THESIS / THÈSE

MASTER IN BUSINESS ENGINEERING PROFESSIONAL FOCUS IN DATA SCIENCE

MERODExBPMN

WAEGEMAN, Amber; De Jaegere, Marian

Award date:
2023

Awarding institution:
University of Namur
KULeuven

KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

MERODExBPMN

Marian De Jaegere and Amber Waegeman

r0752079 and r0745043

Thesis submitted to obtain the degree of

MASTER OF BUSINESS AND INFORMATION SYSTEMS ENGINEERING

Promoter: Prof. Dr. Monique Snoeck
Co-promoter: Prof. Dr. Anthony Simonofski
Tutor: Charlotte Verbruggen

Academic year 2022-2023

KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

MERODExBPMN

English

In recent years, business process management (BPM) has become increasingly important. One subdomain of BPM concerns data-aware process modelling methods that try to extend traditional process modelling with additional data elements. Evaluation of these methods has raised concerns about their usability. Therefore, this thesis sets out to investigate the usability of the MERODExBPMN tool which is part of the MERODE approach, an enterprise engineering methodology that combines complete data- and process-awareness. To this end, an observational study was performed with 20 participants using the current tool interface and a prototype version implementing certain UI design principles. The most important finding is that the redesign according to UI design principles, most importantly feedback and structure, significantly improved the perceived usability of the tool.

French

Ces dernières années, le Business process management (BPM) est devenu de plus en plus important. Un sous-domaine de BPM concerne les méthodes de modélisation des processus tenant compte des données, qui tentent d'étendre la modélisation traditionnelle des processus avec des éléments de données supplémentaires. L'évaluation de ces méthodes a soulevé des inquiétudes quant à leur facilité d'utilisation. L'objectif de ce mémoire est d'étudier la facilité d'utilisation de l'outil MERODExBPMN ,qui fait partie de l'approche MERODE. MERODE est une méthodologie d'ingénierie d'entreprise qui combine une prise en compte complète des données et des processus. À cette fin, une étude observationnelle a été réalisée avec 20 participants utilisant l'interface actuelle de l'outil et un prototype mettant en œuvre certains principes de conception de l'interface utilisateur. La conclusion la plus importante est que la refonte de l'interface selon les principes de conception de l'interface utilisateur, en particulier le retour d'information et la structure, a amélioré de manière significative la convivialité perçue de l'outil.

Thesis submitted to obtain the degree of

MASTER OF BUSINESS AND INFORMATION SYSTEMS ENGINEERING

Promoter: Prof. Dr. Monique Snoeck
Co-promoter: Prof. Dr. Anthony Simonofski
Tutor: Charlotte Verbruggen

Academic year 2022-2023

Acknowledgements

First and foremost, we would like to thank our promotor Prof. Dr. M. Snoeck and our daily supervisor C. Verbruggen for their continuous support. At every step along the way, they were ready to provide guidance to steer us in the right direction.

Additionally, we want to express our gratitude to all 20 participants. They voluntarily spent an hour of their free time with us without receiving any remuneration in return. Without them, this study would not have been possible.

Lastly, we'd like to thank all relatives, friends and others who supported us during the development of this thesis.

Table of Contents

General Introduction

In this age of digital transformation, companies are looking for ways to integrate digital technologies into all areas of their business. Business Process Management (BPM) is seen as one of the key drivers of this transformation, causing the sector to grow rapidly with a forecasted compound annual growth rate of 12% in the coming years (Fortune Business Insights, 2023).

The application of BPM can help a company achieve operational excellence. Nevertheless, traditional BPM techniques have a major flaw: they are activity-centric, meaning that they focus on a company's activities while ignoring its data. This divide between activities and data leads to maintenance complexity and may even cause inconsistencies between a company's different information systems (Dumas, 2011). Hence, the data-aware process modelling paradigm emerged where BPM techniques are extended with data aspects.

Many different data-aware approaches, that enriched process models with data elements, were introduced. However, since this research was largely dominated by experts from the process modelling community, most methods remain flawed with regards to their data models (Snoeck et al., 2023). MERODE, an enterprise engineering method, is currently one of the most complete conceptual modelling methods that has the potential to be both fully data- and process-aware. At this moment in time, tool support is being developed to implement the conceptual link between business processes, more specifically BPMN, and MERODE's data model.

Usability is a general point of concern for all data-centric approaches. It takes two forms: modelling language usability and modelling tool usability. The former considers the modelling language constructs, such as the use of classes in UML. The latter investigates whether supporting modelling tool interfaces are usable, disregarding the usability of the underlying language. A systematic literature review (Ternes et al., 2021) revealed that very little research has been conducted in this second area of usability.

Therefore, this thesis sets out to investigate the modelling tool usability of the MERODExBPMN tool that is currently being developed. It is part of the tool suite that will support the link between MERODE and BPMN. The purpose of the study is to explore current usability concerns in the interface and determine potential improvements.

To this end, a qualitative observational study with 20 participants was carried out. They were given a printout of a solved case and had to use the tool to recreate what they saw on paper. During the entire experiment, participants were questioned about their experience to bring all possible usability issues to light. To conclude the experiment, they were asked to fill in a questionnaire to assess the overall usability of the system. The study made use of a multi-modal observation

approach, utilising audio, tool interactions and the post-experiment survey, to help capture the cognitive processes of the participants.

This thesis is structured as follows. Section 1 provides an overview of the related literature. Section 2 presents a detailed description of the MERODE method. Section 3 describes the study's methodology. Lastly, section 4 presents the results and discussion.

1 Literature review

To showcase the relevance of MERODE's integration with BPMN we start our literature review with the evolution of Business Process Management towards more data-awareness. The second part of the review shifts towards the usability of modelling tools and their interfaces since this is the focus of our research. To find literature in our areas of interest, we made use of Google Scholar. We tracked down other related works through the snowballing search method.

1.1 Business Process Management

The concept Business Process Management (BPM) has numerous definitions. Generally, it is seen as a management philosophy which helps to improve and maintain business performance by focussing on end-to-end business processes (Hammer, 2015; Toufah et al., 2020). Von Scheel et al. (2015) define a business process as "a collection of tasks and activities (business operations and actions) consisting of employees, materials, machines, systems, and methods that are being structured in such a way as to design, create, and deliver a product or a service to the consumer" (Von Scheel et al., 2015). They are commonly represented using business process modelling techniques such as BPMN (OMG, 2014) which provide a visual overview of the work organisation. If BPM is employed correctly, it can help companies achieve and sustain a competitive advantage (Hung, 2006).

Within the literature, there are many different views on the exact definitions of the phases of the BPM lifecycle, but they all overlap to a large extent (Houy et al., 2010). The following lifecycle overview is based on Hammer (2015). According to him, BPM starts by formalizing the processes within an organisation, a highly significant step considering that many organisations have variable operations that lack any form of formal definition. After the formalisation phase, processes need to be continuously managed. Their performance will be measured in terms of KPIs and compared to previously defined targets. If the KPIs fall short of the norm, the root cause should be determined and eliminated. After implementing the solution, the cycle starts all over again. Hoey et al. (2010) provide a concise summary of this lifecycle. They state that it starts with the definition and modelling of processes according to an organisation's strategy. Those processes then get implemented, executed and monitored. If issues arise, improvements will be made.

Over time, BPM has increasingly gained popularity, both in industry and research, as a tool to achieve operational efficiency. According to Havey (2005), there are multiple factors that drive organisations towards BPM: formalizing existing processes, detecting necessary improvements, facilitation of efficient process flows, increase of productivity, analysis of complex problems and regulatory compliance. Even though, an organisation's initial focus may be narrow, applying

BPM will yield a very broad array of benefits. By eliminating non-value adding overheads within the end-to-end processes, BPM tends to reduce costs and assets, whilst increasing speeds, accuracy and flexibility (Hammer, 2015).

1.2 Data-aware process modelling

During the 90s, BPM techniques increasingly got adopted by companies looking for better IT support. However, following this surge in adoption, some issues started to manifest themselves. Traditional BPM methods are activity-centric, meaning that they focus on the company's activities and their order of execution, while ignoring the organisation's data. Consequently, traditional BPM fails to capture the business context, which contains the essential business concepts, their behaviour and relationships (Liu et al., 2007).

It is important to note that most companies that use BPM also employ data engineering techniques. However, there is a lack of integration between these two aspects of enterprise engineering. According to Dumas (2011), this divide between the handling of data and processes causes two potential forms of redundancy: process/function-related data redundancy and business rules redundancy. The former occurs when the business process management system and the database system both store data about the state of the process. The latter arises when business rules are encoded within both the process layer and the database layer. These two forms of redundancy add additional maintenance complexity and may cause inconsistencies.

Once the divide between data and processes was recognized, numerous approaches were introduced that attempted to combine both aspects in a single method. As such, the data-centric modelling paradigm emerged. It is worth noting that most of these methods have been created by experts from the process modelling domain who extended process models with data elements (Snoeck et al., 2023). As a result, these methods consist of a full-fledged process model combined with a slightly flawed data model. According to Steinau et al. (2019), these approaches still lack a general understanding of the relationships between processes and data. They found that most data-centric methods only take modelling aspects into account, ignoring the rest of the process lifecycle. Nevertheless, despite their current shortcomings, the quality of data-centric approaches has been confirmed. In a study by Reijers et al. (2016), practitioners and students corroborated most claims about the functionality of three data-aware approaches. Although it was a small study that only investigated a limited number of approaches, it clearly showed the potential value of the data-centric paradigm.

As stated earlier, there is a wide variety of data-aware process modelling techniques. In a literature review by Steinau et al. (2019), 17 distinct data-aware process modelling approaches have been identified. The authors observed large variability in data representation constructs, behaviour description, interaction

description, process enactment and management of process granularity. As a result, the data constructs of most data-aware methods cannot easily be mapped to existing standardized data modelling practices (Snoeck et al., 2023).

1.3 Usability

One of the major problems users of data-aware modelling languages experience, is a lack of usability. According to the ISO 9241-11 standard, “usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO, 2018). It is important to keep usability in mind when developing a new tool to ensure its adoption by the prospective users.

The following paragraph is based on a 2016 study by Reijers et al. that evaluated data-aware process modelling approaches. According to this study, over the last decade, most new data-centric approaches have focussed on creating distinct modelling notations and design procedures. Within papers concerning the methods, creators make various quality claims about the usability, functionality, efficiency etc. However, none of these claims are substantiated by practitioners as they are merely opinions of the creators. Consequently, Reijers et al. (2016) set out to check whether the needs and preferences of modellers are being met and whether the approaches deliver on their promises. The study defined five quality categories: functionality, usability, efficiency, maintainability and portability. They then identified the claims about three different approaches and sorted them into the previous categories. To evaluate the veracity of the claims, a workshop setting was used where the participants discussed the different quality aspects. In most categories there were mixed results with some claims being supported whilst others were not. However, usability turned out to be a major source of concern across all considered methods. They were considered too complex and non-intuitive. Therefore, the study concludes that new data-centric approaches should focus on the user needs and design the methods around them.

The previous paragraph mainly focussed on the usability of modelling languages. This area of usability considers the modelling language constructs such as, for example, the use of classes and associations in UML. There exists, however, another type of usability, namely modelling tool usability. This perspective examines how usable supporting modelling tools are without considering the usability of the underlying language such as UML or BPMN (Pietron et al., 2018). Very little research has been conducted in this area (Ternes et al., 2021). One example of a study concerning this perspective is a comparison of six distinct UML modelling tools in terms of their usability (Bobkowska & Reszke, 2005). According to the authors, proper usability should enable users to focus on their work instead of on how to use a tool. However, since very little research has been conducted, there exist no universal design recommendations or standards to guide modelling tool developers during the creation of their tool (Ternes et al., 2021).

2 MERODE

The MERODExBPMN tool whose usability we'll be investigating provides support for the integration of BPMN into MERODE which creates a fully data-aware process modelling method. To ensure readers have a minimal understanding of the approach, we will begin by giving a more in-depth explanation of the MERODE methodology which is in essence an enterprise engineering method. Then we will elaborate on the integration of BPMN which makes the method both fully data- and process-aware.

Seen as MERODE got developed by a KU Leuven research team and has not received widespread academic attention, there is little literature to be found on the subject that doesn't originate from KU Leuven. The information in the following sections about MERODE is primarily derived from a textbook written by the main researcher (Snoeck, 2014).

2.1 General introduction to MERODE

MERODE is an enterprise architecture methodology that enables creating and validating models during information systems development. By modelling different aspects of the organisation and linking them together, the method helps gaining insight into the fundamental building blocks of the enterprise. MERODE follows a layered approach. It starts off with the domain modelling phase where business objects, events and lifecycles are defined. Subsequently, information system services get identified. Lastly, the business processes, which consider work organisation, are modelled. All of these aspects are linked together to create a consistent overview of the entire enterprise.

An important feature of MERODE is its model-driven engineering approach which focusses on creating models as representations of the required software instead of manually writing code. A major advantage of this approach is its flexibility. Once the templates, that enable the transformations from model to code, have been created, changes to the model can instantaneously be translated to new code. As a result, coding and testing becomes much less time intensive. Errors in the implementation can be immediately resolved by updating the model and regenerating the code. If the templates have been carefully debugged, there is the additional benefit of zero coding errors occurring in the generated code. Moreover, model-driven engineering allows for the integration of coding best practices into the templates (Cabot, 2020). It is worth noting that model-driven engineering is being replaced by the no-code/low-code movement which in not identical but conceptually very close (Di Ruscio et al., 2022). According to Di Ruscio et al., all model-driven approaches employ models during the development cycle, but don't necessarily include automatic code generation. In contrast, the low-code movement specifically focusses on decreasing manual coding without the absolute

requirement to utilise models. The no-code movement goes even further by requiring end-users to write absolutely no code at all. However, according to Cabot (2020), the two disciplines largely overlap with low/no-code simply being the new differently branded version of model-driven engineering techniques.

The MERODE approach is not just a purely conceptual idea. It is supported by Merlin (Snoeck M., 2020), an online tool where users can create domain models, and a code generator environment which uses the Velocity Template Engine (The Apache Software Foundation, n.d.) to automatically generate a corresponding prototype application. Merlin supports the entire domain modelling phase as defined in the MERODE approach: it allows modelling the existence dependency graph, object interaction and object lifecycles. We will further elaborate on these modelling elements in subsection 2.3. Once the domain model has been created, it can be exported as an XML file which can be used as input for the Code Generator, the second supporting tool. In just a few clicks, the domain model gets transformed into a basic prototype application that allows testing whether requirements are properly implemented. The generated prototype provides feedback when a user tries to perform an action that is not allowed by the domain model, which improves modellers' understanding of whether their model fulfils the requirements.

2.2 MERODE layers

MERODE is organized into three layers. Each layer is aware of the more inner layers upon whose features it can call, whilst remaining completely oblivious to the more outer layers. Consequently, modifications in the outer layers won't affect the inner layers. Effects of modifications in the inner layer, by contrast, will spread to the entire system. Hence, functionalities that rarely change over time are positioned in the kernel. The more a layer is positioned towards the outside of the system, the more flexible it is. In the most outer layer, changes are regularly made. Due to its separate layers, MERODE provides a high degree of adaptability. If a business process or information system service needs to be updated, this can easily be achieved without interfering with the core of the system.

Figure 1 displays MERODE's layers. The kernel is the enterprise layer which holds the essential business concepts in the form of the domain model. It contains the business objects, their lifecycles and business events. The second layer is the business information system services layer which contains input and output services. Input services allow users to inform the enterprise layer of business events that occurred in the real world. Output services, by contrast, do not

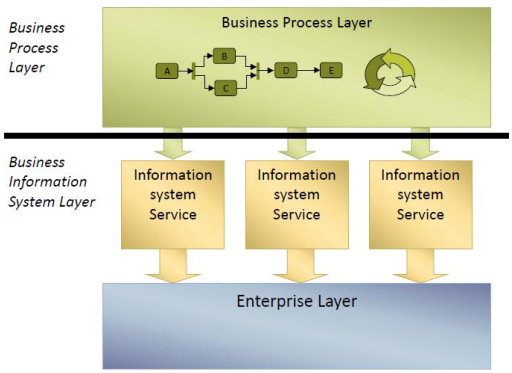


Figure 1: Architectural layers (Snoeck, 2014, fig 2.1, p.33)

make any changes. They are used to extract information from the enterprise layer. Lastly, the most outer layer is the business process layer which defines the work organisation.

2.3 Domain modelling

The MERODE domain model consists of two major components: business object types and business event types. Their characteristics and relationships with each other get defined in three separate sub-models: a data model, an interaction model and life cycles for the business objects.

A business object type is a real-world relevant business concept. It contains a set of similar business objects that can all be identified and described by a number of attributes. An example is the type ‘customer’ which encompasses all customers of an organisation. The second concept, a business event, corresponds to an atomic real-world event that has an impact on at least one business object. It can create, end or modify said object. Every business object type should have at least one creating event and one ending event.

The data model, called the existence-dependency graph (EDG), is a restricted version of a UML class diagram. Its main refinement is the obligation for every class to be part of an existence dependent relationship with another class. Existence dependence is defined as follows: “If each object of a class A always refers to minimum one, maximum one and always the same occurrence of class B, then A is existence dependent of B” (Snoeck & Dedene, 1998). A is called the dependent object type, while B is called the master object type. In the EDG, every object type has to be either a master or a dependent which can easily be achieved by creating a UML class diagram and then continuously reifying all associations that do not express existence dependency. The advantages of this method are improved consistency checking and simpler transformation to code.

Object interaction is captured by the object-event table (OET) which maps the object types of the EDG against the identified event types. In essence, it models object type involvement in event types. It is formatted as a table with the event types as rows and object types as columns. If an object is involved in an event, this will be indicated in the corresponding cell in the table along with the type of involvement: creating, ending or modifying. The EDG is connected to the OET by means of the propagation rule which specifies that a master object type is involved in all event types of its direct and indirect dependent(s).

Finally, lifecycles are modelled using finite state machines (FSMs). They are used to impose sequence constraints on the execution of different events during the lifetime of a business object. It is, for example, impossible to carry out the event ‘ChangeCustomerInformation’ if ‘CreateCustomer’ hasn’t occurred yet.

2.4 Information system services modelling

The information system services layer serves as an intermediate layer between the domain model and the business processes, providing information system services that support the execution of activities in the form of input and output services. The former collect information about real-world events and signal that information to the domain model to ensure it mirrors the real world. The latter query the enterprise layer to extract information needed for the execution of a business process.

There are two types of services: trivial and complex services. Trivial services consist of exactly one business event (trivial input service) or one query (trivial output service), while complex services are composed of multiple trivial services to offer more intelligent process support. A process actor can call upon these complex services when they need richer functionality than the basic trivial services provide.

It is important to note that the input services do not directly manipulate business objects. MERODE provides an intermediate interface called the event-handling layer (EHL). The positioning of this layer is displayed in Figure 2. Input services use the EHL to trigger business events which will then be further handled by this layer. As a result, MERODE combines the advantages of an event-driven architecture with those of the layered architecture (Snoeck et al., 2023).

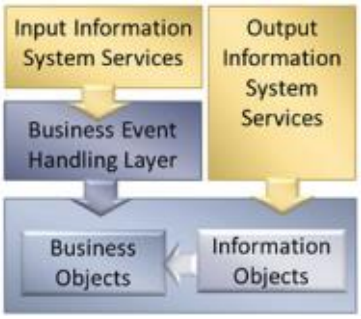


Figure 2: Information system services layer (Snoeck, 2014, fig 9.8, p.215)

2.5 Business process modelling

A business process presents a complementary view to the domain model. It focusses on the work organisation whereas the domain model focusses on modelling and defining the data required in the business processes. The two perspectives are connected through a mapping table where per process all process tasks are identified and for every process task the required input and output services are listed (Snoeck et al., 2023). It is possible that some tasks do not need a mapping: they do not require domain layer elements as their execution does not interact with any data. Due to this simple mapping table, adding or changing business processes becomes straightforward: users only have to add, modify or delete rows within the table while the domain layer itself remains untouched.

Up until recently, the connection between a business process model and the domain model was mainly conceptual without a real implementation. However, an integration of the Camunda BPMN platform (Camunda, n.d.) with the MERODE

domain model is currently in development. The following paragraph will be based on the paper (Snoeck et al., 2023) where the new technique gets introduced.

Next to generating Java applications, which contain default information system services, the Code Generator can also wrap the enterprise layer and expose its services as REST web-services. Instead of using the original Java interface, users can now utilise Camunda TaskForms and Service Tasks which call on the REST web-services to manipulate business objects in the enterprise layer. At present, the mapping table is still an informal requirements engineering instrument. Hence, the TaskForms and Service tasks that link the Camunda Process engine to the underlying MERODE domain model have to be created manually.

To support automated generation of TaskForms and Service Tasks, the MERODExBPMN tool, which is the focal point of this thesis, is being developed. The tool will enable users to link BPMN process tasks with their required MERODE information system services. The user will also be able to use the tool to create complex input services in case a BPMN task triggers multiple business events simultaneously. The task can then be linked to this newly defined complex service. Once the mapping has been completed, the tool can generate an XML file containing said mapping.

2.6 Current tool interface

Figure 3 visualises the current MERODExBPMN interface. An additional visualisation, after uploading the MERODE and BPMN files, has been included in appendix 1. Technically, the tool functions as it should, but no UI design principles were taken into account when designing the layout. Therefore, the usability of the tool should be investigated and improved upon if necessary.

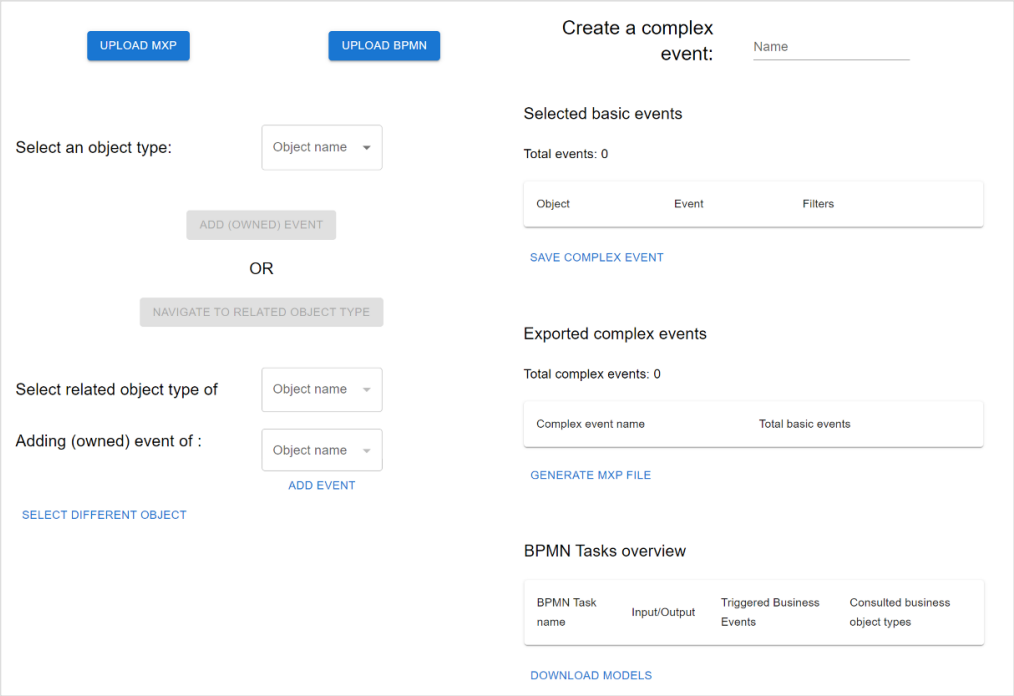


Figure 3: current MERODExBPMN interface layout

3 Methodology

To investigate the usability of the MERODExBPMN interface, 20 one-on-one usability tests were performed during March 2023. All participants are students who completed the course ‘Architecture and Modelling of Management Information Systems’ in 2021 or 2022. This course teaches the basic principles of enterprise architecture and focuses on the practical use of this information through enterprise modelling using the MERODE method (KU Leuven, n.d.). Because it has been a while since these students learned about MERODE, every experiment starts off with a short refresher on the entire modelling method with a focus on the business process layer. Afterwards, participants are asked to fill in a short quiz (appendix 2) about the most important concepts in the MERODE method to assess their understanding of the methodology.

Before a participant gets to interact with the tool, they are given a case description (appendix 3) and the corresponding solution (appendix 4). They are guided through the entire solution to ensure they fully understand it. As a result, any usability issues that come to light during the modelling process are unlikely to stem from a lack of understanding of the underlying methodology or the used case.

During the experiment, participants are asked to create two complex services and perform their mapping in the tool. They have the case solution next to them at all times and only have to find a way to recreate what they see on paper within the tool. At each step of the modelling process, the participants are asked questions to assess whether they comprehend the tool’s functionality or encounter any usability issues. Half of the participants will perform their experiment with the current tool while the other half will receive a prototype version implementing UI design changes.

To conclude the experiment, participants fill in a slightly adapted version of the SUS questionnaire (Brooke, 1996) to assess the perceived usability of the system.

3.1 Research questions

We identified five main research questions to investigate during the experiments.

RQ1: What is the perceived usability of the current/prototype interface?

RQ2: Is there a difference in perceived usability between the current interface and the prototype interface?

RQ3: Which usability issues occur frequently in (one of) the two layouts?

RQ4: Which improvement suggestions are frequently proposed in (one of) the two layouts?

RQ5: Do any of the UI design changes in the prototype get validated by the participants?

3.2 Multi-modal observation

Studies about conceptual modelling are methodologically challenging since cognitive processes and deliberations on decisions are not directly observable (Rosenthal, Ternes, et al., 2020). Therefore, researchers have to rely on other observable aspects of the modelling process such as modeller's interaction with the software.

This thesis adopts a multi-modal observation and data generation approach which combines complementary modes of observation to capture the cognitive process of modellers as proposed by Rosenthal, Ternes et al. (2020). Their underlying assumption is that multi-modal data collection leads to a broader understanding of the individual's modelling process and mitigates the limitations of every separate mode of observation (Rosenthal, Strecker, et al., 2020).

In these experiments, three modes of observation are combined. Audio recordings capture participant's answers to the comprehension questions, screen recordings capture the modelling actions and a survey assesses the overall perceived usability. The survey results will be used to answer RQ1-2 while the audio and screen recordings are crucial in answering RQ3-5.

3.3 Comprehension questions

During the usability test, we ask the participants questions as they perform the tasks we have set. The goal of the questions is to gain insight into the participants' thought processes. Before actually interacting, a five second test (Doncaster, 2014) is performed where the participants' first impressions are elicited after seeing the layout for five seconds. Afterwards they gain control over the tool and start to interact with it. Before performing an action, participants will be questioned about how they want to proceed. If a participant is able to accurately describe the following steps, it shows that the tool intuitively guides the user during the modelling process. After the action has been performed, questions will be aimed at the participant's experience and potential issues they experienced during their modelling interaction. The complete list of comprehension questions can be found in appendix 5.

3.4 Prototype

To enrich our research, we developed a new interface layout in the prototyping software Figma (Figma Inc, 2023) based on UI design principles (Ruiz et al., 2021). To avoid any confusion, the rest of this thesis will refer to the existing interface as interface A while the prototype version will be called interface B. Figure 4 and 5 visualise interface B before and after uploading the MXP and BPMN model.

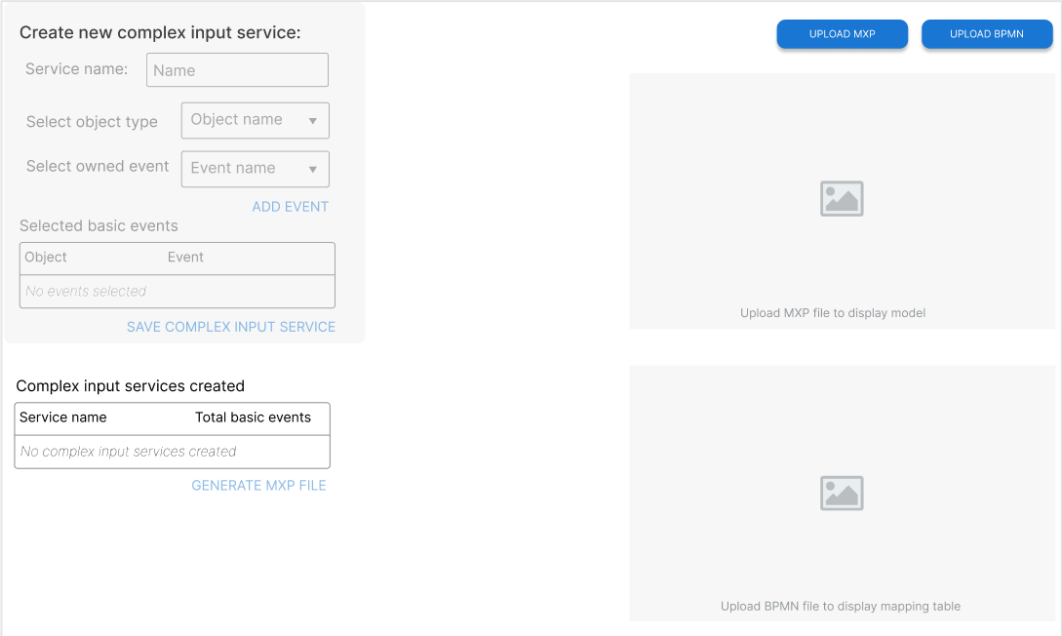


Figure 10: Interface B before uploading

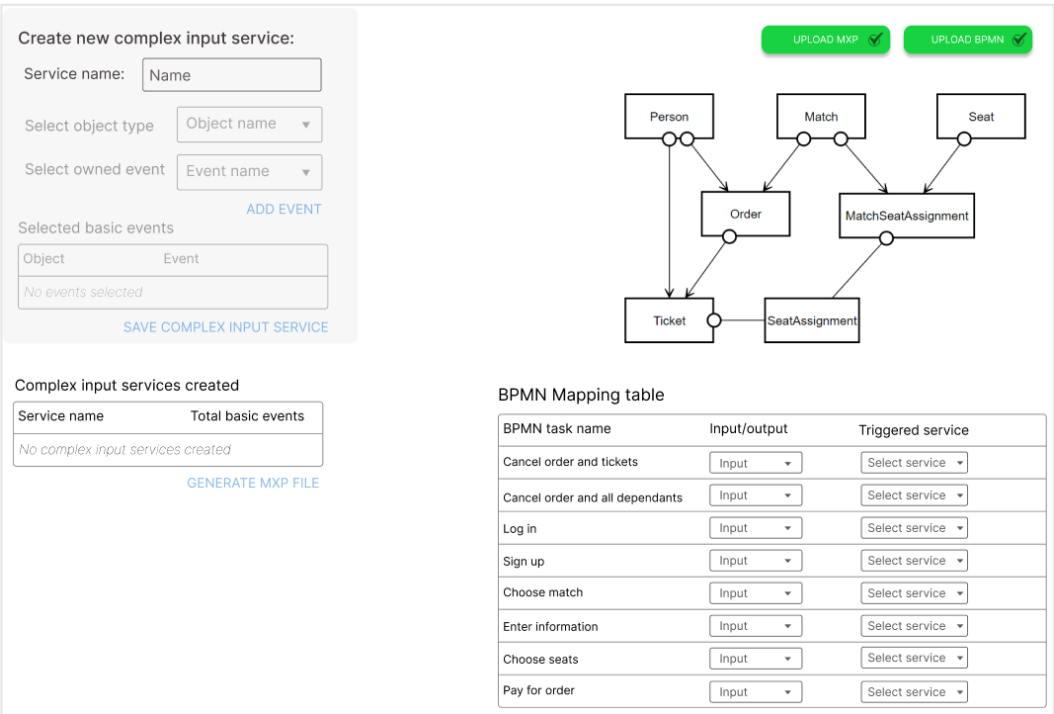


Figure 11: Interface B after uploading

The most important design principles that got implemented are affordance, feedback, structure, consistency and error prevention.

Affordance (Norman, 1988) means that features should contain clues on what they can be used for and when they can be used. Interface B accomplishes this in two distinct ways. First, inactive features are faded to imply they cannot be used, which is visualised in Figure 6. Second, the number of features was reduced by leaving the 'navigate to related object type' button and its corresponding dropdown out of the prototype. Figure 7 showcases the resulting simplification of the interface which makes the functionality of the remaining features more straightforward.

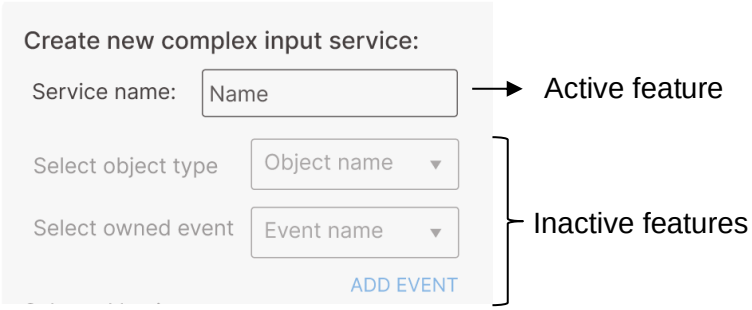


Figure 12: Affordance by fading inactive features

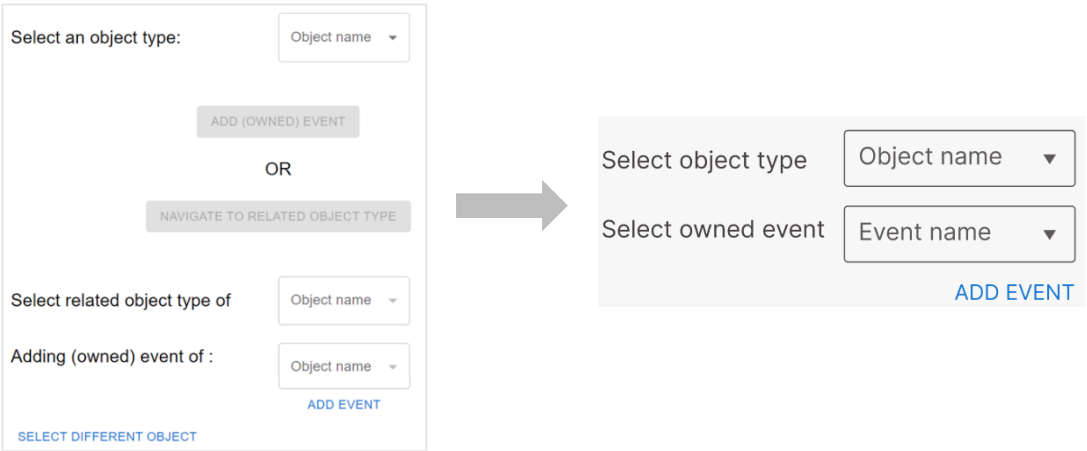


Figure 13: Affordance by reducing the number of features

The second principle, feedback, is straightforward: when an action is performed, the result should be visible (Shneiderman, 1997). In interface B every action always has a clear result which is not the case in interface A. Feedback is implemented in three ways: upon uploading the models the corresponding upload buttons turn green as shown in Figure 8, features that are activated during the modelling process become unfaded as shown in Figure 9 and, lastly, both the EDG and the mapping table are partially visible within the first frame such that participants will see the visualisations popping up upon uploading without having to scroll first which is required in interface A.

Figure 14: Feedback through changing colours

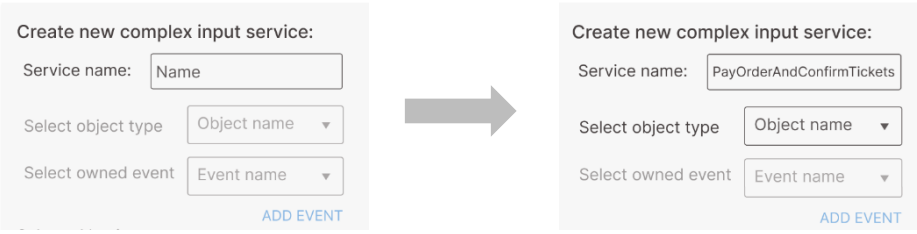


Figure 15: Feedback through unfading features

The third principle is about structuring the interface: all elements that need to be used within the same flow, are grouped close together (Norman, 1983a). Figure 10 visualises the grouping of elements related to uploading on the right and elements related to creating a complex service on the left.

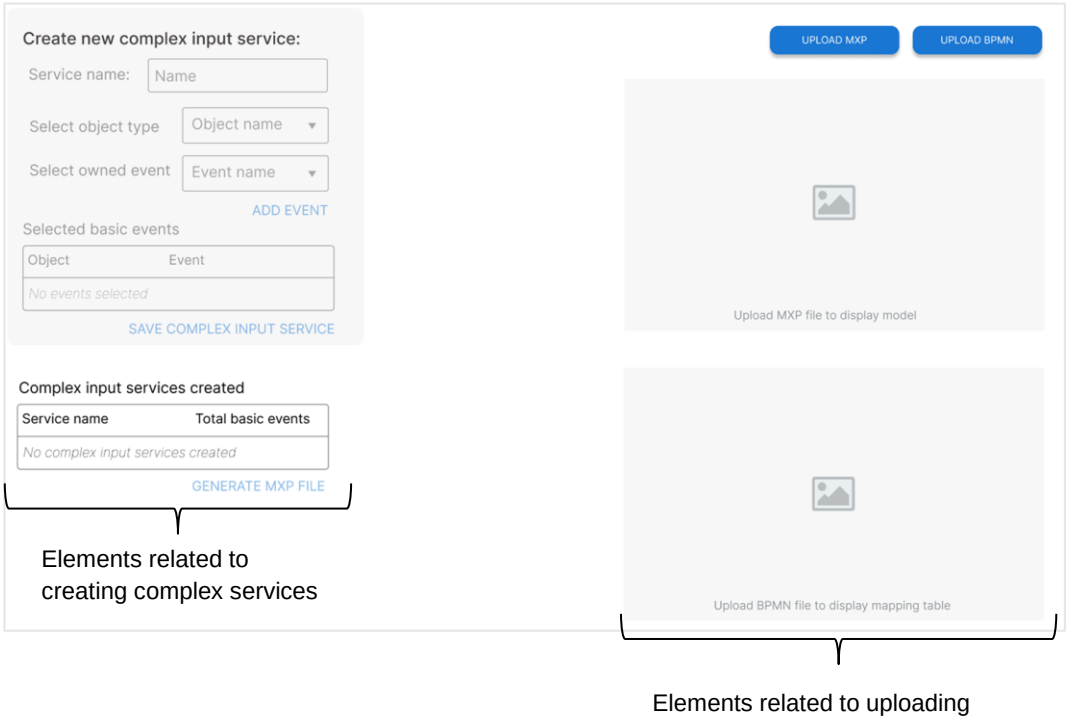


Figure 16: structure through grouping related features

The fourth principle is consistency which ensures that users don't have to figure out whether different terminology, situations or actions refer to the same thing (Nielsen, 2005). Interface B has been made more consistent by including placeholders in the positions where the model visualisations will appear after uploading them. Lastly, the combination of all of the above principles helps to prevent errors (Norman, 1983b).

3.5 SUS questionnaire

The System Usability Scale (SUS) (Brooke, 1996) is used to measure the perceived usability of a system. It is a standardized questionnaire consisting of 10 questions that can be aggregated into a composite measure of the overall usability of a system. This thesis makes use of an adapted nine-question version that is positively worded (Appendix 6). According to Lewis (2018), leaving out an irrelevant question doesn't affect the validity of the SUS score and keeping all questions positive prevents the participants from making scoring errors.

SUS scores range from 0 to 100, but they are not entirely comparable to percentages since respondents tend to stay away from extreme responses in a survey. Hence, very low or high scores are less likely to emerge. To enhance the understanding of the obtained score, this thesis will compare them to SUS norms that assign it a grade ranging from A to F (Lewis & Sauro, 2018).

4 Results and discussion

4.1 Quiz scores

Table 1 contains a detailed overview of all quiz responses.

Participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Layout	A	B	A	B	B	A	A	B	B	A	A	A	B	A	B	B	A	B	B	A
Q1	1	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q2	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3
Q3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Q4	3	2	2	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2
Q5	3	3	2	2	1	3	3	2	3	1	3	3	2	2	3	3	3	3	3	3
Q6	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Q8	2	1	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	1
Total	7	6	6	6	7	8	8	7	7	5	8	8	7	7	8	8	8	8	8	7

Table 1: quiz responses

The quiz scores were highly similar over the two experimental groups: both groups answered on average 90% of the questions right, with a median of 94% for interface A and 88% for interface B.

These high scores suggest that participants have a thorough understanding of MERODE. Even the lowest obtained score of 63% still signals a decent understanding. Additionally, there is no significant difference in scores between the two experimental groups. Therefore, any differences in perceived usability will not stem from one group having a worse understanding of the underlying methodology.

4.2 SUS scores

Table 2 provides a detailed overview of the SUS questionnaire responses.

Participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Layout	A	B	A	B	B	A	A	B	B	A	A	A	B	A	B	B	A	B	B	A
Q1	2	5	3	4	4	2	1	4	4	2	2	5	4	2	4	4	4	4	4	4
Q2	3	5	2	5	4	2	1	4	4	2	2	3	3	2	4	4	3	5	4	3
Q3	4	4	3	5	2	1	1	3	5	3	1	3	5	5	4	4	3	5	5	5
Q4	3	5	3	4	5	3	2	4	4	4	3	2	5	2	4	5	3	4	4	4
Q5	2	5	4	5	5	4	1	3	5	2	5	2	5	2	5	3	2	5	4	3
Q6	5	5	4	5	4	4	3	4	5	4	4	3	5	1	4	5	4	4	5	5
Q7	3	5	2	5	1	1	1	3	5	3	3	1	3	1	4	4	3	5	4	3
Q8	2	4	3	4	2	2	1	5	5	2	3	2	4	3	4	3	4	3	5	3
Q9	4	4	5	5	5	2	4	4	5	2	4	4	4	3	4	4	5	1	5	4
SUS score	53	92	56	92	64	33	17	69	92	42	50	44	81	33	78	75	61	69	75	86

Table 2: SUS questionnaire responses

The aggregated SUS scores differ significantly: interface A received an average score of 45.06 while interface B's average score amounted to 80.28. Comparison to SUS norms assigns them an F and an A- respectively.

It is clear that the combination of design changes in the prototype has a major positive impact on the perceived usability of the interface. The new layout also seems to have reached a level of usability that's acceptable to end-users considering that the industry has taken 80 or more as a benchmark of above average user experience (Lewis & Sauro, 2018).

4.3 First impressions

Every experiment started off with a five second test to elicit the participants' first impressions on the two layouts. There was a noticeable difference in first reactions between the two experimental groups.

Participants who received interface A expressed more negative impressions than positive ones. The most common concerns were the lack of structure (6 out of 10 respondents), unclear flow (5), too much going on (5) and the ugliness of the interface (4). It should be noted that two participants did not agree with the first major concern since they labelled the interface as structured. All other positive comments were limited to one participant only: clean, minimalistic, modern and good-looking.

Interface B's first impressions were more mixed. The most frequent concerns were the lack of colour (3) and the purpose of the placeholders not being immediately understood (2). Participants made more positive comments compared to the current layout: structured (3), clean (2), informative (2) and good-looking (2).

The UI design changes clearly had a positive effect on users' first impressions since most common concerns in interface A, namely structure, crowdedness and looks, actually became strengths of interface B. The result for structure can be quantitatively supported since all participants were asked to rate the structure of the interface on a scale from 1 (chaotic) to 5 (well-structured). One participant did not answer the question so their score was imputed with a three. The current layout received a score of 2.12 while the prototype received a score of 3.84. Hence, the combination of UI design changes clearly improved the perceived structure of the interface.

4.4 Frequent usability issues

For both layouts, all distinct issues that participants ran into when interacting with the tool, were identified. Table 3 and 4 visualise the issues in interface A and B respectively, in order of their occurrence in the modelling flow.

Problems that occurred four times or more were determined to be frequent. Within interface A, twenty-two distinct issues were encountered with nine frequent occurrences. By contrast, participants who received interface B, experienced sixteen distinct issues with five of them being frequent. We will discuss the most interesting findings in the following paragraphs.

ID	Element(s)	Description	Frequency
1	upload MXP and BPMN	Forgets to upload models before starting the modelling process	3
2	upload BPMN	Doesn't see any changes upon uploading BPMN	10
3	upload BPMN	Expected a BPMN visualisation	4
4	name input	Forgets to type in the name	3
5	name input	Doesn't know whether the name has been accepted by the tool (no confirmation)	2
6	name input	Tries to use the name input to generate objects/events	2
7	select object type	Assumes the select object type dropdown is inactive due to its light grey colour	1
8	select object type	Does not understand owning object type should be indicated before choosing events	2
9	2nd time selecting object type	Runs into bug where object type needs to be deselected and reselected	6
10	navigate to related	Gets confused by the navigate to related object type button	7
11	navigate to related	Does not use the navigate to related object type button at all	5
12	adding owned event	Does not recall this button has to be pressed after doing so earlier	3
13	both blue buttons	Gets confused by the complete lack of feedback upon clicking the buttons	10
14	add event	Clicks add event without selecting an event which adds an empty event (bug)	4
15	add event	Does not know whether add event is an intermediary or a finalising button	2
16	add owned event vs add event	Does not intuitively understand the difference between the two buttons	5
17	select different object type	Does not notice the button when it is needed to be able to continue modelling	2
18	select different object type	Clicks it without being navigated to a related object type which leads to a buggy situation	1
19	tasks overview	Tries to start the modelling process in the tasks overview	3
20	scrolling down	Scrolls down too quickly which crashes the tool	1
21	general	Does not know which feature to use next since the flow is unclear	9
22	general	Gets stuck due to a bug	2

Table 3: Issues interface A

ID	Element(s)	Description	Frequency
1	upload MXP and BPMN	Forgets to upload models before starting the modelling process	4
2	upload BPMN	Expected a BPMN visualisation	8
3	placeholder	Clicks on the placeholder to upload the corresponding model (instead of the buttons)	2
4	placeholder	Does not understand what the placeholder is for	3
5	add event	Does not know whether add event is an intermediary or a finalising button	5
6	add event	Expected to add multiple events at once instead of adding them one by one	5
7	add event	Does not recall this button has to be pressed after doing so earlier	1
8	add event	Did not expect 'add event' would add the selected event to the selected basic events table	2
9	add event & save complex…	Does not immediately notice that these are buttons instead of just plain text	1
10	complex input services created	Tries to start the modelling process in the complex input services created table	2
11	mapping table	Thinks default option 'input' requires you to choose some sort of input	1
12	mapping table	Tries to start the modelling process in the mapping table	3
13	mapping table	Takes a while to find complex services since they are on bottom of the services list	5
14	mapping table	Wants to drag and drop complex services from the table with created complex services to the mapping table	1
15	general	Does not know which feature to use next since the flow is unclear	2
16	general	Runs into a bug due to usage of a prototype	3

Table 4: Issues interface B

4.4.1 Feedback

The most remarkable results are related to feedback. Every single participant who received interface A experienced two issues linked to a lack of feedback. They did not see any changes upon uploading the BPMN model, which led them to believe that it was not uploaded correctly, and they all got confused at the complete lack of feedback upon clicking the blue buttons, 'add owned event' and 'navigate to related object type'. As a result, participants generally did not understand that clicking those buttons activated other UI features underneath. When they stumbled across these activated features later on in the modelling process, most participants did not link their activation to the buttons they had clicked earlier. Modellers' lack of understanding about what the button 'add owned event' does exactly, caused an additional issue: the difference between the 'add owned event' button and the 'add event' button was seen as unclear by five participants. Since they did not understand the function of the 'add owned event' button, they assumed it would do the exact same thing as the 'add event' button.

Within interface B, not a single usability issue relates to feedback which clearly validates the UI design changes that implemented more feedback.

4.4.2 Navigate to related object type button

One of the most drastic changes in interface B was the removal of the ‘navigate to related object type’ button (and other related UI components). It should be noted that a modeller can use the tool without these features but their elimination does have some negative implications for the implementation of the tool, more specifically for the generation of the final XML file.

Seven participants who received interface A expressed confusion surrounding the ‘navigate to related object type’ button. Especially at first glance, they did not understand what the button’s purpose was and whether they were supposed to use it. Five participants actually completely ignored the button during the modelling process. Three of those who ignored the button did not understand its functionality, but the other two modellers understood the purpose perfectly. When questioned why they decided to ignore the button, both participants answered that it simply seemed unnecessary. They did not feel like going through the trouble of getting to know a new functionality when they could easily perform the tasks they were set with the other features they had already used.

Leaving the navigation option out of interface B clearly prevented a lot of confusion and simplified the flow of the tool. So, on the one hand, it is a usability improvement. However, on the other hand, its removal degrades the implementation of the tool. Therefore, care should be taken when designing the next version of the interface to support a correct implementation without compromising the usability.

4.4.3 Flow

A major problem in interface A is the lack of a clear flow that intuitively guides the user. In both layouts, some participants, three in interface A and four in interface B, try to start the modelling process within the mapping table. This seems to indicate that neither layout has an intuitive starting point. However, once the correct starting point had been found, the differences in experienced flow issues were remarkable: within interface A, nine participants struggled figuring out which interface features should be used at what times. In interface B, only two participants clicked on a wrong interface feature after figuring out the starting point.

These results suggest that the UI redesign had a major positive impact on the perceived flow. We attribute the improvement to a number of factors: the addition of feedback that clearly guides the user to newly activated features, the removal of the ‘navigate to related object type’ button which simplifies the modelling process and the grouping of related interface features which ensures that all features within the same flow are positioned underneath each other.

4.4.4 Add event button

Half of the participants who received interface B, thought the ‘add event’ button was a finalising button instead of an intermediary step. By contrast, in interface A, only two participants experienced this issue. This may be due to the next part of interface B’s flow still being faded. The next button that will become activated is ‘save complex input service’ which is clearly the finalising step, but since it is slightly faded, it may be difficult to read.

Another issue in interface B that seems to be somewhat correlated with the previous one is that five participants, three of which did not explicitly experience the above problem, wanted to add multiple events at the same time before pressing ‘add event’. They suggested a myriad of ways in which this could be implemented, but all of the options came down to selecting all necessary events before pressing ‘add event’. We would like to quickly note that this is impossible as the order of execution of the events in a complex input service is important and should be compatible with the FSMs. However, this issue clearly signals that they assume the ‘add event’ button is finalising. Therefore, we believe we can extend the amount of participants experiencing confusion surrounding this button to eight. Clearly, this area should be further improved by either creating a more explanatory name or making the next steps in the flow less faded to ensure participants understand the intermediate nature of the ‘add event’ button. In our opinion, a name change would be preferred since making the following steps less faded would interfere with the feedback mechanism that unfading provides.

4.4.5 BPMN visualisation

Upon uploading the BPMN file, both layouts visualise the mapping table. However, in both experimental groups there were a number of candidates who got confused since they had expected to receive the BPMN diagram. This problems seems to have been exacerbated by the inclusion of placeholders in the prototype: within the current layout, four participants had expected the diagram, while in the prototype, eight participants expressed this expectation. The image icon inside of the placeholder seems to suggest that a picture will appear instead of a table. Moreover, the text inside of the placeholders which clearly states which information will be visualised was deemed to small causing most participants not to pay attention to it. Therefore, the placeholder text should become more centred and bigger to avoid the above confusion.

4.5 Frequent improvement suggestions

During the experiments, participants were encouraged to suggest changes to the interface that would make it more usable to them. Table 5 and 6 list all explicit suggestions made in interface A and B respectively.

Since participants had to explicitly state their suggestions without interference from the interviewer, the observed frequencies are quite low. Therefore, we determine ideas that occur three times or more to be frequent, which yields seven frequent suggestions for interface A and two for interface B. To provide some structure, suggestions were divided into five categories that will be elaborated on in the following paragraphs.

FEEDBACK	
Suggestion	**Frequency**
Visual indicators to show successful upload (one suggested green colour)	4
Visual indicators to show results of modelling actions	3
Visual indicators to highlight selected basic events upon clicking add event	2
Confirmation button to finalize the name input	2
Visible indication on borders of dropdowns upon activation	1
LAYOUT	
Suggestion	**Frequency**
Add…/Navigate… buttons adjacent + respective dropdowns close by	3
Select different object type button next to object type dropdown	2
Name input next to the save button	2
Colour to guide the user	2
Make the triggered event dropdown box wider	2
Differentiate between add owned event button and add event button	2
Dividers between different parts of the tool (left/right & top/bottom)	1
Differentiate between upload buttons and add../navigate buttons (different look)	1
Headings to indicate different parts of the tool	1
Include a dropdown to choose add owned event/navigate option => Only the dropdown for the chosen option should be visualised	1
Dropdowns related to add/navigate in same order as those buttons	1
MODEL VISUALISATIONS	
Suggestion	**Frequency**
Include the BPMN diagram	3
EDG and task overview next to or above each other	2
If you click on an object type in the EDG, it should be automatically selected	1
Placeholders for EDG and tasks overview	1
Replace EDG by OET	1
ADDITIONS/CHANGES	
Suggestion	**Frequency**
Option to showcase which events are part of the exported complex events	3
Make the button names more self-explanatory	3
Clear out the name input after a complex event has been created	2
Info icons with information about the different elements	2
Replace 'select different object' by a refresh or cross icon => clearer name	1
Put an upload icon next to the upload buttons for extra clarity	1
Tool should never indicate a default option in the adding owned… dropdown	1
FLOW	
Suggestion	**Frequency**
Make the website flow structurally: clear sequence of steps to perform	4
Onboarding flow	2
Divide the tool into multiple sequential pages to make the flow intuitive	2
Name input and upload buttons centred on top to indicate the flow starts there	1
As much horizontality as possible: as much information per layer as possible	1
Visual links between different elements	1
Have 1 clear vertical flow: name input -> buttons & dropdowns -> summary	1

Table 5: Improvement suggestions interface A

LAYOUT	
Suggestion	**Frequency**
Provide more structure in the service list (additional dropdowns to choose complex/trivial or to select related object types to narrow down the event list)	3
Make the EDG and the mapping table the same size	2
Switch the EDG and mapping table's position	2
Make the text in the placeholders bigger and centred	2
Use colours to guide the user	2
Put the upload buttons on the top left (more important location)	1
Put the newly created complex services on top in the triggered service dropdown	1
MODEL VISUALISATIONS	
Suggestion	**Frequency**
Include the BPMN diagram	3
Do not include the EDG	1
When hovering over an object type in the EDG, the tool should show its events => Combines EDG and OET into one	1
Include all possible models but only showcase them when clicking a button	1
Replace EDG by a list with all possible events => Tool will backtrace the owning object type: not a concern of the user	1
ADDITIONS/CHANGES	
Suggestion	**Frequency**
Make add event, save complex event & generate MXP buttons more clear by changing the plain text into buttons with a coloured background	2
Only allow saving complex services consisting of 2 or more events	1
Include a button to remove selected basic events	1
Don't let input/output dropdown in the mapping table default to input	1
Replace 'upload MXP' by 'upload MERODE'	1
Option to showcase which events are part of the created complex service	1
Click on the placeholder to upload the models	1
Change the name of the 'Add event' button	1
Language detection algorithm to put most relatable service on top in the service dropdown	1
FLOW	
Suggestion	**Frequency**
Less restrictions on which features are active and which aren't	1
Onboarding flow	1
Select all object types involved and their respective events at the same time	1
Drag and drop created complex services into the mapping table => supports the flow of the creation process	1

Table 6: Improvement suggestions interface B

4.5.1 Feedback

Two frequent suggestions in layout A were related to feedback: participants expressed that they wanted some sort of visual indicator to show the model uploads were successful and to indicate the results of modelling actions. One candidate in particular even mentioned they would like the upload buttons to turn green after uploading which is exactly what got implemented within interface B. It should be noted that not a single participant who received interface B made a suggestion related to feedback which seems to validate the feedback oriented design changes.

4.5.2 Layout

This category has the largest number of suggestions overall with eleven ideas in interface A and seven in interface B. However, each layout has only one frequent suggestion. Within interface A, participants got confused by the ‘add owned event’ and ‘navigate to related object type’ buttons’ placement. To reduce this confusion, they proposed to make those buttons horizontally adjacent instead of being positioned one above the other. In addition, their respective dropdowns that get activated upon clicking the buttons should be located right underneath the corresponding button to visually indicate their relationship. We agree with the participants that this adjustment would help decrease the confusion surrounding the navigation option. A more drastic option would be leaving it out completely as in interface B. However, this removal did come at the cost of complicating the tool’s implementation.

Within interface B, participants experienced a problem that does not exist in the current layout: complex services get added to the bottom of the triggered services dropdown which means participants had to scroll all the way down this list when trying to map a complex service to its corresponding BPMN task. By contrast, within interface A, complex services are added to the top of the list which is clearly seen as superior. Participants suggested that the list should be more structured with additional dropdowns to specify the object types involved or whether the service is trivial or complex. Both options would help narrow down the contents of the list. Even though, participants in interface A did not experience this problem, we believe these suggestions to be useful, especially in situations where models may get very large which can make the list of events enormous.

4.5.3 Model visualisations

When it comes to visualising the models involved, there is a clear consensus across the two layouts: the BPMN diagram should be included. Participants expressed that having all tasks listed in the mapping table was not sufficient since they needed the diagram to distinguish between tasks with similar names.

Within interface B, three participants suggested all events should be included in some way. This could be done by providing the OET or an interactive version of the EDG that showcases all owned events upon hovering over an object type or just a plain list of all events. Participants that received interface A seemed more focussed on restructuring the visualisations already there, than on suggesting even more visualisations to be added. Only one of them proposed to include the OET. However, we do believe that including the OET would be a major improvement to the modelling flow since every single participant in both layouts went and looked at the OET printout multiple times during the modelling process. Since interface space is limited, we believe it would be optimal to include all possible visualisations (EDG, OET, BPMN) in the tool, but to let the modeller choose which are shown. This was also explicitly suggested by one participant.

4.5.4 Additions/changes

There were two frequent suggestions made regarding changes in interface A. First, participants suggested to make button names more self-explanatory. They mainly referred to the ‘add owned event’ and ‘navigate to related object type’ buttons. Yet again, this signals participant’s confusion surrounding those buttons. Second, participants proposed to include an option to showcase which events are part of a created complex service. Currently, in both layouts, once you have created a complex service, it is impossible to see which events it consists off. We agree with our participants that this doesn’t promote collaboration and will become an obstacle when modelling larger cases that have an elevated number of complex services.

4.5.5 Flow

Four participants who received interface A suggested that it should present a clear sequence of steps to perform so that they could intuitively follow the flow. Within interface B, not a single participant made such a suggestion, validating our improvements to the flow.

4.6 Post-experiment review

When the interactive part of the experiment had been completed, participants were asked two questions. First, they were asked how difficult it had been to perform the assignments they had been given on a scale from one, effortless, to five, difficult. One participant did not give an exact score so their score has been imputed with a three. Participants using interface A gave an average score of 3.03, signalling a rather neutral experience. When asked for an explanation, participants stated that there is a major learning curve and they needed some guidance. However, once they had a basic understanding of the tool, the difficulty decreased. By contrast, interface B received an average score of 1.87 which denotes the experience as rather simple. These results clearly signal a difference in difficulty level favouring the prototype.

The second question elicited participants’ general experience using the tool on a scale from one, bad, to five, good. Two participants did not give an exact score so their scores have been imputed with threes. Interface A received an average score of 2.72 signalling a mostly neutral experience. Participants explained that the tool needs some more structure and its looks could be improved upon. Interface B obtained an average score of 3.65 signalling a somewhat positive experience. Participants pointed out that knowing how to use the tool comes with practice, but apart from this learning curve, the tool is quite intuitive and easy to work with. The results are not as clear cut as the previous question, but the prototype’s UI changes do seem to have had a slightly positive impact.

4.7 Overview of Results

In this section, we briefly summarize the results and how they answer each research question.

RQ1: What is the perceived usability of the current/prototype interface?

Interface A obtained an average SUS score of 45.06, while interface B received a score of 80.28. To obtain a grade of perceived usability, these scores were compared to SUS norms leading to an F and A- respectively.

RQ2: Is there a difference in perceived usability between the current interface and the prototype interface?

There is a significant difference between the perceived usability of the two layouts. Interface A is seen as highly unusable while interface B delivers an above average user experience.

RQ3: Which usability issues occur frequently in (one of) the two layouts?

We identified nine frequent issues in interface A, listed in Table 7, compared to five frequent issues in interface B, listed in Table 8. Interface B's UI design changes, especially those related to feedback and the 'navigate to related object type' button, seem to have decreased the amount of problems.

Element(s)	Description	Frequency
upload BPMN	Doesn't see any changes upon uploading BPMN	10
upload BPMN	Expected a BPMN visualisation	4
2nd time selecting object type	Runs into bug where object type needs to be deselected and reselected	6
navigate to related	Gets confused by the 'navigate to related object type' button	7
navigate to related	Does not use the 'navigate to related object type' button at all	5
adding owned event and navigate to related	Gets confused by the complete lack of feedback upon clicking the buttons	10
add event	Clicks 'add event' without selecting an event which adds an empty event (bug)	4
add owned event vs add event	Does not intuitively understand the difference between the two buttons	5
general	Does not know which feature to use next since the flow is unclear	9

Table 7: Frequent issues interface A

Element(s)	Description	Frequency
upload MXP and BPMN	Forgets to upload models before starting the modelling process	4
upload BPMN	Expected a BPMN visualisation	8
add event	Does not know whether add event is an intermediary or a finalising button	5
add event	Expected to add multiple events at once instead of adding them one by one	5
mapping table	Takes a while to find complex services since they are on bottom of the services list	5

Table 8: Frequent issues interface B

RQ4: Which improvement suggestions are frequently proposed in (one of) the two layouts?

We identified seven frequent suggestions in interface A, listed in Table 9, compared to two frequent issues in interface B, listed in Table 10. Interface B's UI design changes solved a few usability issues leading to a smaller amount of frequent suggestions.

Suggestion	Frequency
Visual indicators to show successful upload (one suggested green colour)	4
Make the website flow structurally: clear sequence of steps to perform	4
Visual indicators to show results of modelling actions	3
Add…/Navigate… buttons adjacent + respective dropdowns close by	3
Include the BPMN diagram	3
Option to showcase which events are part of the exported complex events	3
Make the button names more self-explanatory	3

Table 9: Frequent suggestions interface A

Suggestion	Frequency
Provide more structure in the service list (additional dropdowns to choose complex/trivial or to select related object types to narrow down the event list)	3
Include the BPMN diagram	3

Table 10: Frequent suggestions interface B

RQ5: Do any of the UI design changes in the prototype get validated by the participants?

Clearly, the combination of all UI design changes had a major positive impact on the perceived usability of the tool. Specific improvements that got validated relate to the increase in feedback, removing the 'navigate to related object type' button and restructuring the interface.

All participants in interface A experienced multiple issues concerning a lack of feedback. By contrast, in interface B, not a single feedback related usability issue emerged which validates the UI design changes implementing more feedback.

Additionally, by removing the 'navigate to related object type' button from interface B, a lot of confusion could be avoided and the flow of the tool was simplified. Combined with the introduction of more structure, this resulted in the number of participants experiencing flow related issues dropping from nine to two.

4.8 Limitations

Since the experiment required prior knowledge of MERODE, participants were recruited from the pool of students who completed the course Architecture and Modelling of Management Information Systems in 2021 or 2022. Consequently, it had been at least 8 months since these students had actively studied the MERODE method. While the pre-experiment quiz scores were very high, some problems with

participants' understanding of the underlying methodology emerged during the experiments. Within both layouts, it became apparent that participants sometimes got confused about the methodology's concepts and required some clarification before continuing the modelling process. Even some participants who obtained a perfect score on the quiz, got confused about the methodology later on. Hence, it seems that the quiz was unable to fully assess the participants' understanding and that it is likely that some of the usability issues that arose stemmed from the participants' understanding of the underlying methodology and not the interface itself.

To compare the two experimental groups, we made use of SUS norms that provide a good overall interpretation of the SUS means (Lewis, 2018). However, empirical research has shown that different types of products have significantly different SUS means which could diminish the accuracy of the general norms (Lewis, 2018).

Interface B was created with the design software Figma which enabled us to create an almost seamless prototype of the application. However, it should be noted that every frame and its interactions with other frames have to be designed separately. Therefore, only the happy path was implemented since modelling the entire application would be an enormous task. As a result, participants could not make big mistakes: they could try to click on a wrong feature, but since it is not part of the happy path, nothing would happen. Participants could also deduce which interface features should probably be used next by hovering over all features to see which one is clickable. We do not believe this will have had a major impact on our findings since we always asked participants up front before interacting with the tool how they would like to perform certain actions. From their answers we can deduce that interface B has a much clearer flow that guides the user through the modelling process compared to the existing layout. However, it should still be taken into account that this may have had a small positive impact on perceived usability. The setup of the experiment could have been improved in two ways: either by implementing interface B as a real application or by also making a prototype of interface A. Both methods would enable a fair comparison.

Four participants, three in interface A and one in interface B, were unable to create both complex services due to time restrictions. When there was too little time to perform the entire experiment, we decided to leave out the creation of the second complex service to ensure that participants would still be able to finish up the rest of the experiment. Time running out had little to do with the experiment itself, but was mostly due to participants arriving late. Not being able to create a second complex service may have had a negative effect on their perceived usability since most other participants expressed that the creation of the second service was much easier seen as they were getting used to the tool.

General Conclusion

We started off this thesis by investigating the relevance of business process management and its evolution towards more data-awareness. We found that the importance of data-centric methods has been increasing but wide-spread adoption is being held back by a number of issues, such as usability.

Within conceptual modelling usability research there are two distinct research areas. Most commonly, researchers will investigate the usability of modelling languages themselves but not the used modelling tools. By contrast, the second area is focused on the usability of modelling tools' interfaces without considering how usable the underlying methodology is. Very little research has been performed in the latter area.

We decided to extend the user interface design research for conceptual modelling tools by performing a usability study with the MERODExBPMN tool and a prototype version that implements UI design principles. The following paragraphs will elaborate on the results and further areas of research.

Overall, there was a clear difference in perceived usability between interface A and B with usability grades of F and A- respectively. Clearly, interface A is seen as highly unusable by the participants with interface B offering a major improvement.

The design changes related to implementing more structure were successful: participants rated interface A as somewhat chaotic while interface B was seen as rather structured. Whereas structure and the lack thereof used to lead to a negative first impression, it got converted into a strength in interface B.

The inclusion of feedback related design changes also helped to improve the participants' experience. Within interface A, every single participant ran into problems due to a lack of feedback. They expressed a need for visual indicators that signalled the results of their actions to help guide them through the application. By contrast, within interface B, not a single participant experienced problems related to feedback. Consequently, they also did not propose any feedback-related suggestions.

Leaving out the 'navigate to related object type' button turned out to have a positive impact on the usability of the tool. The original button led to a lot of confusion and half of the participants simply chose to ignore it since it is not perceived as crucial to the tool's functioning. By removing the navigation option, this confusion was avoided. However, it should be noted that its removal comes at a cost regarding the implementation. Hence, care should be taken when redesigning the tool to support a correct implementation without compromising the usability.

The combination of the three UI design changes above, namely structure, feedback and leaving out the navigation option, simplified the flow of the tool and

helped to prevent errors. Interface B guides users through the entire modelling process, clearly indicating the sequence of steps that should be performed.

To identify points of concern that should be further improved upon, participants were asked to suggest any changes to the interface that would benefit their experience. The following paragraphs will elaborate on the three most promising finds that were not yet implemented in interface B.

In the mapping table, services get matched with their corresponding BPMN task. To do so, users have to find the correct service in a long list of all possible services in the MERODE model. Participants noted that this process would quickly become cumbersome as the size of models increased. Therefore, they suggested to provide some more structure with mechanisms that narrow down the amount of options shown in the list.

To streamline the modelling process, participants suggested that the BPMN diagram and some form of visualisation of MERODE's events should be included to avoid having to go back and forth between browser tabs. However, interface space is limited and adding two additional visualisations could potentially interfere with the structural flow of the page. Therefore, we propose to add multiple visualisations (BPMN, EDG, OET) to the tool, but give the user the choice which one gets visualised.

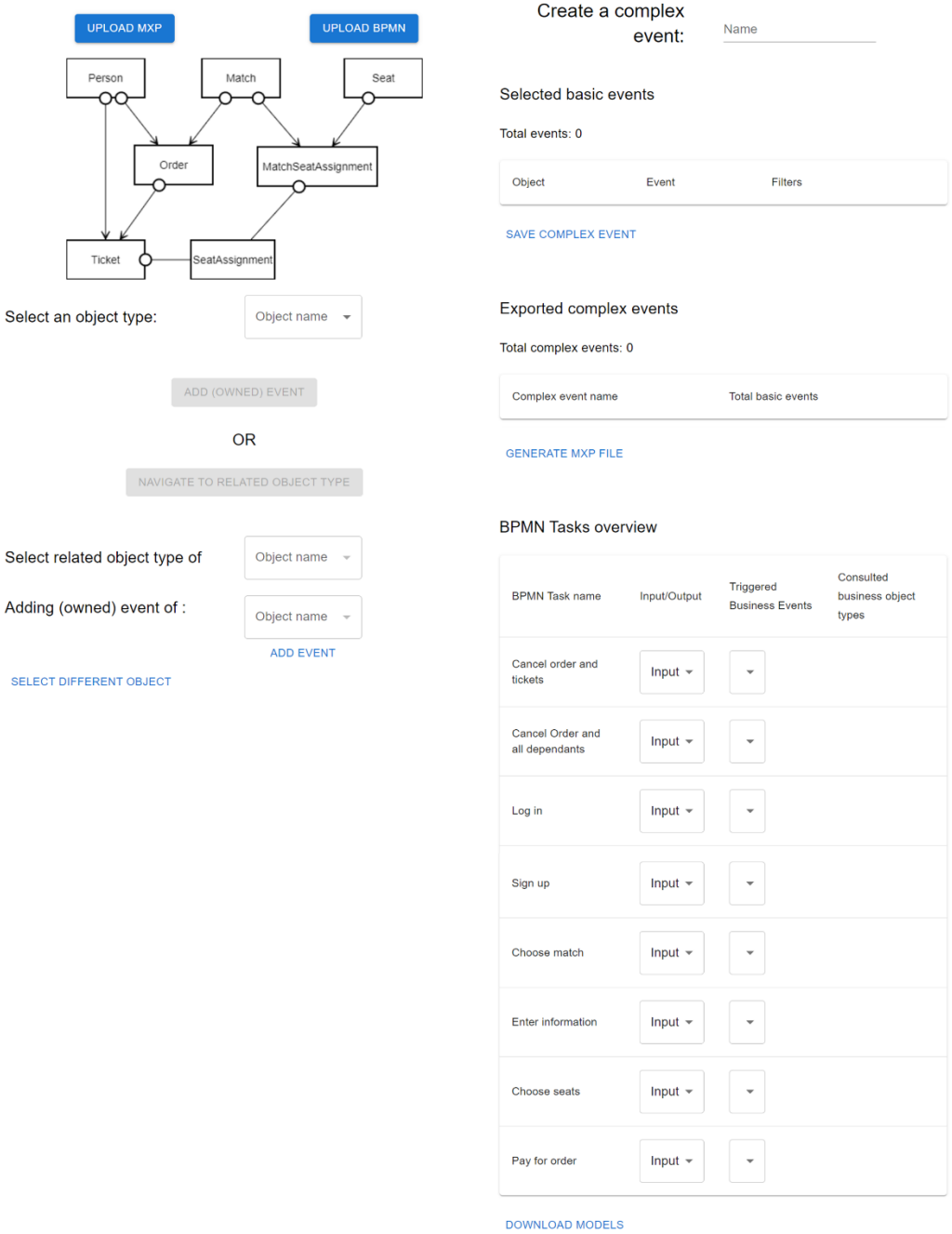
Lastly, our participants identified a major flaw in the current design: after creating a complex service, it becomes impossible to see which events it consists off. This is highly problematic since it prevents successful collaboration and will become a major obstacle in larger cases. Consequently, a feature to showcase the underlying events of a complex service should be added.

In conclusion, our research has outlined why the current interface is perceived as difficult to use and which design changes were paramount to improving the usability. Despite certain limitations, we believe this thesis has clearly validated the design changes that were implemented and provided some interesting focus points for future improvements.

The results of this study are highly relevant since one of the major shortcomings of current data-aware process modelling approaches is their lack of usability. The MERODExBPMN tool introduces usability on two levels: within the tool itself, but also within the overarching MERODE method by automating the generation of Camunda forms which currently has to be done manually. The addition of this automation will further improve the usability of the MERODE approach which will benefit future users.

Appendices

Appendix 1: Current tool interface after uploading both models



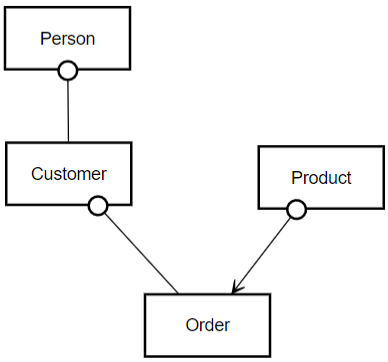
Appendix 2: Quiz

Question 1: Which definition is best for the domain model?
1) part of the enterprise model, namely, the part that describes business objects and their relationships
2) part of the enterprise model describing activities and the executing actors in an organisation
3) part of the enterprise model describing which services are offered
Answer: 1

Question 2: Which is the outermost layer in the MERODE method?
1) Information systems layer
2) Domain layer
3) Business process layer
Answer: 3

Question 3: One of the key concepts in MERODE is existence dependency. Within such a relationship, one of the objects exists first and is the master on which the other object depends.
If you live in a student dorm, which of the following exists first?
1) your rental contract
2) your student room
3) your student dorm
Answer: 3

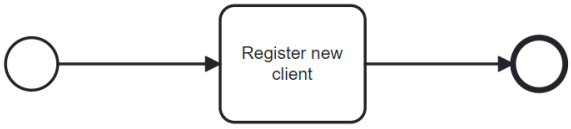
Question 4: Which of the statements about the drawing underneath is correct?



1) Person is existence dependent on customer.
2) An order can only exist if it is connected to a customer and a product.
3) A person can have multiple orders.
Answer: 2

Question 5: Which of the following statements is correct?
1) The object event table is used to map the link between BPMN and MERODE.
2) The object event table visualises which events can be executed within the different states of a business object.
3) The object event table lists all business event types and links them to the object type that owns them and other object types that are involved in them.
Answer: 3

Question 6: The business process underneath showcases the client registration process in an insurance company. Within the ‘register new client’ activity, a client file gets created and is immediately assigned to a broker that will be the personal manager of the client. What would be the best solution in such a situation?



1) The atomic events (cr_client and cr_assignment) should be grouped in a complex service.
2) The atomic events (cr_client and cr_broker) should be grouped in a complex service.
3) No grouping is needed.
Answer: 1

Question 7: How do the different layers in MERODE get linked together?
1) The business process layer can signal new information directly to the domain layer.
2) All layers are aware of each other and can call upon each other’s features.
3) The business process layer can call upon the input and output services in the information systems layer which signal business events to the domain layer.
Answer: 3

Question 8: Which of the following statements is correct?
1) Within the mapping table, BPMN tasks are directly linked to the domain model.
2) A complex service needs to be created for a BPMN task that can trigger multiple business events.
3) A BPMN task cannot be linked to a trivial input or output service.
Answer: 2

Appendix 3: Case description

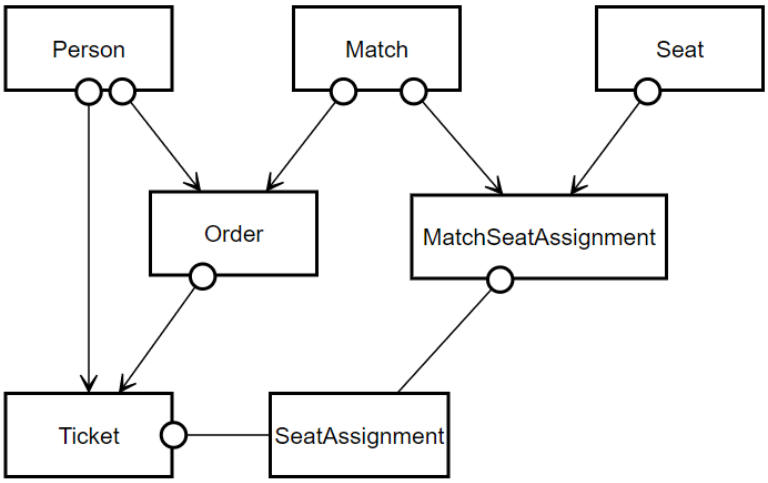
The volleyball club of Leuven is working on an application to simplify the ticket ordering process for a single match (subscriptions are out of scope for this application). They have hired some analysts who have created a BPMN model and a MERODE model based on the following description.

A person can create an order for a specific match. An order can consist of multiple tickets for several people, but all tickets are for the same match. For every ticket, a seat should be chosen. It is important to note that not all seats are always available in every match. You can, of course, only choose a seat that is available.

When a person opens the application, they can either log in or sign up if they are new users. Once the person is known to the system, they can specify the match for which they want to create an order. Once the person has chosen their desired match, they need to enter the email addresses and names of the people they want to order tickets for. If these people don't exist yet in the system, it will first save these new person profiles and, at the same time, create the tickets in the order. If the people are already known, the system will simply create the tickets. At this moment in time, the tickets are 'reserved'. They will only become 'confirmed' once a payment has been accepted later on in the process. If one of the people in the order is recognized as a banned person, they system will automatically cancel the entire order. Once all tickets have been created, the user can choose 1 seat per ticket out of the list of available seats. If the amount of tickets within the order exceeds the number of available seats left, the entire order will be cancelled. Lastly, the order needs to be paid for. This can be done online. If the payment succeeds, the process ends. If the payment defaults, the entire order will be cancelled.

Appendix 4: Case solution

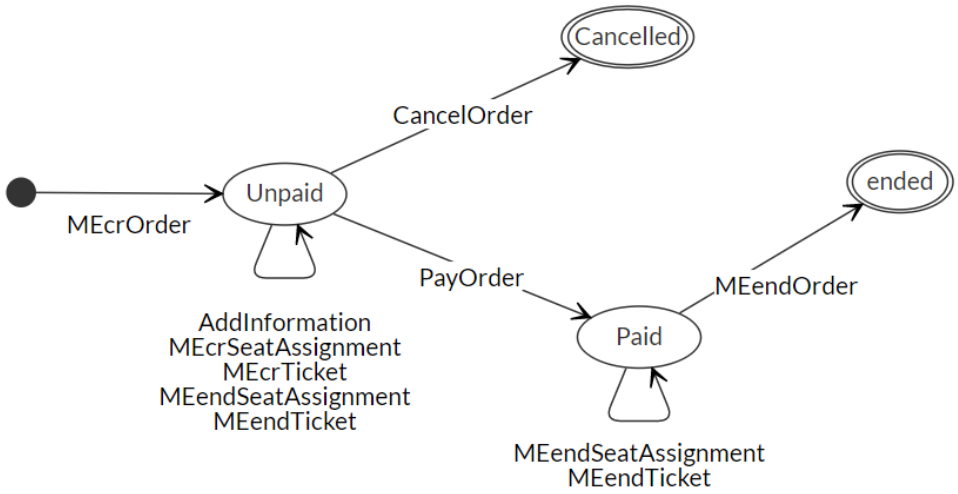
MERODE: EDG



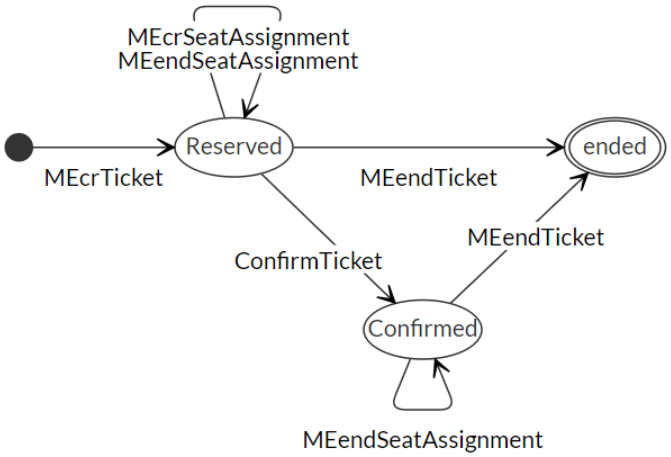
MERODE: OET

	Order	Ticket	Match	Person	Seat	SeatAssignment	MatchSeatAssignment
EVcrOrder	O/C		A/M	A/M			
EVendOrder	O/E		A/M	A/M			
EVcrTicket	A/M	O/C	A/M	A/M A/M			
EVendTicket	A/M	O/E	A/M	A/M A/M			
EVcrMatch			O/C				
EVendMatch			O/E				
EVcrPerson				O/C			
EVendPerson				O/E			
EVcrSeat					O/C		
EVendSeat					O/E		
EVcrSeatAssignment	A/M	A/M	A/M A/M	A/M A/M	A/M	O/C	A/M
EVendSeatAssignment	A/M	A/M	A/M A/M	A/M A/M	A/M	O/E	A/M
EVcrMatchSeatAssignment			A/M		A/M		O/C
EVendMatchSeatAssignment			A/M		A/M		O/E
AddInformation	O/M		A/M	A/M			
PayOrder	O/M		A/M	A/M			
ConfirmTicket	A/M	O/M	A/M	A/M A/M			
CancelOrder	O/E		A/M	A/M			

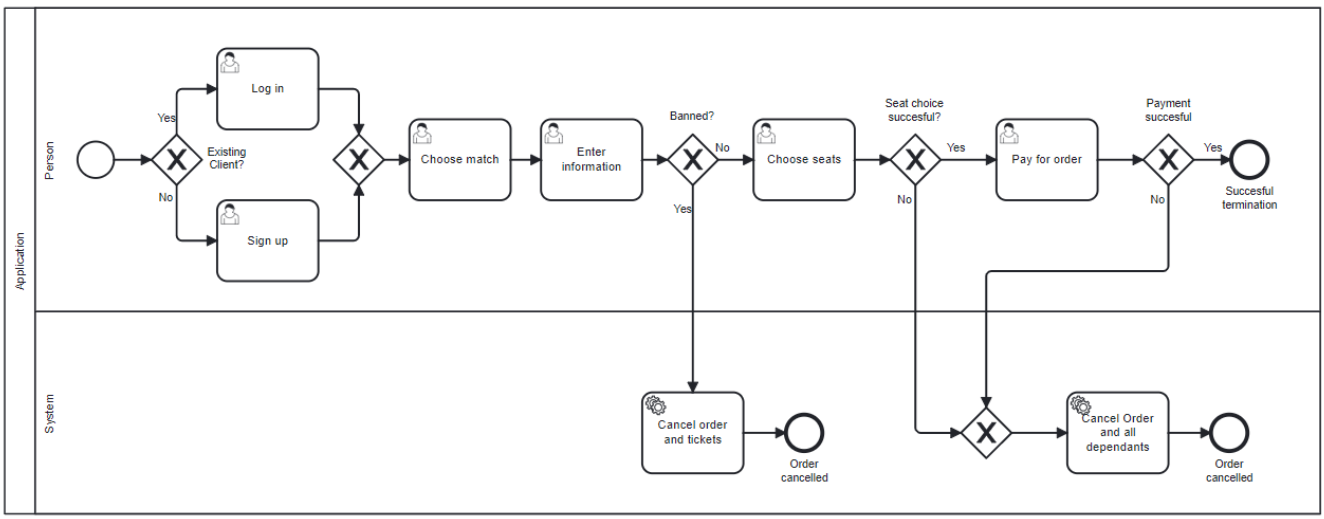
MERODE: non-default FSM Order



MERODE: non-default FSM Ticket



BPMN



Complex input services:

CrNewPersonAndTicket: EVcrPerson, EVcrTicket

PayOrderAndConfirmTickets: PayOrder, ConfirmTicket

CancelOrderAndTickets: EVendTicket, CancelOrder

CancelOrderAndDependants: EVendSeatAssignment, EVendTicket, CancelOrder

Mapping:

Task	**Services**
Sign up	EVcrPerson
Log in	-
Choose match	EVcrOrder
Enter information	AddInformation, EVcrTicket, CrNewPersonAndTicket
Choose seats	EVcrSeatAssignment
Pay for order	PayOrderAndConfirmTickets
Cancel order and tickets	CancelOrderAndTickets
Cancel order and all dependants	CancelOrderAndDependants

Appendix 5: Comprehension questions

We first show the participants the layout for 5 seconds and ask some questions afterwards. This technique is based on Doncaster (2014):
• What are your first thoughts on this layout?
• How would you rate the structure of this page from one, chaotic, to five, well-structured?
• Which elements stood out to you?

We show the tool again and ask how they want to proceed:
• What would be the first action you would like to perform?
• How would you perform this action?

If they don't immediately want to upload the models, we guide them a little:
• Do you think you need to do something else first?
• You don't have to start creating these events completely from scratch. Do you see a way to link the models you have already created with the tool?

Once the participant has identified that they should upload the models:
• Which feature is used to upload the models?

Once the models have been uploaded:
• What did you expect to happen when the models were uploaded? Is the actual change in line with your expectations?
• Do you have any additional remarks about the current layout?

After giving the task of creating a complex event, but before actual interaction:
• How would you create a complex event? You can give us an overview of how you think the general creation process will proceed, without clicking on any features yet.

After the participant interacts with the first feature:
• Did this do what you expected?
• What is your next step?

After these questions, there was no more specific template since it now depends on the participant's input. We do have some general questions we asked in specific situations:
1) Whenever the participant performs an action:
    • Did this do what you expected/wanted?
    • Is the result of your action clearly visible?
2) When a participant keeps clicking on the same feature:
    • Is there a reason you keep going back to that feature?
    • Would you like some more feedback when clicking this feature?
3) When a participant hesitates during an answer:
    • Is there something causing confusion?
    • Could you verbalize your thought process here?

After the participants have created two complex input services, they are asked to perform the mapping of these services in the tool. This is a very quick and simple task so the number of comprehension question in this part is limited.

- Which part of the interface is used for mapping?
- How do you think this feature will work?
- Did this feature function as expected?

When the interactive part of the experiment is over, there are a few more questions to gain insights in the participant's overall experience:

- How difficult was it to complete these assignments on a scale from one, effortless, to five, very difficult?
- How would you rate your overall experience using the tool on a scale from one, bad, to five, good, and why?
- If you could change some elements of the tool, what would you change?

Appendix 6: SUS questionnaire

Questions:
1. I found the tool to be simple.
2. I thought the tool was easy to use.
3. I think that I could use the tool without the support of a technical person.
4. I found the various functions in the tool were well integrated.
5. I thought there was a lot of consistency in the tool.
6. I would imagine that most people would learn to use the tool very quickly.
7. I found the tool very intuitive.
8. I felt very confident using the tool.
9. After the refresher, I could use the tool without having to learn anything new.

All SUS statements have to be scored by the participants on a scale from 1 (strongly disagree) to 5 (strongly agree).

One question was left out since it was irrelevant to our application. It asks respondents whether they would like to use the tool frequently. However, none of our participants still uses MERODE so the responses to this question were likely to distort the results.

To calculate the SUS score, the scores for each individual item, ranging from 0 (score 1) to 4 (score 5), have to be summed up. The aggregate score, then, has to be multiplied by 100/36 to obtain a score that could range from 0 to 100.

List of figures

List of tables

Sources

Bobkowska, A. E. & Reszke, K. (2005). Usability of UML Modeling Tools. In *Software Engineering : Evolution and Emerging Technologies* (Vol. 5, pp. 75–86).

Brooke, J. (1996). SUS-A quick and dirty usability scale. In Jordan P. W., Thomas B., Weerdmeester B., & McClelland I. L. (Eds.), *Usability evaluation in industry* (pp. 189–194).

Cabot, J. (2020). Positioning of the low-code movement within the field of model-driven engineering. *Proceedings - 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings*, 535–537. https://doi.org/10.1145/3417990.3420210

Camunda. (n.d.). *Camunda Modeler: Design Business Processes and Decision Models*. Https://Camunda.Com/Platform/Modeler/.

Di Ruscio, D., Kolovos, D., de Lara, J., Pierantonio, A., Tisi, M. & Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin? *Software and Systems Modeling*, *21*(2), 437–446. https://doi.org/10.1007/s10270-021-00970-2

Doncaster, P. (2014). *The UX five-second rules: guidelines for user experience design's simplest testing technique* (1st ed.). Morgan Kaufmann.

Dumas, M. (2011). On the Convergence of Data and Process Engineering. In J. Eder, M. Bielikova, & A. M. Tjoa (Eds.), *Advances in Databases and Information Systems. ADBIS 2011. Lecture Notes in Computer Science, vol 6909* (pp. 19–26). Springer.

Figma Inc. (2023). *Figma [mobile app]*.

Fortune Business Insights. (2023). *Business Process Management Market,* Retrieved April 13, 2023, from *https://www.fortunebusinessinsights.com/business-process-management-bpm-market-102639*.

Hammer, M. (2015). What is Business Process Management? In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1. International Handbooks on Information Systems.* (pp. 3–16). Springer.

Harmon, P. & Wolf, C. (2016). The State of Business Process Management. *BP Trends*.

Havey, M. (2005). *Essential business process modeling*. O'Reilly and Associates.

Houy, C., Fettke, P. & Loos, P. (2010). Empirical research in business process management - analysis of an emerging field of research. *Business Process Management Journal*, *16*(4), 619–661. https://doi.org/10.1108/14637151011065946

Hull, R. (2008). Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges. In R. Meersman & Z. Tari (Eds.), *On the Move to Meaningful Internet Systems: OTM 2008. Lecture Notes in Computer Science, vol 5332.* (pp. 1152–1163). Springer, Berlin, Heidelberg.

Hung, R. Y. Y. (2006). Business Process Management as competitive advantage: A review and empirical study. *Total Quality Management and Business Excellence*, *17*(1), 21–40. https://doi.org/10.1080/14783360500249836

ISO. (2018). *9241-11 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts.*

KU Leuven. (n.d.). *Architecture and Modelling of Management Information Systems (B-KUL-D0I71A)*. Retrieved February 15, 2023, from Https://Onderwijsaanbod.Kuleuven.Be/Syllabi/e/D0I71AE.Htm#activetab=doelstellingen_idp694288.

Künzle, V. & Reichert, M. (2011). PHILharmonicFlows: Towards a framework for object-aware process management. *Journal of Software Maintenance and Evolution*, *23*(4), 205–244. https://doi.org/10.1002/smr.524

Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, *34*(7), 577–590. https://doi.org/10.1080/10447318.2018.1455307

Lewis, J. R. & Sauro, J. (2018). Item Benchmarks for the System Usability Scale. *Journal of Usability Studies*, *13*(3). https://www.researchgate.net/publication/330225055

Liu, R., Bhattacharya, K. & Wu, F. Y. (2007). Modeling Business Contexture and Behavior Using Business Artifacts. *International Conference on Advanced Information Systems Engineering*, *4495*, 324–339.

Nielsen, J. (2005). *Ten usability heuristics.*

Nigam, A. & Caswell, N. S. (2003). Business artifacts: An approach to operational specification. *IBM Systems Journal*, *42*(3), 428.

Norman, D. A. (1983a). Design principles for human-computer interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1–10.

Norman, D. A. (1983b). Design rules based on analyses of human error. *Communications of the ACM*, *26*(4), 254–258.

Norman, D. A. (1988). *The psychology of everyday things*. Basic books.

OMG. (2014). *Business Process Model and Notation (BPMN) Version 2.0.2*.

Pietron, J., Raschke, A., Stegmaier, M., Tichy, M. & Rukzio, E. (2018). A Study Design Template for Identifying Usability Issues in Graphical Modeling Tools. *MoDELS (Workshops)*, 336–345.

Reijers, H. A., Vanderfeesten, I., Plomp, M. G. A., Van Gorp, P., Fahland, D., van der Crommert, W. L. M. & Garcia, H. D. D. (2016). Evaluating data-centric process approaches: Does the human factor factor in? *Software and Systems Modeling*, *16*(3), 649–662. https://doi.org/10.1007/s10270-015-0491-z

Rosenthal, K., Strecker, S. & Pastor, O. (2020). Modeling Difficulties in Data Modeling: Similarities and Differences Between Experienced and Non-experienced Modelers. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12400 LNCS*, 501–511. https://doi.org/10.1007/978-3-030-62522-1_37

Rosenthal, K., Ternes, B. & Strecker, S. (2020). Understanding individual processes of conceptual modeling: A multi-modal observation and data generation approach. *Modellierung 2020. Bonn: Gesellschaft Für Informatik.*, 77–92.

Ruiz, J., Serral Id, E. & Snoeck, M. (2021). Unifying functional User Interface design principles. *International Journal of Human–Computer Interaction*, *37*(1), 47–67.

Shneiderman, B. (1997). *Designing the user interface: strategies for effective human-computer interaction* (5th ed., Vol. 3). Addison-Wesley Reading, MA.

Snoeck, M. (2014). *Enterprise Information Systems Engineering*. http://www.springer.com/series/8371

Snoeck M. (2020). Merlin: An Intelligent Tool for Creating Domain Models. In F. Dalpiaz, J. Zdravkovic, & P. Loucopoulos (Eds.), *Research Challenges in Information Science. RCIS 2020. Lecture Notes in Business Information Processing, vol 385* (Vol. 385, pp. 549–555). Springer International Publishing. https://doi.org/10.1007/978-3-030-50316-1

Snoeck, M. & Dedene, G. (1998). Existence Dependency: The key to semantic integrity between structural and behavioural aspects of object types. *IEEE Transactions on Software Engineering*, *24*(4), 233–251.

Snoeck, M., Verbruggen, C., De Smedt, J. & De Weerdt, J. (2023). Software and Systems Modeling Supporting data-aware processes with MERODE. *Software and Systems Modeling*. https://doi.org/10.1007/s10270-023-01095-4

Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M. & Reichert, M. (2019). DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. *Software and Systems Modeling*, *18*(4), 2679–2716. https://doi.org/10.1007/s10270-018-0695-0

Ternes, B., Rosenthal, K. & Strecker, S. (2021). User Interface Design Research for Modeling Tools: A Literature Study. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, *16*(4). https://doi.org/10.18417/emisa.16.4

The Apache Software Foundation. (n.d.). *Apache Velocity Engine*. Https://Velocity.Apache.Org/Engine/2.3/.

Toufah, N., Jaegler, A., Kacem, T., Toufah, N. J. & Acem, T. J. K. (2020). The Business Process Management: A Successful Tool for Enhancing Moroccan Firms' social and financial Performance. *Projectics / Proyéctica / Projectique*, *26*, 95–114. www.cairn.info

Von Scheel, H., Von Rosing, M., Fonseca, M., Hove, M. & Foldager, U. (2015). Phase 1: Process concept evolution. In *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM* (Vol. 1, pp. 1–9). Elsevier Inc. https://doi.org/10.1016/B978-0-12-799959-3.00001-X

Weske, M., van der Aalst, W. M. P. & Verbeek, H. M. W. (2004). Advances in business process management. *Data and Knowledge Engineering*, *50*(1), 1–8. https://doi.org/10.1016/j.datak.2004.01.001

FACULTY OF BUSINESS AND ECONOMICS
Naamsestraat 69 bus 3500
3000 LEUVEN, België
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
feb.leuven@kuleuven.be
www.feb.kuleuven.be