiné" un processus avec des flèches,... Plus ressemblant à un FLOWCHART.

Outil qu'il utilise pour ses modélsiations : DRAW.YOU ou outil microsoft VISO

2. **Connaissez vous d'autres méthodes de modélisation de processus que BPMN?**

- Flowchart : quand c'est moins formel
- UML mais n'a jamais utilisé car plus dans un contexte de code,... dans son cas c'est pas du tout utile, sinon UML peut etre utile grâce aux attributs. Selon lui UML et BPMN sont complémentaires.
- PETRI NETS: permet de modéliser une machine, automate, permet de trouver des failles dans automates, quand c'est plus complexes style ALLUMER-ETEINDRE la lumiere, qqc de plus machinal et/ou plus mathématique.Petri Net permet de voir si il n'y a pas de deadlock (exmple: dans un carrefour sans stop, les voitures vont attendre

que celui de droite passe donc personne n'avance, c'est un dead lock), permet de vérifier qu' un programme sait exécuter et qu'il n'y a pas "un bug de non marche".

3. **Pourquoi BPMN et pas une autre méthode?**

- BPMN car Orange lui demande de faire sous forme de BPMN.
- Selon lui c'est la méthode la plus adéquate car : BPMN d'office quand il y a des interactions entre différents pools MAIS si il n'y a que 1 acteur alors plutôt flowchart car moins formel. Selon lui, BPMN est utilisé quand il y a une interaction entre humains alors que si on fait juste au niveau informatique alors c'est ok d'utiliser UML.
- Il aime bien BPMN car notion de vertical et horizontal, petite notion de temps
- Voir norme ISO

4. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

Un peu les 2, tout ce qu'il fait est documenté au cas ou après il n'est plus là comme ça c'est compréhensible par tout le monde.

5. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Il n'utilise pas la bonne sémantique mais plutôt un BPMN très simplifié, une sémantique pas du tout correcte car seulement pour une portée de quelques personnes. Si la portée est de millier de personnes alors il respecte à fond la nomenclature pour que partout dans le monde entier on puisse le comprendre.

Il demande toujours du feedbacks pour être sûr que le processus soit bien représenté. Et il utilise la base de la nomenclature, sinon les gens ne comprendraient pas tout. Quand il faut quelque chose de plus compliqué il doit faire une fiche explicative allant avec le diagramme pour que tout le monde comprenne.

6. **Si oui, pensez-vous qu'il serait intéressant de créer une formation pour les consultants et pour les clients afin de les familiariser avec les standards BPMN ?**

Oui cela serait bien pour le client afin que le consultant et le client se mettent d'accord sur une nomenclature commune.

Appendix B.10: *Jean-Charles Verdier, 02/04/2024, office HMP, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

D'abord il a été stagiaire chez Orange pour étudier le processus de "screening" puis il est revenu en mission afin d'implémenter un nouveau process pour la gestion des personnes devant se faire screener (= quand des gens ont besoin d'infrastructures critiques dans leur travail, ils doivent faire une demande auprès de l'ANS qui va checker leur background pour voir s' ils peuvent avoir accès à ces infrastructures, sorte de contrôle de police)

Son travail= intégrer le processus de screening dans le process d'unboarding

Il a repris les processus d'unboarding et les a modifiés pour ajouter tous les process liés au screening. Pour cela il a fait ça sous forme de BPMN/diagrammes. Il y a un processus particulier pour chaque type de population (ex: partner,...). Il a dû faire un BPMN par population.

2. **Connaissez vous d'autres méthodes de modélisation de processus que BPMN?**

Pendant ses études il a vu UML, use case diagrams (ressemble a des diagrammes de séquence, mais décrit le use case dans ce schéma) et diagrammes de séquence

3. **Pourquoi BPMN et pas une autre méthode?**

- Modèle le plus clair pour présenter les process
- Il avait le choix entre BPMN ou diagramme de séquences mais il trouvait plus visuel les BPMN
- Il a choisi BPMN car Orange utilisait déjà ça dans d'anciens diagrammes, donc il reprenait les BPMN de Orange et les adaptait/ modifiait. Il ne rajoutait pas de nouveaux éléments qui n'étaient pas dans les BPMNs qu' orange faisait déjà.
- Il a aussi utilisé des diagrammes de séquences, car on le lui a demandé mais finalement ils n'ont pas retenu ceux-là (il voulait aussi garder une uniformité entre les diagrammes de l'entreprise qui étaient déjà des BPMN) .

4. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

Dans un but de standardisation, pour que tout le monde le comprenne. Un projet se fait étape par étape et à chaque étape il faut présenter le projet donc il montre les BPMN. Cela lui permet d'expliquer le processus à toute l'équipe, les BPMNs servent pour le moment à standardiser mais plus tard, à la fin de la mission, ce sera plus de l'automatisation pour toute l'entreprise.

5. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Ces diagrammes ne suivent pas la nomenclature précise, il a utilisé des symboles, éléments que Orange utilisait déjà dans d'autres BPMNs, il n'a rien ajouté de nouveaux.

Au debut il faisait ses BPMNs sur Powerpoint, mais maintenant il fait sur VISIO donc c'est beaucoup mieux. Il n'a pas vu de limite car il modélise des process assez simples, clairs,... il a juste besoin des rectangles et des losanges.

6. **Si oui, pensez-vous qu'il serait intéressant de créer une formation pour les consultants et pour les clients afin de les familiariser avec les standards BPMN ?**

Surement oui, il doit demander à son client.

Appendix B.11: *Sofiane Bengaoua, 02/04/2024, office HMP, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

Pour un sujet de cyber défense chez l'ORÉAL. Il travaille sur l'automatisation de réponses à accidents (outils SOR, security orchestration and response, outil low code qui permettent de modéliser sous forme de diagrammes pour des équipes)

Dans le cadre de projet interne chez Headmind Partners, il travaille avec un éditeur (automatisation de réponse accident). Avant le playbook, création d'un squelette (diagramme sur papier) ensuite enrichi avec du code pour que les briques interagissent entre elles. Il a

également fait un guide d'intégration SOR. LAB CYBERDÉFENSE, guide qui permet d'expliquer les solutions SOR et cette année ils ont fait 9 de ces playbooks, (ex: attaque d'un DDOS, process à suivre)

Dans ses missions clients, il ne voyait pas la valeur ajoutée d'utiliser BPMN. Sauf dans son premier projet où il a dû faire les BPMN pour que tout le monde comprenne, pour être en accord avec les templates des équipes. E n'était pas une nécessité absolue dans ce genre de schémas MAIS ok pour des processus plus complexes alors ça peut être intéressant d'utiliser les pictogrammes.

Quand c'est multiniveau, mieux d'utiliser BPMN mais pour des schémas très simples alors non (plutôt flowchart). Selon lui intéressant dans des cas orienté ORGANISATION manager/business (matrice d'escalade,...) mais pas des cas TECHNIQUES (création ligne de service, SOR,... ne nécessitent pas d'expliquer les rôles etc). Selon lui, le modèle BPMN est ok pour modéliser des relations.

2. **Connaissez vous d'autres méthodes de modélisation de processus que BPMN?**

Que UML et BPMN un peu.

3. **Pourquoi BPMN et pas une autre méthode?**

Il utilise surtout des WORKFLOWs et pas de BPMN.

Il ne connaissait que UML qui est pour des base de donnéesdonc ça n'allait pas, il ne connaissait pas BPMN donc a fait WORKFLOW pour un soucis de faire au plus simple. Il utilise outil open source pour modéliser un diagramme sans aucune méthodologie particulière.

SI il connaissait en fait le modèle BPMN mais ne savait pas le nom. Il devait créer une ligne de service pour un client sur JIRA, il fallait modéliser de la phase de création de formulaire jusqu'à la phase de processus en tant que tel, soucis d'automatisation. Il a alors utilisé le modèle BPMN sans savoir qu'il s'agissait de ce modèle.

Exemple de processus qu'il a dû modéliser:

- Intégrer des sites pour une raison de sécurité: travaillait sur WAF qui fait office de DMS, chaque fois que le client (l'oreal) voulait créer un nouveau site, il devait passer par un formulaire et préciser le site,... L'oréal devait répondre à des questions,.. Puis à la fin il y avait la fermeture du ticket.

- Matrice de décision; a dû utiliser BPMN car c'était le template du client et il a trouvé ça intéressant de le faire.

Il utilise DRAW.YO (outil no code) mais ne respecte pas une nomenclature particulière

4. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

Standardiser le processus pour que tout le monde le comprenne et une automatisation de celui-ci sur base de API.

5. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Objectif: quelqu'un qui n'est pas développeur peut analyser une réponse à incident grâce à ces notebooks, sans considérer l'aspect API par la suite. DONC oui simplification de la nomenclature.

6. **Si oui, pensez-vous qu'il serait intéressant de créer une formation pour les consultants et pour les clients afin de les familiariser avec les standards BPMN ?**

Selon lui, si pour le client il y a une valeur ajoutée alors oui BPMN et oui une formation serait nécessaire mais sinon pour lui non.

Pour lui, une formation très basique est ok mais sinon pas réellement intéressant car il veut faire au plus simple et faire quelque chose de rapide.

Appendix B.12: *Anika, 04/04/2024, Teams, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

Chez Scarlet pour un changement de CRM. Il fallait redessiner tous les nouveaux FLUX, il y en avait 180 au total, elle devait implémenter les nouveaux et les faire valider.

Elle a dû attendre que les business owner décrivent les processus et ensuite elle dessinait (sur VISIO) puis revenait vers eux pour voir si ils étaient corrects.

Exemple de flux:

- Achat d'une carte sim, création d'un compte client
- Déménagement d'une personne au niveau du wifi, quand tu le dis sur le site, flux derrière (provisioning)
- Comment les gens reçoivent un téléphone chez eux
- Facture envoyée à la fin du mois
- Support après vente, quand le client a un problème, call center à l'etranger

2. **Pourquoi BPMN et pas une autre méthode?**

Le client demandait BPMN. Elle a du faire une formation avec un employé client comme ça deux personnes étaient aptes à modéliser correctement.

Dans l'entreprise ils utilisaient tous BPMN, donc tout le monde maîtrisait le langage BPMN, ils avaient fait des flux auparavant mais pas vraiment correctement représentés.

3. **Avez vous utilisé ces diagrammes dans le but d'automatiser le processus étudié ou pour standardiser un processus afin que tout le monde sache l'utiliser ?**

Les processus existaient déjà donc c'était dans le but de les standardiser pour les implémenter dans un nouveau système CRM. Au final un peu les 2 mais plus pour les standardiser.

4. **Avez-vous eu besoin de simplifier vos diagrammes pour qu'ils soient compris de tous ?**

Il fallait faire au plus simple, mais elle allait quand même en profondeur.

Il faut toujours commencer par le flux high level, puis plus précis avec des sous niveaux, donc tout n'est pas au même niveau.

Comme tout le monde avait accès aux BPMNs qu'elle faisait, ils devaient être assez clairs.

5. **Voyez-vous des limites au modèle BPMN?**

- Oui, il y a plusieurs interpretations differentes possibles, ce n'est pas hyper clair mais peut être que c'est un problème chez scarlett aussi, elle ne sait pas trop.
- Ce n'est pas toujours clair de savoir quel symbole utiliser.
- Quand le processus est complexe c'est dur d'utiliser BPMN car pas ce n'est pas toujours clair.

Appendix B.13: *Lucky KAKPO, 12/04/2024, office HMP, Interview guide with notes taken of the consultant's answers.*

1. **Dans quel type de mission avez-vous utilisé la modélisation BPMN?**

BPMN: business analyst quand tu décris un business process (ex: en tant que user)

UML : pour décrire une base de données, relations entre les relations, ex: un client achète dans telle branche etc

Mission:

Il était d'abord stagiaire chez ORANGE pour revoir comment orange utilisait les outils dans les magasins. Ils avaient des écrans mais éteint, le but était de mieux utiliser les outils digitaux dans les magasins (télévision,...), à travers les écrans, les comportements des vendeurs (ex: un vendeur vient te parler quand tu entres dans le magasin),

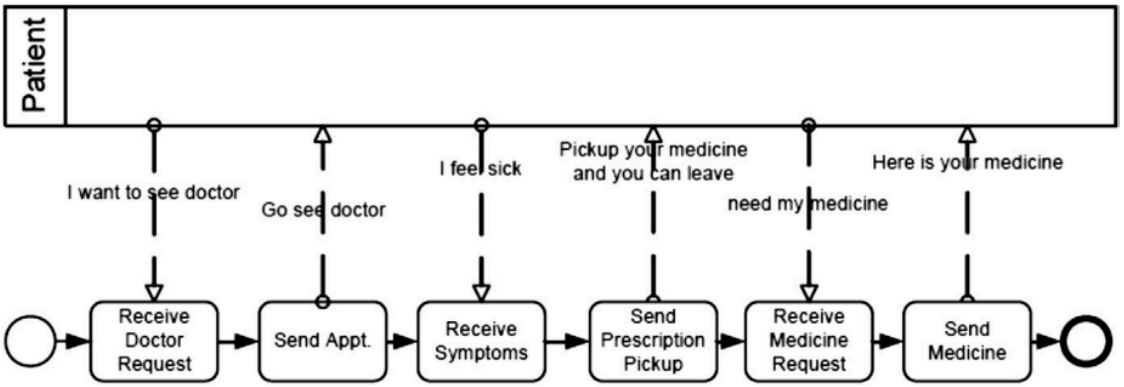
Il devait analyser des bases de données avec des infos sur chaque magasin dans chaque ville. Donc, il devait désigner dans power BI (tu peux désigner des bases de données et puis les visualiser) le but était de définir correctement les relations entre les magasins, les clients et les vendeurs. Donc 3 relations, (entités= client, vendeur, magasin), chaque entité avait des attributs (nom, prenom, localisation,...). Il devait définir les relations entre les 3 tables (1 client a acheté dans tel magasin, tel vendeur travaille la,...). SI modélisation pas correct alors relation pas bien décrite. Il y a 3 types de relations (1 to 1 (pas dans sa mission), 1 to many, many to many (plusieurs vendeurs qui ont travaillé dans plusieurs magasins)). Il devait vérifier les relations pour être sûr.

ERD: type d'UML, interactions entre les bases de données. Ce n'est pas une description de business process. Il faut donc bien séparer les 2 catégories: BPMN pour la partie description de processus, UML pour la partie data.

Appendix C: *Example of a BPMN representation of a private process.*

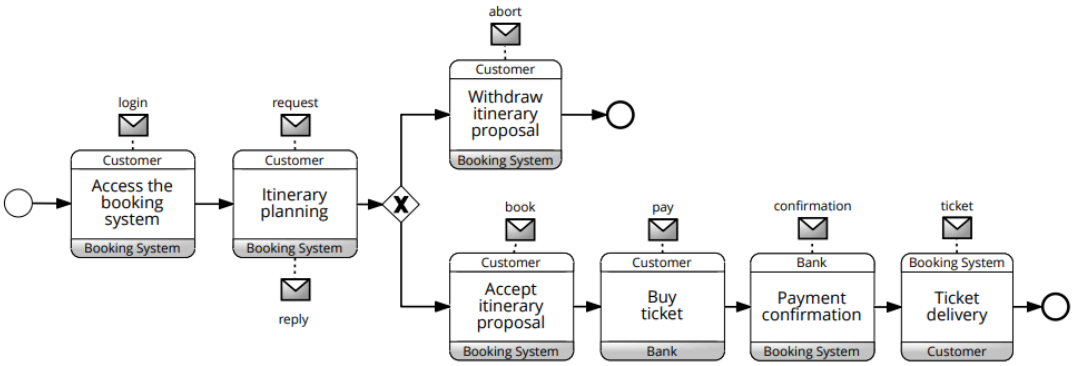
Source : Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix D: *Example of a BPMN representation of a public process.*



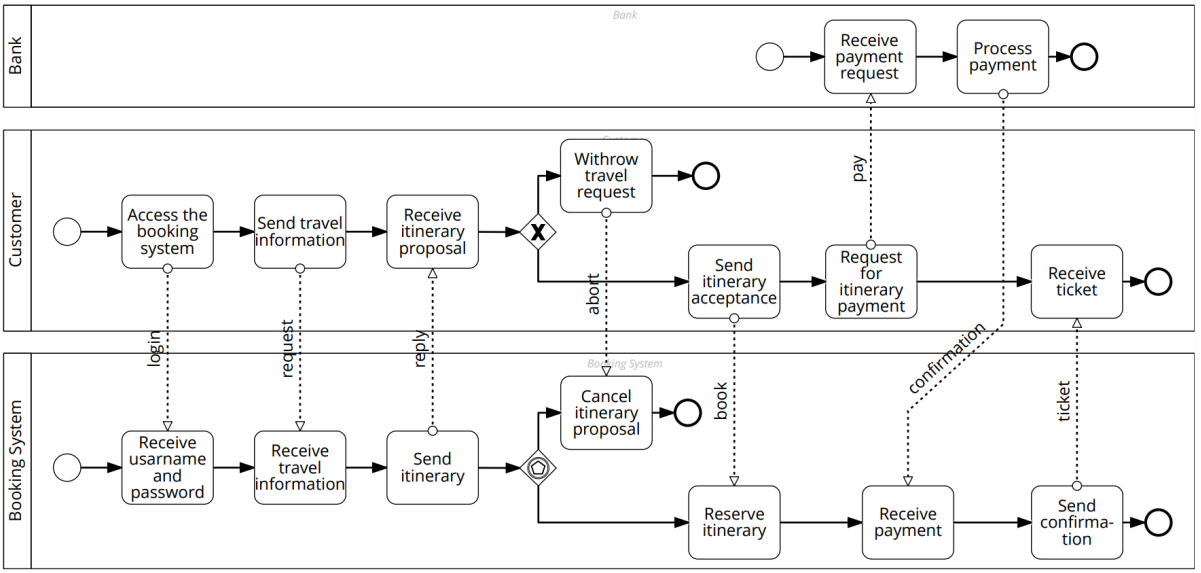
Source : Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix E: *Example of a BPMN choreography diagram.*



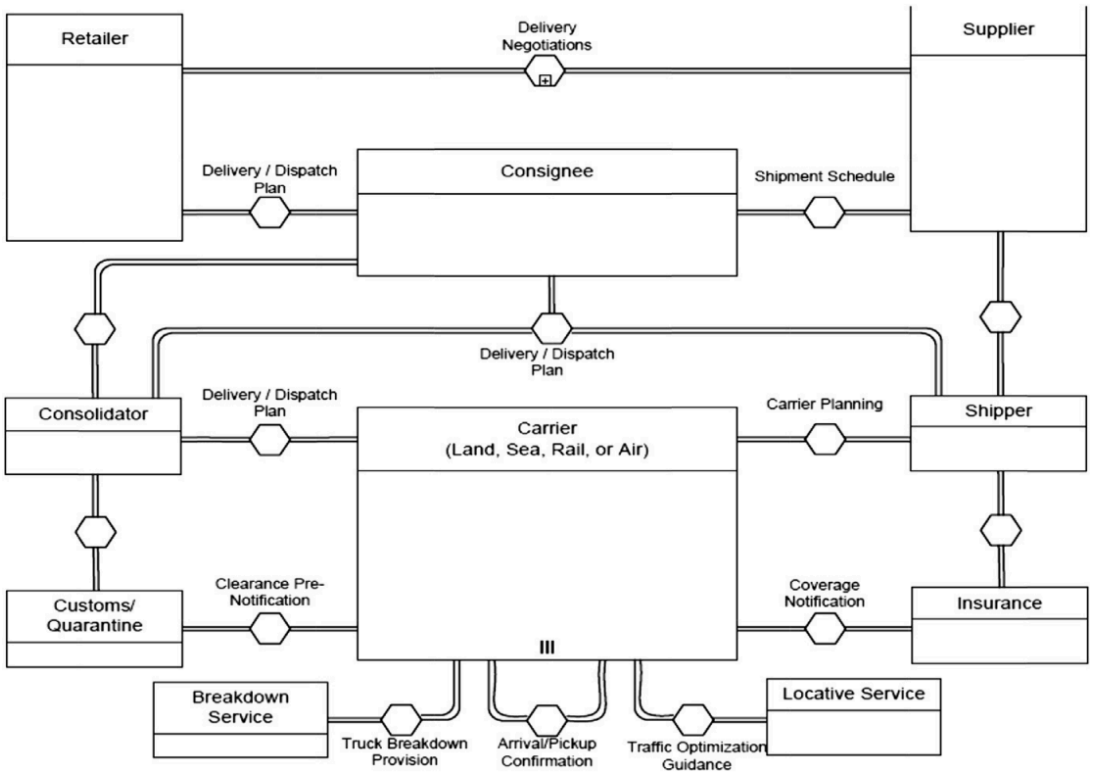
Source: Corradini, F., Morichetta, A., Polini, A., Re, B., & Tiezzi, F. (2020). Collaboration vs. choreography conformance in BPMN. *Logical Methods in Computer Science*, *16*.

Appendix F: *Example of a BPMN collaboration diagram.*



Source: Corradini, F., Morichetta, A., Polini, A., Re, B., & Tiezzi, F. (2020). Collaboration vs. choreography conformance in BPMN. *Logical Methods in Computer Science*, *16*.

Appendix G: *Example of a BPMN conversation diagram.*



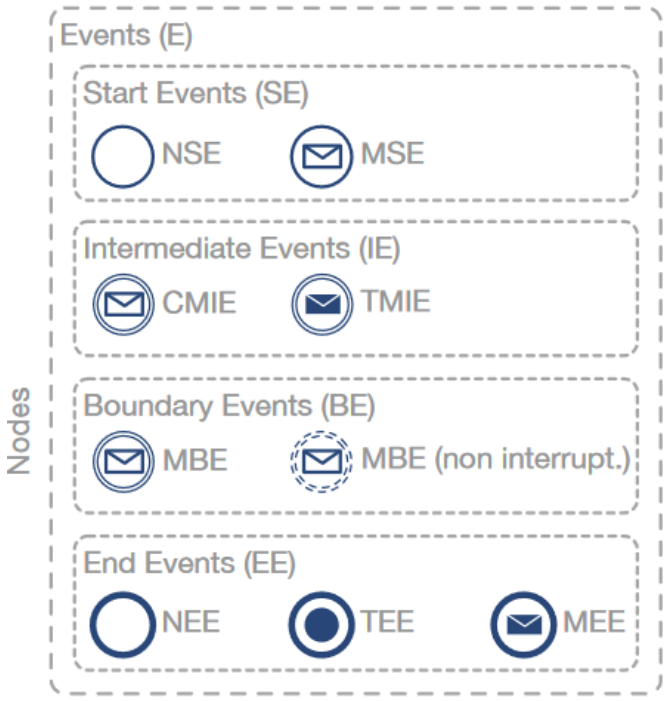
Source : Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix H: ***Table showing the graphical elements of BPMN conversations and their descriptions.***

BPMN 2.0.2	Conversations Description
Conversation	A Conversation defines a set of logically related Message Flows. When marked with a (+) symbol it indicates a Sub-Conversation, a compound conversation element.
Call Conversation	A Call Conversation is a wrapper for a globally defined, re-usable Conversation or Collaboration. A call to a Collaboration is marked with a (+) symbol.
Conversation Link	Connects Conversations and Participants.

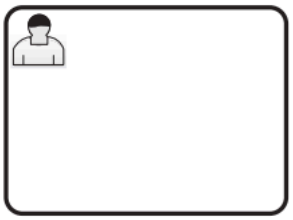
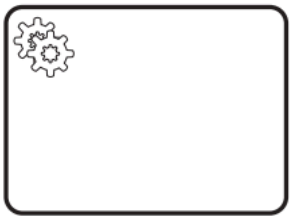
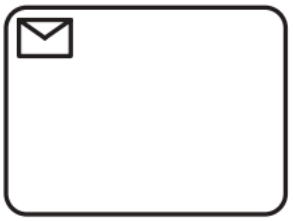
Source : Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix I: *Graphical representations of some events in BPMN notation.*



Source: Houhou, S., Baarir, S., Poizat, P., & Quéinnec, P. (2019). A First-Order Logic semantics for Communication-Parametric BPMN collaborations. In *Business Process Management. BPM 2019. Lecture notes in computer science* (Vol. 11675, pp. 52–68). Springer, Cham. https://doi.org/10.1007/978-3-030-26619-6_6

Appendix J: ***Graphical representation of BPMN tasks with their description.***

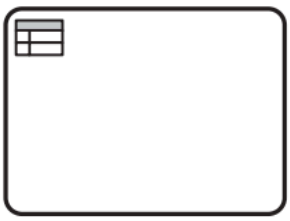
BPMN 2.0.2	Task Description
 None	No special task type is indicated.
 User Task	A User Task is a typical "workflow" task in which a human performer performs the task with the assistance of a software application and could be scheduled through a task list manager of some sort.
 Manual Task	A Manual Task is a task that is expected to be performed without the aid of any business process execution engine or application.
 Service Task	A Service Task is a task that uses some sort of service, which could be a web service or an automated application.
 Receive Task	A Receive Task is a simple task that is designed to wait for a message to arrive from an external participant (relative to the process).

Send

A Send Task is a simple task that is designed to send a message to an external participant (relative to the process).

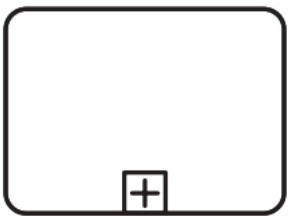
Script

A Script Task is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret. When the task is ready to start, the engine will execute the script. When the script is completed, the task will also be completed.



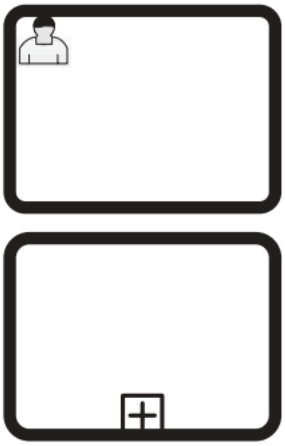
Business Rule

A Business Rule Task provides a mechanism for the process to provide input to a Business Rules Engine and to get the output of calculations that the business rules engine might provide. The input/output specification of the task will allow the process to send data to and receive data from the Business Rules Engine.



Sub-Process

A Sub-Process is a type of activity within a process, but it also can be "opened up" to show a lower-level process. This is useful for process decomposition or general process organization.



Call Activity

A Call Activity is a type of activity within a process. It provides a link to reusable activities: for example, it will call a task into the Process (see upper figure on the left) or another Process (see lower figure on the left).

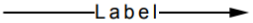
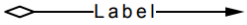
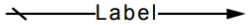
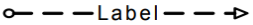
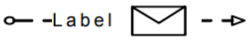
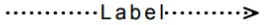
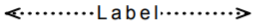
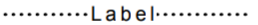
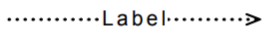
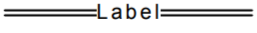
Source: Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix K: *Graphical representation of BPMN gateways with their description.*

BPMN 2.0.2	Gateway Description
 Gateway	Gateways are used to control how process paths converge and diverge within a process.
  Exclusive Gateway	The Event Gateway, when splitting, routes sequence flow to only one of the outgoing branches, based on conditions. When merging, it awaits one incoming branch to complete before continuing the flow. The Gateway can be displayed with or without the "X" marker, but the behavior is the same.
 Inclusive Gateway	The Inclusive Gateway, when splitting, allows one or more branches to be activated, based on conditions. All active incoming branches must complete before merging.
 Parallel Gateway	The Parallel Gateway, when splitting, will direct the flow down all the outgoing branches. When merging, it awaits all the in branches to complete before continuing the flow.
  Event-based Gateway	The Event Gateway is always followed by catching events or receive tasks. The flow of the Process is routed to the subsequent event/task which happens first. When merging, it behaves like an Event Gateway. This Gateway can be configured such that it can be used to start a Process, based on the first event that follows it (see the lower figure on the left).
 Parallel Event-based Gateway	The Parallel Event Gateway is only used for starting a Process. It is configured like a regular Event Gateway, but *all* of the subsequent events must be triggered before a new process instance is created.
 Complex Gateway	The Complex Gateway defines behavior that is not captured by other gateways. Expressions are used to determine the merging and splitting behavior.

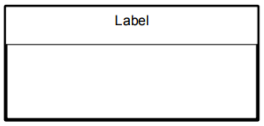
Source: Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix L: *Graphical representation of BPMN object connection.*

Kind:	Depiction:	Specific Depiction Resolution:	
		bpmnElement:	BPMNShape Attributes:
Sequence Flow		SequenceFlow where default is false and conditionExpression is unspecified.	None
Conditional Sequence Flow		SequenceFlow where default is false and conditionExpression is specified (exception when source is a Gateway).	None
Default Sequence Flow		SequenceFlow where default is true and conditionExpression is unspecified.	None
Message Flow		MessageFlow	messageVisibleKind is unspecified.
Initiating Message Flow with Decorator		MessageFlow	messageVisibleKind is initiating.
Non-Initiating Message Flow with Decorator		MessageFlow	messageVisibleKind is non-initiating.
Association		Association where associationDirection is none.	None
Directional Association		Association where associationDirection is one.	None
Bi -Directional Association		Association where associationDirection is both.	None
Data Association		None	The targetElement of the BPMNEdge is itself of type BPMNEdge where bpmnElement is of type SequenceFlow
Directed Data Association		DataInputAssociation or DataOutputAssociation	None
Conversation Link		ConversationLink	None

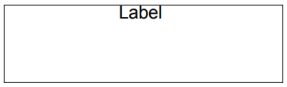
Source: Object Management Group. (2013). Business Process Model and Notation (BPMN) : Version 2.0.2. In *OMG*. https://www.omg.org/spec/BPMN/2.0.2/PDF

Appendix M: *Graphical representation of BPMN pools.*

Kind:	Depiction:	Specific Depiction Resolution:	
		bpmnElement:	BPMNShape Attributes:
Horizontal Pool		Participant where ParticipantMultiplicity is unspecified or set and its maximum attribute is 1.	None or isVertical is false
Horizontal Pool - with Multi Instance Participant		Participant where ParticipantMultiplicity is set and its maximum attribute is > 1..	None or isVertical is false
Vertical Pool		Partcipant where ParticipantMultiplicity is unspecified or set and its maximum attribute is 1.	isVertical is true
Vertical Pool - with Multi Instance Participant		Participant where ParticipantMultiplicity is set and its maximum attribute is > 1.	isVertical is true

Source: Object Management Group. (2013). Business Process Model and Notation (BPMN) : Version 2.0.2. In *OMG*. https://www.omg.org/spec/BPMN/2.0.2/PDF

Appendix N: *Graphical representation of BPMN lanes.*

Kind:	Depiction:	Specific Depiction Resolution:	
		bpmnElement:	BPMNShape Attributes:
Horizontal Lane		Lane	None or isVertical is false
Vertical Lane		Lane	isVertical is true

Source: Object Management Group. (2013). Business Process Model and Notation (BPMN) : Version 2.0.2. In *OMG*. https://www.omg.org/spec/BPMN/2.0.2/PDF

Appendix O: *Graphical representation of BPMN artefacts.*

Kind:	Depiction:	Specific Depiction Resolution:	
		bpmnElement:	BPMNShape Attributes:
Group		Group	None
Text Annotation		Text Annotation	None

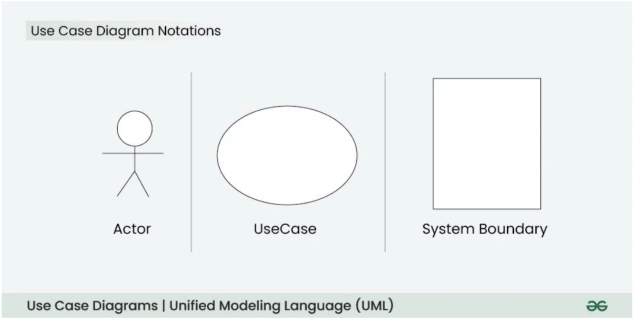
Source: Object Management Group. (2013). Business Process Model and Notation (BPMN) : Version 2.0.2. In *OMG*. https://www.omg.org/spec/BPMN/2.0.2/PDF

Appendix P: *Graphical representation of BPMN data objects with their description.*

BPMN 2.0.2	Data Description
 Data Object	A Data Object represents the data that are used as inputs and outputs to the activities of a process. Data Objects can represent singular objects or collections of objects.
 Data Input	A Data Input is an external data input for the entire process. It is a kind of input parameter.
 Data Output	A Data Output is the data result of the entire process. It is a kind of output parameter.
 Data Store	A Data Store is a place where the process can read or write data (e.g., a database or a filing cabinet). It persists beyond the lifetime of the process instance.
 Collection of Data Objects	A Collection of Data Objects represents a collection of data elements related to the same data entity (e.g., a list of order items).

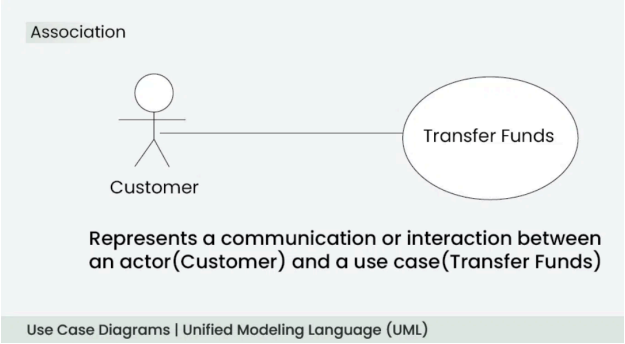
Source: Von Rosing, M., White, S., Cummins, F., & De Man, H. (2015). Business Process Model and Notation (BPMN). In *The Complete Business Process Handbook* (Vol. 1, pp. 429-457). Elsevier - Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-799959-3.00021-5

Appendix Q: *Graphical representation of elements of use case semantics.*



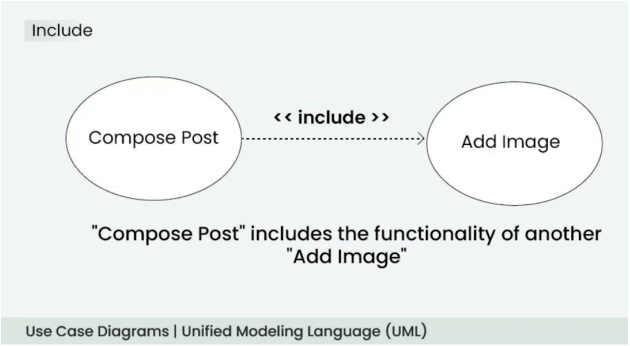
Source: *Use case diagrams Unified Modeling Language (UML)*. (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

Appendix R: *Example of an association relationship in the UML Use Case semantics.*



Source: *Use case diagrams Unified Modeling Language (UML)*. (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

Appendix S: *Example of an "include" relationship in the UML Use Case semantics.*

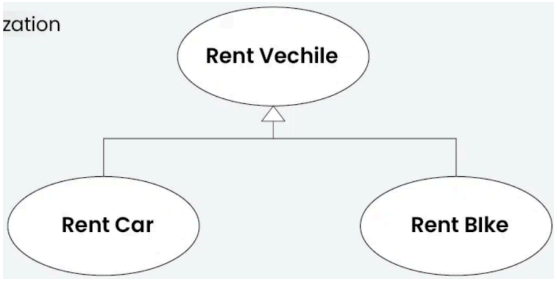


Source: *Use case diagrams Unified Modeling Language (UML)*. (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

Appendix T: *Example of an "extend" relationship in the UML Use Case semantics.*

Source: *Use case diagrams Unified Modeling Language (UML)*. (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

Appendix U: *Generalisation relationship in the UML Use Case semantics.*



Source: *Use case diagrams Unified Modeling Language (UML)*. (2024, February 9). GeeksforGeeks. Retrieved April 15, 2024, from https://www.geeksforgeeks.org/use-case-diagram/

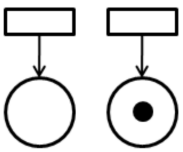
Appendix V: *Graphical representation of Petri Nets elements with their description..*

Element	Description
 Transition	A transition is a modeling unit representing a process step.
 Place  Place with tokens	A place can contain tokens that define a part of the state of the net. A distribution of tokens over the places is called a marking, representing the state of the net.
 Inflow Arc	An inflow arc is the arrow from a (incoming) place to a transition. If all incoming places connected to a transition have enough tokens according to the firing rules, the transition is enabled and can fire.
 Outflow Arc	An outflow arc is the arrow from a transition to a (outgoing) place. If a transition fires, it consumes tokens from its incoming places, and produces tokens in its outgoing places according to the firing rules.
 Reset Arc	When a place is connected to a transition with a reset arc, all tokens are removed from it when the transition fires.
 Inhibitor Arc	When a place connected to a transition via an inhibitor arc contains a token, the transition cannot be enabled.

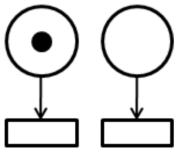
Source: Raedts, I. G. J., Petkovic, M., Usenko, Y. S., Werf, van der, J. M. E. M., Groote, J. F., & Somers, L. J. A. M. (2007). Transformation of BPMN models for behaviour analysis. In J. C. Augusto, J. Barjis, & U. Ultes-Nitsche (Eds.), *Proceedings of the 5th International Workshop in conjunction with the 9th International Conference on Enterprise Information Systems (MSVVEIS 2007) 12-13 June 2007, Funchal, Madeira, Portugal* (pp. 126-137). INSTICC Press.https://doi.org/10.5220/0002428801260137

Appendix W: *Graphical representation of Petri Nets transitions.*

W.1: *Graphical representation of a Petri Nets source transition.*



W.2 : *Graphical representation of a Petri Nets sink transition.*



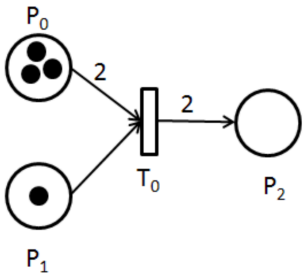
Source: Callou, G., Maciel, P., Tutsch, D., Arajo, J., Ferreira, J., & Souz, R. (2012). A Petri Net-Based Approach to the Quantification of Data Center Dependability. In *InTech eBooks*. https://doi.org/10.5772/47829

Appendix X: *Graphical representation of Petri Nets firing phenomenon.*

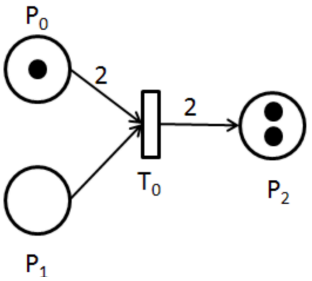
X.1: *Mathematical formalism of a PN.*

P = {p0, p1, p2}
T = {t0}
F = {(p0, t0), (p1, t0), (t0, p2)}
W = {(p0, t0)=2, (p1, t0)=1, (t0, p2)=2}
m0 = (3, 1, 0)

X.2: *Graphical representation of a PN before the firing of* t_0.

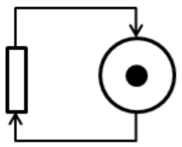


X.3: *Graphical representation of a PN after the firing of* t_0.



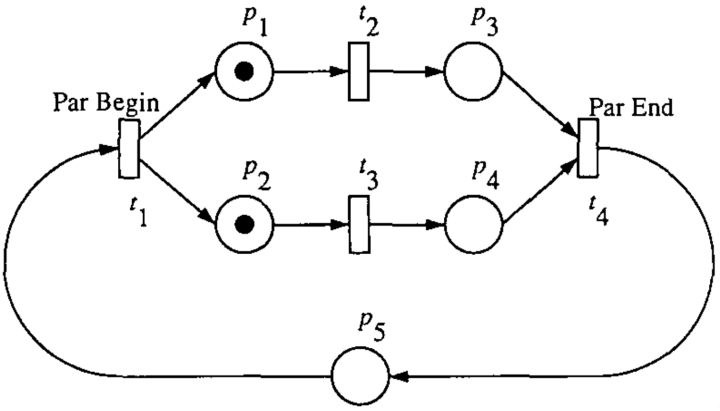
Source : Callou, G., Maciel, P., Tutsch, D., Arajo, J., Ferreira, J., & Souz, R. (2012). A Petri Net-Based Approach to the Quantification of Data Center Dependability. In *InTech eBooks*. https://doi.org/10.5772/47829

Appendix Y: *Graphical representation of a self-loop in Petri Nets.*



Source : Callou, G., Maciel, P., Tutsch, D., Arajo, J., Ferreira, J., & Souz, R. (2012). A Petri Net-Based Approach to the Quantification of Data Center Dependability. In *InTech eBooks*. https://doi.org/10.5772/47829

Appendix Z: *Petri Nets diagram modelling deterministic parallel activities.*



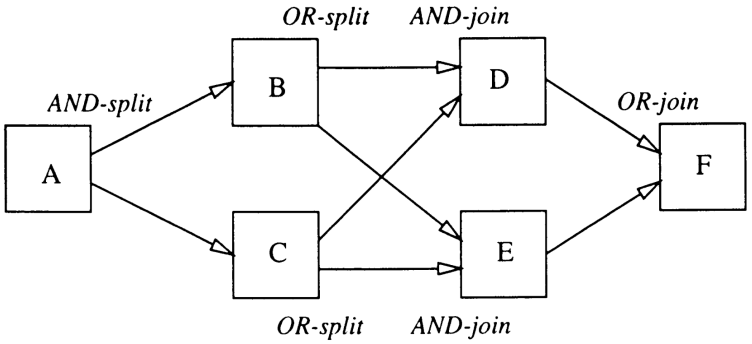
Source: Murata, T. (1989). Petri nets : Properties, analysis and applications. *Proceedings Of The IEEE*, *77*(4), 541-580. https://doi.org/10.1109/5.24143

Appendix AA: *Typical interpretation of Petri Nets transitions and places.*

Input Places	Transition	Output Places
Preconditions	Event	Postconditions
Input data	Computation step	Output data
Input signals	Signal processor	Output signals
Resources needed	Task or job	Resources released
Conditions	Clause in logic	Conclusion(s)
Buffers	Processor	Buffers

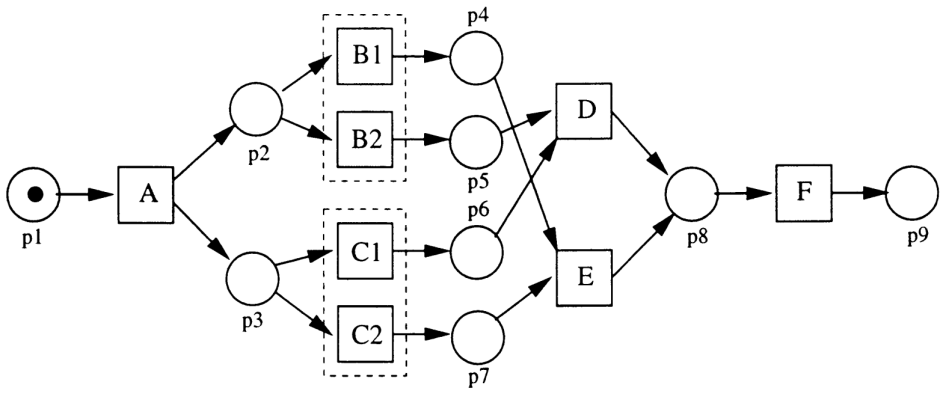
Source: Murata, T. (1989). Petri nets : Properties, analysis and applications. *Proceedings Of The IEEE*, *77*(4), 541-580. https://doi.org/10.1109/5.24143

Appendix AB: *Graphical representation of a workflow procedure as an event-based approach where transitions are modelled explicitly and the states between subsequent transitions are modelled implicitly.*



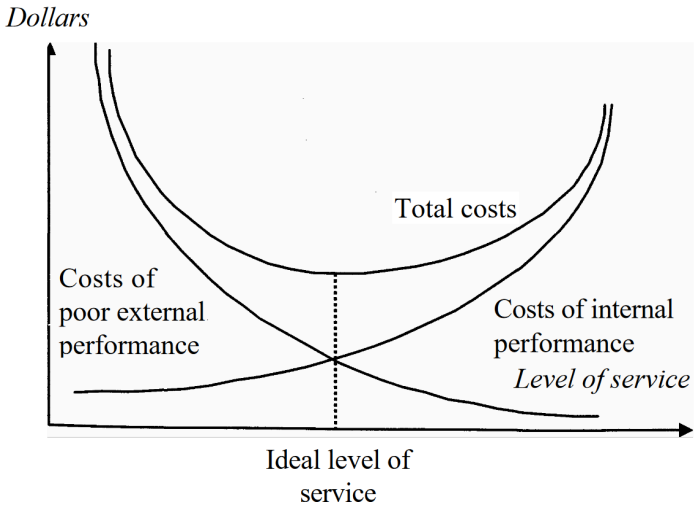
Source: van der Aalst, W. (1998). Three Good Reasons for Using a Petri-Net-Based Workflow Management System. In *Information and Process Integration in Enterprises. The Springer International Series in Engineering and Computer Science* (vol 428, p. 161-182). Springer, Boston. https://doi.org/10.1007/978-1-4615-5499-8_10

Appendix AC: *Graphical representation of the workflow procedure of appendix AB converted in Petri Nets to obtain a state-based description where tasks are modelled by transitions and intermediate states are modelled by places.*



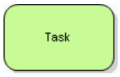
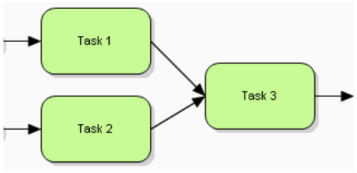
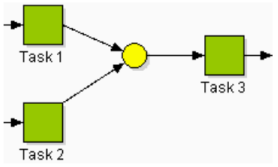
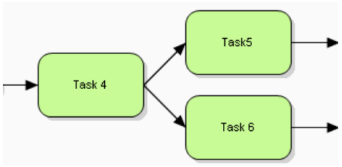
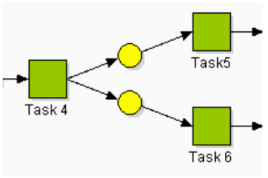
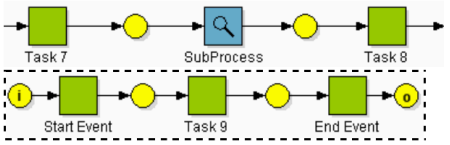
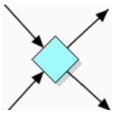
Source: van der Aalst, W. (1998). Three Good Reasons for Using a Petri-Net-Based Workflow Management System. In *Information and Process Integration in Enterprises. The Springer International Series in Engineering and Computer Science* (vol 428, p. 161-182). Springer, Boston. https://doi.org/10.1007/978-1-4615-5499-8_10

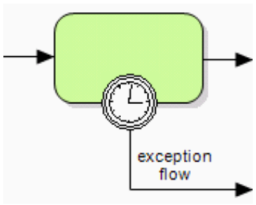
Appendix AD: *Graph on weighing external performance versus internal effort.*



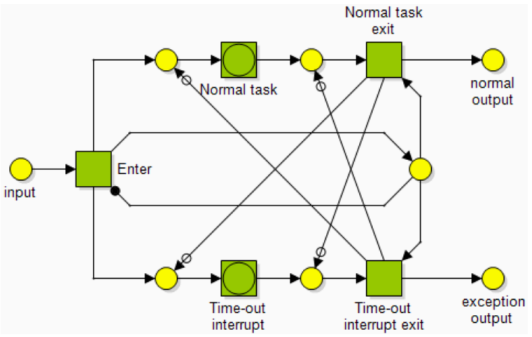
Source: van der Aalst, W., & Hee, K.M. (2002). Workflow Management: Models, Methods, and Systems. In *The MIT Press eBooks*. https://doi.org/10.7551/mitpress/7301.001.0001

Appendix AE: *Table showing the rules for transforming a BPMN diagram into a PN in order to analyse the accuracy of the BPMN.*

BPMN elements	Petri net elements
 Start Event	 Emitter connected to place
 End Event	 Place connected to Collector
 Task	 Transition
 Task with two incoming sequence flows	 Merge-like behaviour
 Task with two outgoing sequence flows	 Fork-like behaviour
  A connected Subprocess	 A connected Subnet
 XOR Gateway	 XOR
 AND Gateway	 Transition



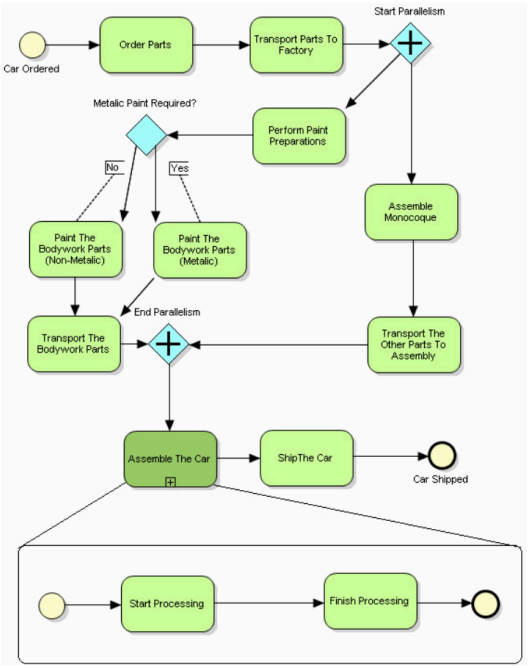
Time-out interruptible task

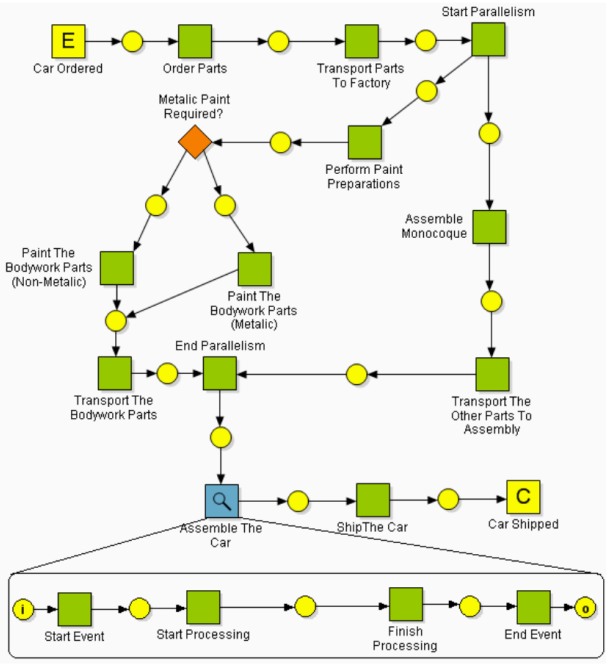


The translation result of the time-out interruptible task

Source: Raedts, I. G. J., Petkovic, M., Usenko, Y. S., Werf, van der, J. M. E. M., Groote, J. F., & Somers, L. J. A. M. (2007). Transformation of BPMN models for behaviour analysis. In J. C. Augusto, J. Barjis, & U. Ultes-Nitsche (Eds.), *Proceedings of the 5th International Workshop in conjunction with the 9th International Conference on Enterprise Information Systems (MSVVEIS 2007) 12-13 June 2007, Funchal, Madeira, Portugal* (pp. 126-137). INSTICC Press.https://doi.org/10.5220/0002428801260137

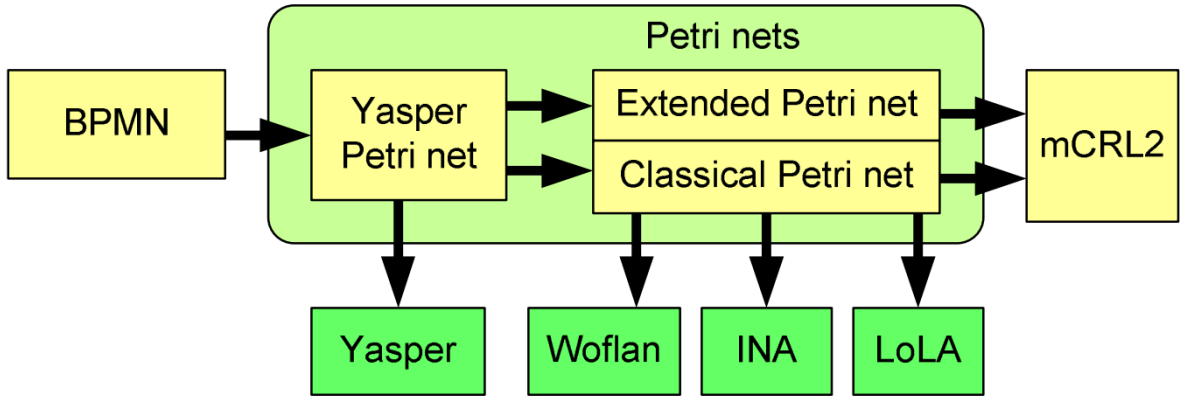
Appendix AF: *Example of a BPMN model and derived Petri net, illustrating the precision with which timing exceptions and other specific behaviours are mapped to preserve the semantics of the model.*





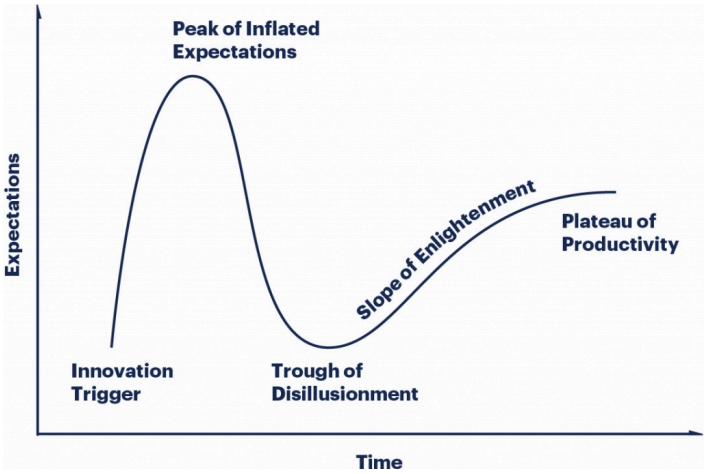
Source: Raedts, I. G. J., Petkovic, M., Usenko, Y. S., Werf, van der, J. M. E. M., Groote, J. F., & Somers, L. J. A. M. (2007). Transformation of BPMN models for behaviour analysis. In J. C. Augusto, J. Barjis, & U. Ultes-Nitsche (Eds.), *Proceedings of the 5th International Workshop in conjunction with the 9th International Conference on Enterprise Information Systems (MSVVEIS 2007) 12-13 June 2007, Funchal, Madeira, Portugal* (pp. 126-137). INSTICC Press.https://doi.org/10.5220/0002428801260137

Appendix AG: *Diagram showing the tools and transformations used by Raedts et al (2007) to analyse Petri Nets and transformations from, to and between Petri Nets.*



Source: Raedts, I. G. J., Petkovic, M., Usenko, Y. S., Werf, van der, J. M. E. M., Groote, J. F., & Somers, L. J. A. M. (2007). Transformation of BPMN models for behaviour analysis. In J. C. Augusto, J. Barjis, & U. Ultes-Nitsche (Eds.), *Proceedings of the 5th International Workshop in conjunction with the 9th International Conference on Enterprise Information Systems (MSVVEIS 2007) 12-13 June 2007, Funchal, Madeira, Portugal* (pp. 126-137). INSTICC Press.https://doi.org/10.5220/0002428801260137

Appendix AH: *Gartner Hype Cycle which represents the adoption, maturity, and impact of emerging technologies.*



Source: Raza, M. (2020, May 5). *Introduction to the Gartner Hype Cycle*. BMC Blogs. Retrieved April 27, 2024, from https://www.bmc.com/blogs/gartner-hype-cycle/

Abstract:

Business Process Model and Notation (BPMN) is widely used for business process modelling because of its intuitive graphical notation and its ability to bridge the gap between business users and technical developers. This Master's thesis presents a critical examination of BPMN and compares it with Use Case diagrams and Colored Petri Nets.

The study begins with an in-depth analysis of BPMN, highlighting its strengths as a standardised and comprehensive approach to business process visualisation. Next, Unified Modeling Language (UML) Use Case diagrams are examined for their effectiveness in capturing functional requirements and their simplicity in representing user interactions. Petri Nets are then analysed for their robustness in modelling processes and their mathematical rigour in validating processes.

The comparative analysis reveals that while BPMN excels at providing a detailed, standardised description of business processes, UML Use Case offers a simpler approach to capturing user requirements, and Colored Petri Nets are a powerful tool for modelling complex and concurrent systems. The results suggest that the choice of modelling technique should be guided by the complexity of the processes, the audience for the modelling and the level of detail required.

This thesis contributes to the ongoing discussion on business process modelling by providing a complete evaluation of BPMN, UML Use Case and Colored Petri Nets, offering insight into their respective advantages and limitations. The comparative analyses are intended to guide practitioners in choosing the most appropriate modelling technique for their specific business process needs.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management
Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique
www.uclouvain.be/lsm