Comparing Change Primitives versus Change Patterns

Support using Think Aloud

MMag. Anthi Xydis

Supervisor: Assoc.-Prof. Dr. Barbara Weber

Innsbruck, February 23, 2015
Acknowledgement

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this master thesis. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during this study.

I express my warm thanks to my supervisor Barbara for her continuous support and contribution during the very long journey of my master thesis. Thank you for you essential feedback that enriched my work. I would also like to thank you for not losing faith in me and providing me with lots of motivation.

I would also like to thank Jakob Pinggera for his fast and uncomplicated feedback on PPM and always providing me his technical support.

I am sincerely grateful to all of you participating on this exploratory study and making this work possible.
Abstract

Business process modeling has become of significant importance, adding value to the companies, since they often model their processes for documentation or analysis purposes. Companies strive for standardizing their business processes focusing on process model quality. Recent studies support that the process followed to create the process model may have an impact on the process model.

This thesis analyses the process of model creation, in particular the Process of Process Modeling (PPM). In this context, the first objective is to investigate the formalization of process models under the consideration of change primitives and change patterns. Since existing studies have focused on PPM and change primitives, the second objective aims at addressing this gap. In particular, it examines the challenges arising from change patterns utilization during model creation.

For addressing the research objectives, modelers completed two tasks by means of change primitives and change patterns. The modeler’s interactions with the modeling environment were captured with a specialized modeling environment, named Cheetah Experimental Platform (CEP). Moreover, the modeler’s thinking process was recorded, according to the Think aloud Method. The obtained information is first analyzed based on a quantitative approach, using a predefined coding schema. Grounded theory is applied as a qualitative research approach, in order to identify and categorize reoccurring challenges resulting from change patterns utilization.

Summarizing, this exploratory study proposes think aloud method as a mean for comparing change patterns and change primitives when analyzing PPM. Further, raises awareness regarding challenges resulting from change patterns, impacting the process model quality.
**Contents**

List of Figures ................................................................................................................ IV
List of Tables ................................................................................................................... V

1 Introduction ..................................................................................................................... 1
   1.1 Research Objectives ............................................................................................... 2
   1.2 Research Methods ................................................................................................. 2
   1.3 Related Work ......................................................................................................... 3
   1.4 Structure ................................................................................................................. 4

2 Background .................................................................................................................... 6
   2.1 Basic Concepts ....................................................................................................... 6
      2.1.1 Business Process Modeling ........................................................................... 6
      2.1.2 Problem Solving ......................................................................................... 8
   2.2 Process of Process Modeling ................................................................................ 10
      2.2.1 Analyzing the PPM and its Phases ............................................................... 10
      2.2.2 Recording the Process of Process Modeling ............................................... 12
   2.3 Two Ways of Modeling ........................................................................................ 13
      2.3.1 Change Primitives ....................................................................................... 14
      2.3.2 Change Patterns ......................................................................................... 14

3 Study Planning and Execution .................................................................................... 20
   3.1 Setup of Exploratory Study .................................................................................... 20
      3.1.1 Subjects ......................................................................................................... 20
      3.1.2 Factor and Factor Levels .............................................................................. 21
      3.1.3 Objects ......................................................................................................... 21
      3.1.3.1 Level of Difficulty (1) and Task Representation (2) ................................. 22
      3.1.3.2 Formalization Phase (3) and Task Information (4) ................................. 22
      3.1.4 Research Questions ...................................................................................... 23
      3.1.5 Instrumentation ............................................................................................. 24
   3.2 Design of Exploratory Study .................................................................................. 24
5.2.1 Change Pattern Support / Preflight..............................................................63
  5.2.1.1 Tool Category .......................................................................................63
  5.2.1.2 Pattern Category ..................................................................................64
  5.2.1.3 Domain Category ................................................................................64
  5.2.1.4 Critic Category ....................................................................................65
  5.2.1.5 Summary ..............................................................................................65
5.2.2 Change Pattern Support / Mortgage ..........................................................66
  5.2.2.1 Tool Category .......................................................................................66
  5.2.2.2 Pattern Category ..................................................................................67
  5.2.2.3 Domain Category ................................................................................68
  5.2.2.4 Critic Category ....................................................................................69
  5.2.2.5 Summary ..............................................................................................69

6 Conclusion and Limitations .............................................................................70
  6.1 Summary ..................................................................................................70
  6.2 Limitations ...............................................................................................71
  6.3 Future Work ..............................................................................................71

Appendix ...........................................................................................................73
List of Abbreviations ..........................................................................................79
Bibliography ......................................................................................................80
List of Figures

Figure 2.1: The cycle of the three phases of the PPM .................................................. 13
Figure 2.2: Overview of Change Patterns ................................................................. 16
Figure 2.3: Insert Process Fragment (AP1) Pattern ................................................. 17
Figure 2.4: Delete Process Fragment (AP2) Pattern ................................................ 18
Figure 2.5: Embed Process Fragment in Loop (AP8) Pattern ............................... 18
Figure 2.6: Embed Process Fragment in Conditional Branch (AP10) Pattern 19
Figure 2.7: Update Condition (AP13) Pattern ......................................................... 19
Figure 3.1: Overview of the Exploratory Study Design ....................................... 25
Figure 3.2: Cheetah Experimental Workflow Design ............................................ 26
Figure 4.1: Stages of Verbal Protocol Analysis ...................................................... 34
Figure 4.2: Objects in Data, Information and Knowledge ................................... 37
Figure 4.3: Overview of ProM Framework ............................................................. 38
Figure 4.4: Stages of the Verbal Protocol Analysis .............................................. 40
Figure 4.5: Screenshot of Code Protocol in Atlas.ti ............................................. 44
Figure 4.6: Data Analysis Procedure for VPA ..................................................... 49
Figure 4.7: Example Code Pattern for Think aloud Schema ............................... 50
Figure 4.8: ProM Log Event for Mortgage / CP - Distribution of codes ............ 50
Figure 5.1: Code Pattern of 1st Experimental Unit - Pre-Flight / BPMN ............. 54
Figure 5.2: Code Pattern of 2nd Experimental Unit - Pre-Flight / Change Pattern ... 55
Figure 5.3: Code Pattern of 3rd Experimental Unit - Mortgage / Change Pattern ... 56
Figure 5.4: Code Pattern of 4th Experimental Unit - Mortgage / BPMN ............. 57
List of Tables

Table 2.1: Characteristics of Change Primitives and Change Patterns ........................................... 15
Table 3.1: Factor and Factor Levels ..................................................................................................... 21
Table 3.2: Study Setup .......................................................................................................................... 24
Table 4.1: Overview of Alternative Techniques ..................................................................................... 32
Table 4.2 Examples of Coding with Think aloud Encoding Schema ................................................. 46
1 Introduction

Process-aware information systems (PAISs) provide additional value for enterprises, offering a more dynamic and flexible IT support for business processes, therefore it is essential to improve process models and their quality (Weber, Rinderle, & Reichert, 2006). However, the literature research shows that serious problems are frequently encountered in practice (Weber et al., 2008; Reichert & Weber, 2012). To improve process model quality, recent studies focus on the process of modeling creation, stressing that process model quality is highly dependent upon the process that was followed to create it (Pinggera et al., 2012). Following this consideration, the Process of Process Modeling (PPM) was introduced by Pinggera et al. (2010), emphasizing on the process of process modeling when creating a model in line with the process requirements using change primitives.

A key mean to improve process model quality is the correctness-by-construction principle offered by change patterns (Weber et al., 2013). Correctness-by-construction means that those change patterns given to the process designers, are the ones ensuring that a process model is transformed into another process model. Although change patterns offer a structured way of modeling, there is a lack of empirical research on using change patterns and the challenges that may arise. Most of the studies analyzed the model creation process when using change primitives (eg., Pinggera, 2014).

This exploratory research aims to address this gap. The process of creating process models is investigated when a process modeler is constructing a syntactically correct model using change primitives and change patterns. Moreover, the specialized environment developed at the University of Austria, Cheetah Experimental Platform (CEP) is used to facilitate the process of process modeling investigation in a systematic manner (cf. Pinggera et al., 2012; Pinggera et al., 2010). The scope of this study is to provide an in-depth understanding of how the usage of change patterns versus change primitives impacts upon the PPM and the challenges modelers face during model creation.

To obtain a better understanding of the PPM in this study not only the modelers' interactions with the modeling environment are recorded, but also think aloud
techniques are applied to gather the spoken-out-loud descriptions of the subjects. The collected data is then analyzed to provide insights of the PPM when utilizing change primitives vs. change patterns and the challenges designers face during the PPM using change patterns. Results showed that utilizing change patterns is more difficult than using change primitives, in particular for inexperienced modelers, impacting the PPM. The findings of this thesis, verified both quantitatively and qualitatively, provide a contribution toward understanding the PPM when using change patterns support. The main outcome of this thesis is that it recognizes the need of improvement of the modeling approach based on change patterns.

1.1 Research Objectives

The main objective of this thesis is to conduct a comparison of change primitives versus change patterns support using the think aloud method.

The first sub-objective is to explore how users deal with the two modeling tools (change primitives and change patterns) when completing the modeling tasks. More specifically, the aim is to examine the impact of the utilized modeling tool on the PPM. Thus, Research Question (RQ) 1 can be formulated as follows:

RQ1: How does the impact on the PPM vary when utilizing change patterns support compared to change primitives?

The second sub-objective is to study change patterns support from the participants’ perspective. In particular, the goal is to obtain an in-depth understanding of the challenges faced by each modeler when utilizing change patterns support for model creation. Thus, Research Question (RQ) 2 can be formulated as follows:

RQ2: Which challenges did the modelers face when utilizing change patterns for model creation?

1.2 Research Methods

In order to investigate the PPM (cf. Section 2.2) when utilizing change primitives (cf. Section 2.3.1) and change patterns (cf. Section 2.3.2), the subjects’ verbalizations were investigated while creating two process models (cf. Section 3.1.3). The core modeling
tasks were of comparable size and difficulty. The subjects were divided in two groups, being videotape-recorded as they attempted to solve the problem assigned to them (i.e., while conducting the two modeling tasks). Subjects’ interactions with the modeling tool were automatically captured using CEP. In addition, think aloud techniques were applied for data collection, i.e., verbal utterances of the participating subjects were collected during model creation (cf. Section 4.1).

To answer RQ1 quantitative data analysis techniques were used. More specifically, the data was coded using a predefined coding schema (cf. Section 4.6) and data analysis tools (e.g., a data mining technique) were used to identify code patterns (cf. Section 4.3).

To answer RQ2 grounded theory was applied for data analysis (cf. Section 4.7) to extract concepts from the transcripts with the purpose of identifying relevant challenges and grouping them into categories.

The protocols were analyzed with the help of Verbal Protocol Analysis (VPA) (cf. Section 4.5), in order to address RQ1 (cf. Section 4.6) and RQ2 (cf. Section 4.6).

### 1.3 Related Work

This exploratory study aims at investigating how the PPM is affected by the tools used for the process of model creation. Moreover, it explores the challenges faced by the modelers when using process model creation patterns. In this context, this study relates to the research on the PPM and process model creation patterns. This section presents related research focused on the interactions with the modeling environment, i.e., the PPM and on change patterns for process model creation. More specifically, existing research has investigated the PPM focusing on the formalization phase of process models by analyzing the modeler’s interactions with the modeling environment and identifying three distinct modeling styles (Pinggera et al., 2013). For this investigation, CEP was used as described in the work of Pinggera, Zugal, & Weber (2010). Similar to this thesis, a comprehensive study of the PPM is presented in the PhD thesis of Pinggera (2014), focusing on the formalization of process models in a single modeler setting, asking the participants to create a formal process model based on a complete description of a process. Additionally, modeling styles in business process modeling were analyzed
in the research of Pinggera, Soffer & Zugal et al., (2012). In contrast to theses works, this thesis focuses on the formalization of process models, asking the participants to think aloud while creating a formal process model according to the description requirements.

The modeling tools available to the modeler might influence the PPM. Several works have been published investigating the PPM by considering change primitives for process modeling (e.g., Pinggera, 2014, Pinggera et al., 2013 and Pinggera et al., 2012). The PPM when utilizing change patterns is usually not investigated. Generally, Weber et al. (2008) describes how change patterns and change support may enhance flexibility in PAISs, introducing a set of change patterns and change support features (Weber et al., 2007). In this context, only two studies have focused until now on the PPM when utilizing change patterns (Weber et al., 2013 and 2014). In contrast to this thesis, these studies focused on model interaction without utilizing think aloud techniques.

All in all, this thesis investigates the PPM by considering change primitives and change patterns, applying the think aloud research technique to complement the analysis of the modelers’ interactions with the modeling environment. Moreover, it builds upon the results obtained in Weber et al (2013), describing challenges modelers face during the PPM using change patterns.

1.4 Structure

The remainder of this thesis is structured as follows:

Chapter 2 explains the concepts of business process modeling and problem solving. Moreover, the process of process modeling and the two ways of modeling (change patterns and change primitives) are depicted.

Chapter 3 describes the overall planning and execution of the experimental procedure in depth.

Chapter 4 presents the tools used for the data analysis, introducing the think aloud method. Moreover depicts the data validation and the verbal protocol analysis
techniques. In addition, the protocol analysis addressing the research questions is presented.

In Chapter 5, the findings regarding both research questions are presented. On the one hand, quantitatively addressing RQ1 and on the other hand, qualitatively addressing RQ2. The former describes and compares the code patterns deriving from think aloud encoding schema, while the latter provides a more detailed analysis of the transcripts based on grounded theory.

Chapter 6 concludes the thesis by summarizing the main findings of the exploratory study and also providing suggestions for further research in this field.
2 Background

Chapter 2 mainly acts as an introduction to the process of process modeling (PPM) and the two ways of modeling a business process (i.e., change primitives and change patterns) in order to understand the remainder of the thesis. Therefore, in Section 2.1 the basic concepts of business process modeling and problem solving are described, creating the basis for the next chapters. Subsequently, in Section 2.2 the PPM is introduced. Section 2.3 provides background on the two ways of modeling, change primitives (cf. Section 2.3.1) and change patterns (cf. Section 2.3.2).

2.1 Basic Concepts

In this section the terminology of the most important concepts related to business process modeling is defined. Furthermore, the relation between problem solving and process model creation is explained.

2.1.1 Business Process Modeling

In order to define the concept of business process modeling the terms model, modeling, process and business process should be separately explained.

A **model** is a simplified representation (Powell & Batt, 2008) and an abstraction (Crapo *et al.*, 2000) of reality externalized in a **language** (Krogstie, 1995). Therefore, it is supposed to be simpler than, but still similar to the phenomenon it represents and also to have the same structure and way of functioning. Krogstie (1995) defines language as a set of related symbols constituting the graphemes of the language, capable of causing a contrast in meaning.

The process of building and using a model (Powell & Batt, 2008) is called **modeling**, which is described as “the process of developing an analogical system of relations, and the resulting model is comprised of entities and the relationships between them” (Crapo *et al.*, 2000; Willemain, 1995). Crapo and colleagues refer to modeling as a flexible, recursive process which is dependent on the individual practitioner, the practitioner’s skill level and the modeling task.

A **process** can be defined as the transformation of inputs into outputs. **Input** is the raw materials for a process that will produce a particular ouput, while **output** is a product
created by a system (Bocij et al., 2009). As a result, a **business process** is a closed, repeating set of interconnected activities or tasks, which can be decomposed into a number of (sub-) processes, which are performed by agents (human or machine) in a temporal or causal order required to fulfill a business purpose the aim of which is to create value (Staud, 2001; Frank & Van Laak, 2003; Hammer & Champy, 1994). More precisely, ‘activities’ or ‘tasks’ are accomplished by agents (human or computer) in organisational units using required resources (production factors) (Maier et al., 2009). ‘Interconnected series of activities’ implies that there is a specific order determining the flow of the business process. ‘Closed’ means that a splitting-off of the process is possible. Furthermore, the ‘business purpose’ of a business process is the creation of value (Maier et al., 2009).

The process of creating a business process model is called **business process modeling**, which attempts to “standardize the management of business processes that span multiple applications, multiple data repositories, multiple corporate departments, or even multiple companies” (Minoli, 2008). The resulting models capture business processes at a rather high level of abstraction and provide a basis for process analysis, simulation, execution and visualization. Consequently, a business process model includes “all the activities of a business process and their attributes as well as the control and data flow between the activities” (Reichert & Weber, 2012). Henceforth the term ‘model’ will refer to business process model.

**Business Process Modeling Notation** (BPMN) is the standardized language for graphically representing business processes (Dijkman et al., 2007). The basic sub-set of BPMN used in this study is described below. A **node** represents process steps (i.e., activities), events (start/end node) or control connectors (i.e, gateways), while a **sequence flow** connects activities and control connectors with each other. An **event** may signal the process start (start event), the process end (end event), and might also occur during the process (intermediate event). An **activity** can be atomic or complex activity. A task refers to an atomic activity and stands for work to be performed within a process, while a complex activity includes a sub-process or a reference to a sub-process schema (Dijkman et al., 2007; Weber, Reichert, & Rinderle-Ma, 2008). A **gateway** is “a routing construct used to control the divergence and convergence of sequence flow”. They are
used as either merging or splitting connectors. Finally, a process fragment denoted as a
generalized concept covering atomic or complex activities (i.e., sub-processes) and
hammocks (i.e., sub-graphes with single entry and single exit node), (Weber, Reichert, & Rinderle-Ma, 2008)

In the following, the basic control-flow patterns are explained in short, creating the
basis for Section 2.3.2 (Dijkman et al., 2007; Wohed et al., 2006).

- **Sequence**: the ability to link two activities; is represented by a sequence flow
  between two activities.

- **Parallel split**: the ability to capture a split from a single thread of control into
  multiple threads of control, which can be execute in parallel; is represented by a
  parallel fork gateway (AND-split) for creating concurrent sequence flows.

- **Synchronisation**: the ability to capture a synchronisation of multiple parallel
  subprocesses/activities into a single thread of control; is represented by a parallel
  join gateway (AND-join) for synchronizing concurrent sequence flows.

- **Exclusive choice**: the ability to represent a decision point in a business process
  where one of several branches is chosen; is represented by an XOR split
  gateway for selecting one out of a set of mutually exclusive alternative sequence
  flows.

- **Simple merge**: the ability to depict a point in the business process where two or
  more alternative branches come together without synchronisation; is represented
  by an XOR join gateway for joining a set of mutually exclusive alternative
  sequence flows into one sequence flow.

### 2.1.2 Problem Solving

Research on human cognition and problem solving has shown that modeling is closely
related to problem-solving considering it as a subset of problem-solving (Willemain, 1995). Also the creation of a process model can be considered as a complex problem
solving task. Problem-solving is also defined in the literature as “answering a question
for which one does not directly have an answer available” (Van Someren et al., 1994).
Possible reasons are, either the answer cannot be given due to problems by directly
retrieving it from memory, or the answer is not immediately recognized as the solution to the problem. Thus, problem-solving involves a combination of constructing solutions and constructing justifications of these solutions (Van Someren et al., 1994). Therefore, it is essential to understand the role of the human brain and especially of human memory on the process followed by users when modeling a complex task (cf. Section 2.2.1).

The information (e.g., verbalizations) produced by a subject during problem-solving is the content of the subject’s working memory. Consequently, the information processing within the human mind is influenced by the working memory. Working memory is generally described as “a system for simultaneous storage and processing of information” (Oberauer, 2002, p. 411). More in detail, it is a brain system providing temporary storage and manipulation of the information required for a wide range of complex cognitive activities such as language comprehension, learning, and reasoning (Baddeley A., 1992). Working memory, together with other types of memory, compose the modal model, also known as a Model of the Mind (Pinggera et al., 2012) or Model of Cognition (Crapo et al., 2000). The model consists of the Long-term Memory (LTM), being “the part where knowledge is stored more or less permanently”, the Sensory System, transforming information from the environment into an internal form and, finally, the Working Memory (WM) which is used cognitively in order to temporarily store schemas, i.e., it is the part where the currently ‘active’ information resides (Van Someren et al., 1994; Costain, 2008).

The most important models are of Baddeley and Hitch (1974) and Cowan (2008), yet they have different opinions whether working memory is a part of long-term memory or not. Nevertheless, all authors agree on the fact that working memory has a limited capacity (Cowan, 2008; Crapo et al., 2000; Baddeley, 1992) which, as Oberauer and colleagues (2000) support, acts “as a limiting factor of performance in cognitive tasks, especially complex reasoning task”.

The working memory capacity is measured in chunks represented as labels or pointer to nodes in LTM, where the real information is stored (Ericsson & Simon, 1993). Chunks were first introduced in 1956, when memory capacity was thought to be a constant in short term processing and equal to seven plus or minus two items (Crapo et al., 2000).
Cowan (2008) notes that according to later studies, the capacity and error-free limit of working memory is only three or four units. However, for the sake of this thesis the exact amount of items that can be stored in working memory is not important, only the fact that working memory limits storage of chunks to a small number affecting modelers’ problem-solving process.

The way how modelers address problem-solving tasks varies, depending strongly on their WM capacity (Pinggera et al., 2012). Usually, they tend to break down problem-solving tasks into smaller parts based on the size of their chunks, rather than solving them as a whole. As a result, the utilization of working memory plays a key role in modeler’s verbalization and modeling process (Van Someren et al., 1994; Pinggera, et al., 2012). Moreover, Pinggera and colleagues acknowledge that modelers with “a better understanding of the modeling tool, the notation, or a superior ability of extracting information from requirements can utilize their working memory more efficiently when creating process models”. On the other hand, the problem-solving task itself and the mental effort\(^1\) required while solving difficult tasks might influence the utilization of working memory and thus the development of the solution. Consequently in case of overstraining the modeler’s working memory capacity, errors might occur affecting the creation of process models and the modeler’s modeling style (Pinggera et al., 2012).

### 2.2 Process of Process Modeling

This section deals with the PPM introduced by Pinggera and colleagues (2010). Moreover, it depicts Cheetah Experimental Platform (CEP) that was used for the modeling sessions conducted as part of this thesis.

#### 2.2.1 Analyzing the PPM and its Phases

The term process of process modeling was first introduced by Pinggera and colleagues (Pinggera et al., 2010). According to the authors, the modeling process is a highly flexible and iterative process, depending on the modeler and the modeling task itself. The PPM refers to the process taking place during the *formalization phase*\(^2\), where a modeler is expected to create a syntactically correct process model reflecting the given

---

\(^1\) The amount of working memory a certain task thereby utilizes (Pinggera et al., 2012)

\(^2\) is the phase during a process modeler creates a syntactically correct model, reflecting a given domain description (cf. Pinggera, et al., 2012).
process description by interacting with the process modeling tool. The modeler’s interactions during the modeling process form a three-phase cycle (cf. Figure 2.1) consisting of the following successive phases: comprehension, modeling, and reconciliation.

**Comprehension Phase**

Comprehension of a problem is an important step in solving that problem (Costain, 2008). It is reflected in the actions (reading or thinking) of the subjects made with the intention to understand the process description and the requirements for the model. Accordingly, during the **comprehension phase** the user tries to “comprehend” the given process description with the requirements to be modeled. Moreover, the user also attempts to understand the model (or part of the model) that has been created so far. At this point the modeler has filled his or her working memory with the information extracted from the requirements and/or the hitherto created model. The slots needed for storing knowledge in working memory varies among the modelers, according to their abilities and knowledge organization (Pinggera et al., 2012). As a result, the number of the comprehension phases depends on the experience and skills of the modeler.

**Modeling Phase**

In the modeling phase the modeler is involved with the creation of the model. In order to proceed on changing the hitherto created process, the process modeler uses the information acquired and stored in working memory during the comprehension phase (Pinggera et al., 2012). Again, the utilization of working memory differs from one process modeler to another, influencing the number of modeling steps before the modeler needs to re-examine the requirements to be modeled, in order to acquire more information (Pinggera et al., 2012).

**Reconciliation Phase**

**Reconciliation phase** often occurs after the modeling phase. Reconciliation refers to the reorganization of the process model and the utilization of the process model’s **secondary notation** with the aim to improve its understandability. Thus, it embraces the lay-outing of the process model including the renaming of activities. The number of
the reconciliation phases depends on the “modeler’s ability of placing elements correctly when creating them”, while the accurate use of secondary notation is influenced by the modeler’s personal style (Pinggera et al., 2012).

2.2.2 Recording the Process of Process Modeling

In order to systematically investigate the PPM in terms of the research questions presented in Section 1.2, a modeling environment is required. Moreover, it is important that the designers’ interactions with the process model can be traced during the formalization phase. Therefore, an environment is needed supporting the detailed analysis of the PPM, with the ability to log, replay and analyze the PPM.

CEP was used in this exploratory study to observe and analyze the PPM, providing the option of replaying all intermediary versions of the process model created and allowing the step by step execution of modeling processes. CEP supports the investigation of the PPM as well as the efficient execution and monitoring of experimental procedures. Additionally, CEP comprises a set of frequently used components, such as demographic and cognitive surveys, tutorials, and basic modeling functionalities provided by Cheetah Modeler.

The modelers are guided by CEP to follow exactly the study setup. All changes to the process model together with the corresponding timestamp are stored in form of an event log (also denoted as PPM instance). Therefore, the modeling process can be replayed “at any point in time without interfering with the modeler or her problem solving efforts” (Pinggera et al., 2012).

A complete overview of modeler’s interactions categorized by the phase during which they might occur is depicted in Figure 2.1. Modelers’ interactions are either classified as modeling interactions (e.g., creating/deleting elements) or into reconciliation interactions for reorganizing the process model (e.g., layouting of edges, renaming of activities). Comprehension phases are defined as phases without interactions with the system.
Additionally, Pinggera and colleagues (2011) use modeling phase diagrams to visualize all modeling activities throughout the creation of a process model in a log. A diagram “quantitatively” demonstrates the three phases, depicting their sequence while the user creates the model step by step. More in detail, a modeling phase diagram shows how the size of the model, i.e., the number of elements, changes over time. The authors developed an algorithm in order to extract modeling phase diagrams from the log in an automated manner and measure the PPM.

### 2.3 Two Ways of Modeling

A process model can be created using change primitives (c.f. Section 2.3.1) or change patterns (c.f. Section 2.3.2). Change primitives provide a way of model creation where no assumption regarding the structure of the process model is needed. On the other hand, change patterns offer well-defined semantics and assure correctness-by-construction, but require block-structured process models.

---

3 Comprehension, Modeling and Reconciliation
This section deals with the definition of the two different ways of creating a process model in order to address the research questions. Understanding change patterns and change primitive support makes it easier to identify the impact they have on the process of process modeling as well as the challenges modelers face when utilizing change patterns.

### 2.3.1 Change Primitives

The transformation of a process model to another one requires a number of operations (Weber, Rinderle, & Reichert, 2007). In the case of change primitives, the application of multiple low-level change operations is needed operating on single elements of process model. Typical change primitives are add node, delete node, add edge, remove edge or move edge (Weidlich et al., 2010; Weber et al., 2007).

Weber et al. (2007) claim that applying a set of change primitives for altering a process model increases the complexity of change as well as the efforts for conducting the respective change. Moreover, soundness of the resulting model and data flow correctness should be checked thoroughly after the structural adaptation (Reichert & Weber, 2012).

### 2.3.2 Change Patterns

An alternative to change primitives are change patterns which enable structural modification of a process model using high-level change operations (e.g., to add an activity in parallel to another one) instead of low-level change primitives. Change patterns are high-level change operations “built upon a set of change primitives”, thus providing a higher level of abstraction (Reichert & Weber, 2012). Change patterns are independent of concrete implementations and constitute solutions to typical changes.

Table 2.1 shows the main characteristics of change patterns compared to change primitives (Reichert & Weber, 2012). It is argued that change pattern facilitate changes since less number of edit operations are needed in order to accomplish the creation of a process model. While change primitives operate on single elements of a process model, change patterns raise the level of abstraction for expressing changes by providing abstractions which are above the level of single node and edge operations. Moreover, in contrast to change primitives, change patterns guarantee correctness-by-construction.
Finally, while change primitives do not make any assumption regarding the structure of the process model, change patterns require process models to be block structured.

<table>
<thead>
<tr>
<th>Change Primitives</th>
<th>Change Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operate on single elements of process model</td>
<td>Provide high-level change operations</td>
</tr>
<tr>
<td>Correctness has to be checked after adaptation</td>
<td>Correctness-by-construction</td>
</tr>
<tr>
<td>No assumption regarding structure of process model</td>
<td>Process model needs to be block-structure</td>
</tr>
</tbody>
</table>

Table 2.1: Characteristics of Change Primitives and Change Patterns (adapted from Reichert & Weber, 2012)

Weber and colleagues (2007) proposed a set of 14 change patterns (cf. Figure 2.2) relevant for control flow changes in PAISs to complement existing workflow patterns (Weber et al., 2008). For the exploratory study conducted as part of this thesis the users were provided with seven patterns, since no more patterns were needed for creating the process model. The available patterns are described in the following. For each pattern a brief description, a description of the addressed problem, design choices describing different pattern variants and an illustration is provided.
Pattern AP1 (Insert Process Fragment) allows users to insert process fragments (cf. Figure 2.3). Three variants of AP1 exist: (1) serial insert, (2) parallel insert, and (3) conditional insert. In the case of a serial insert the new process fragment is inserted between two directly succeeding activities. As illustrated in Figure 2.3, a parallel insert enables the insertion of a new process fragment in parallel to another one, without an additional condition, while a conditional insert facilitates the association of newly added process fragment with an additional condition (Weber et al., 2007).
Change Pattern AP1: INSERT Process Fragment

**Description:** A process fragment X is added to a process model S

**Problem**
In a real world process a task has to be accomplished which has not been modeled in the process model so far.

**Solution Design Choices**

- A) The new process fragment X is inserted between two directly succeeding activities (*serial insert*)

```
S
A --------> B
\    .----> X
/    .----> S'
\     .----> A
    \  --------> B

serialInsert
```

- B) X is inserted between two activity sets (*insert between node sets*)
  
a. Without additional condition (*parallel insert*)

```
S
A --------> B --------> C
\    \    \    X
/    /    /    |
\     \     \   |
    \     \     \ |
        \     \   |
          \     \  |
            \    \ |
              \   |
                \|
                X

parallelInsert
```

b. With additional condition (*conditional insert*)

```
S
A --------> B
\    \----> X
/    /    | cond
\     \   |
    \     \   |
        \     \  |
          \    \ |
            \   |
              \|
                X

conditionalInsert
```

**Figure 2.3:** Insert Process Fragment (AP1) Pattern (adapted from Reichert & Weber, 2012)

**Pattern AP2 (Delete Process Fragment),** is used to remove process fragments (cf. Weber *et al.*, 2008). This provides a solution to a real world process where a planned task should be skipped or deleted (cf. Figure 2.4).
### Change Pattern AP2: DELETE Process Fragment

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th>A process fragment is deleted from a process model ( S ).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem</strong></td>
<td>In a real world process a planned task (process fragment) has to be skipped or deleted</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>The task (process fragment together with edges and gateways) is removed from the process model</td>
</tr>
</tbody>
</table>

![Diagram](image1.png)

Figure 2.4: Delete Process Fragment (AP2) Pattern (adapted from Reichert & Weber, 2012)

*Pattern AP 8 (Embed Process Fragment in Loop)* shown in Figure 2.5 is used to embed an existing process fragment in a loop, when a repeated execution of a process fragment is needed.

### Adaptation Pattern AP8: Embed Process Fragment in LOOP

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th>Adds a loop construct to a process model ( S ) surrounding an existing process fragment.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem</strong></td>
<td>A process fragment is actually executed at most once, but needs to be recurrently based on the same condition.</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>A loop is added to an existing process fragment.</td>
</tr>
</tbody>
</table>

![Diagram](image2.png)

Figure 2.5: Embed Process Fragment in Loop (AP8) Pattern (adapted from Reichert & Weber, 2012)

*Pattern AP 10 (Embed Process Fragment in Conditional Branch pattern)* illustrated in Figure 2.6 allows embedding an existing process fragment in a conditional branch, which is executable only under specific conditions.
Change Pattern AP10: EMDED Process Fragment in Conditional Branch

**Description**
An existing process fragment X should be executed optional, i.e., if certain conditions are met.

**Problem**
In a real world process a process fragment (task) should executed only under a particular condition.

**Solution**
The respective process fragment is embedded in a conditional branch.

![Embed Process Fragment in Conditional Branch](image)

Figure 2.6: Embed Process Fragment in Conditional Branch (AP10) Pattern (adapted from Reichert & Weber, 2012)

Using Pattern AP13 (*Update Condition change pattern*), the transition conditions in a process model can be updated. This pattern is demonstrated in Figure 2.7.

Change Pattern AP13: UPDATE Condition

**Description**
A transition condition in the process model S is updated.

**Problem**
A transition condition has to be modified as it is no longer valid.

**Solution**
A transition condition in the process model is updated.

![Update Condition](image)

Figure 2.7: Update Condition (AP13) Pattern (adapted from Reichert & Weber, 2012)

Summing up, the seven change patterns available for the exploratory study were serial insert, parallel insert, conditional insert, delete process fragment, embed in loop, embed in conditional branch and update condition. Additional to these patterns the users were allowed to rename an already modeled activity and to undo their last modeling action (cf. Section 3.2).
3 Study Planning and Execution

This chapter deals with the setup (cf. Section 3.1) as well as the design of the exploratory study (cf. Section 3.2). It provides more insight regarding the preparation and the execution of the study (cf. Section 3.3), which aims at comparing the process of creating a formal process model using change primitives as well as change patterns from an informal description.

3.1 Setup of Exploratory Study

This section describes the study setup consisting of the following single elements, the subjects (cf. Section 3.1.1), the factor and factor levels (cf. Section 3.1.2) and the objects (cf. Section 3.1.3). Moreover, depicts the collected data addressing the research questions (cf. Section 3.1.4) and the instrumentation used for the study (cf. Section 3.1.5).

3.1.1 Subjects

Consistent with the theoretical background regarding the think aloud method (cf. Section 4.1) the experimental subjects\(^4\) were carefully selected so that the impact of possible disruptive effects of thinking aloud is minimized (Van Someren et al., 1994; 34). Additionally, in order to avoid an overload of the working memory the difficulty of the modeling tasks was balanced with the knowledge of participants. Therefore, the experimental procedure was directed only at subjects, with an experience on business process modeling, especially with BPMN and CEP. The targeted subjects were moderately familiar with BPMN, in order to “avoid problems with the modeling notation, but still encounter some challenges when creating the process models of the given difficulty” (Pinggera et al., 2012). Consequently, twelve subjects were chosen that matched the above criteria: eight students, who had all participated in a graduate course on Business Processes and Workflows at the University of Innsbruck and four research assistants currently working at the Computer Science Department of the University of Innsbruck. Participation in the study was voluntary.

\(^4\) “The person who applies the methods or techniques to the experimental units” (Juristo & Moreno, 2001)
### 3.1.2 Factor and Factor Levels

A *factor* is the independent variable under examination that influences the outcome of the study (Juristo & Moreno, 2001). Each factor has several possible alternatives, called *factor levels*.

In this exploratory study **one factor** is studied, the task type, with two possible alternatives, i.e., **two factor levels**. The first factor level is Change Primitive Support (BPMN) as described in Section 2.3.1 and the second one is Change Pattern Support (CP) as depicted in Section 2.3.2. Table 3.1 summarizes the factor and factor levels of the exploratory study.

<table>
<thead>
<tr>
<th>Factor Level 1</th>
<th>Task Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Primitive Support (BPMN)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factor Level 2</th>
<th>Task Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Pattern Support (CP)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Factor and Factor Levels

### 3.1.3 Objects

The selected objects were the **Pre-Flight** and the **Mortgage modeling tasks**. The former expresses the process for conducting a general aviation flight, describing the pilot’s activities. The pilot is expected to carry out some tests in the preflight inspection phase, before starting the engine of the airplane. The latter comprises the verification process describing the activities of a bank managing a customer’s mortgage request. Again a number of tests and registrations should be performed, before accepting the customer’s request. The exact instructions of the experimental tasks can be found in Appendix I. The experimental tasks were created in accordance with the think aloud theory (cf. Section 4.1), which recognizes problem selection as an important first step in setting up a verbalization study (Hughes & Parkes, 2003). In order to guarantee the appropriateness of the tasks, the problem selection was based on the following four
criteria. If all these criteria are met, then the target process occurs in ‘optima forma’, which means that it is verbalizable\(^5\).

### 3.1.3.1 Level of Difficulty (1) and Task Representation (2)

The main criteria for the selection of the experimental tasks, used as measurements, were the *level of difficulty* and the *task representation* with respect to the involved process. The level of difficulty of these tasks should be analogous to the experience of the subjects, but the tasks should be equally difficult compared to each other. A modeling task being appropriate for the subjects means that the users should not be able to model the business process in an automated manner (Van Someren *et al.*, 1994). Moreover, it should not require high mental-effort constraining the subject’s working memory capacity. The second criterion implies that the task should be representative and relevant regarding to the business process involved and not unusual in any sense.

The appropriateness of the level of difficulty was assured by providing the subjects with similar tasks to the modeling tasks they have worked before during their studies. As mentioned in Section 3.1.1, the subjects were moderately familiar with business process modeling notation, as well as with change pattern support (analogous to their experience). Additionally, both process models had almost the same number of activities, which were 10 activities on average.

Furthermore, a practical way to check if the task is representative enough, as proposed in the literature, is to ask “the assistance of another expert in selecting several problems” (Van Someren *et al.*, 1994). Hence, the contribution of two experts was asked for the selection of process models.

### 3.1.3.2 Formalization Phase (3) and Task Information (4)

Another important criterion for the modeling task affecting the outcome of the exploratory study is to provide the ability to observe the modeler’s *formalization phase* during the experiment procedure. The ability to capture modeler’s formalization phase depends on the form of the modeling *task information*.

\(^5\)in the sense that it involves verbalizable contents on working memory (Van Someren *et al.*, 1994).
Since the protocols result via the think aloud method, their validity and completeness depends on the information received by the subjects. In case of non-verbal and complicated information, verbalization requires time and space in working memory, as it becomes a cognitive process itself (Van Someren et al., 1994). Therefore, the selected experimental tasks did not contain complicated information and required only a very simple verbalization in order to be is easy for the subjects to verbalize the contents of their working memory without overloading their working memory capacity6 (cf. Section 2.1.2).

Still, the authors of the think aloud method stress, that “also within a task area that is suitable it is not easy to select a task which gives good data” (Van Someren et al., 1994). For that reason and in order to ensure that all the above described criteria were met, a pilot was performed. The pilot is described in Section 3.3.1.

3.1.4 Research Questions

Goal of the exploratory study was to investigate two main research questions. Subsequently, it is described how the data collected in the exploratory study is used to answer the two research questions.

RQ1: How does the impact on the PPM vary when utilizing change patterns support compared to change primitives?

For addressing RQ1, the goal of this exploratory study was to collect the think aloud protocols (that capture every modeler’s sequence of thoughts) and the PPM instances (that capture all interactions with the modeling tool).

RQ2: Which challenges did the modelers face by utilizing change patterns for model creation?

To investigate RQ2, the subjects were asked to fill out a survey collecting demographic data and a cognitive load survey, evaluating the mental effort for creating each modeling task using the different tools based on a seven point Likert scale (cf. Section

---

6 Think aloud method assumes that “only the contents of working memory are verbalized instead of the entire cognitive process” (cf. Section 4.1).
Moreover, to target RQ2, the modelers’ verbalizations regarding change patterns, collected in think aloud protocols, were investigated.

### 3.1.5 Instrumentation

The equipment necessary for the exploratory study execution were a video camera and a laptop computer. The video camera was needed in order to record the verbalizations of the modeling process as well as to capture the modeling actions of the subjects. Therefore, it was targeting the display screen of the laptop, keeping the user’s anonymity. The computer was essential for the subjects to create the modeling part of the study with the use of the graphical process editor CEP (cf. Section 2.2.2). The editor provided basic functionality for creating a BPMN model (cf. Section 2.3.1) and change pattern support including the change patterns described in Section (cf. Section 2.3.2).

### 3.2 Design of Exploratory Study

The study design describes under exactly what conditions the exploratory study was conducted determining the variables affecting the study, providing a series of ways to adjust the real world before observing it (Juristo & Moreno, 2001). As already explained, the exploratory study was designed to collect data from subjects with moderate modeling skills creating two formal process models using change primitives or change patterns support from an informal description. Consequently, this exploratory study was composed of one factor with two factor levels and two objects resulting into four units. The study setup composing the study objects is illustrated in Table 3.2.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Factor Level</th>
<th>Object</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Change Primitive</td>
<td>Preflight</td>
<td>Group 1</td>
</tr>
<tr>
<td>2</td>
<td>Change Pattern</td>
<td>Preflight</td>
<td>Group 2</td>
</tr>
<tr>
<td>3</td>
<td>Change Pattern</td>
<td>Mortgage</td>
<td>Group 1</td>
</tr>
<tr>
<td>4</td>
<td>Change Primitive</td>
<td>Mortgage</td>
<td>Group 2</td>
</tr>
</tbody>
</table>

Table 3.2: Study Setup
The first unit, executed by group 1, embraces the pre-flight modeling task to measure the factor level change primitive support. The second unit, carried out by group 2, was the pre-flight modeling task measuring the task type change pattern support. The third and fourth units were conducted by group 1 and 2 respectively, including the mortgage modeling task measuring change pattern support and change primitive support.

The design of this exploratory study is denoted as **randomized balanced single factor design with one alternative per unit and repeated measurements**. ‘Randomized’ design indicates that the factor alternatives were assigned to objects in an absolutely random order. The subjects were also randomly assigned to groups. ‘Balanced’ means that each factor alternative was used by each subject (i.e., each group). Given that this exploratory study examines the influence of only one factor (the task type) the design is denoted as ‘single factor’ design with two alternatives, while ‘one alternative per unit’ was used. Finally, two modeling tasks were used as ‘repeated measurements’ in order to test the factor alternatives.

![Figure 3.1: Overview of the Exploratory Study Design](image)

An overview of the design of this study is provided in Figure 3.1. Subjects were randomly assigned to two groups of similar size. The study procedure was divided in two runs, with each group receiving two modeling tasks (repeated measurements) one for each factor alternative (balanced). Learning effects were avoided since the subjects did not deal twice with the same task.
3.2.1 Process of Exploratory Study

The detailed study process is illustrated in Figure 3.2 based on the *experimental workflow design* supported by CEP (cf. Section 2.2.2). The modeling session started with a short introduction regarding the structure of the exploratory study and the think aloud method. Immediately following, the subject received the assignment sheet including an introductory text for the think aloud method and the change patterns, instructions for the study procedure and for performing the modeling tasks, an exemplary process model, the two process model descriptions, and a group code (cf. Appendix I). The subjects were requested to read until the exemplary process model and stop before the modeling tasks’ descriptions. They were allowed to read the task descriptions only after they have accessed the CEP platform, in order to ensure accurate measurement of session times.

The study procedure begins with the subject entering the corresponding code identifying the group he or she belongs to. The upcoming survey, which the subject is asked to fill out, is mandatory and collects the subject’s demographic data. The demographic survey is independent of the group code. The collected demographical data help to understand the modeling capabilities of the subjects and support data validation (cf. Section 4.4).

![Figure 3.2: Cheetah Experimental Workflow Design](image-url)
As aforementioned, the study procedure is divided into two runs. In the first run, Group 1 got a tutorial representing the usage of change primitives (factor level 1), while Group 2 of change patterns (factor level 2). To ensure a certain minimum level of familiarity with the modeling tools (i.e., change patterns and change primitives) and to minimize the “impact of learning how to use the modeling tool”, a process modeling tutorial appears (cf. Pinggera et al., 2010). The interactive tutorial explains the functionalities of Cheetah Modeler either for change primitives or change pattern support. The tutorial comprises a screen cast that the user should follow and perform the corresponding modeling task. Thus, the subject is able to understand the used notation and get familiar with the tool (Pinggera et al., 2010). Then they are presented with the modeling tasks as described in Section 3.2. In the second run, the modeling tools are switched. In order for the subjects to apply the first factor alternative, the change primitives, a palette was available in CEP including the following elements of BPMN: ‘Activity’, ‘AND’, ‘XOR’, ‘Start Event’, ‘End Event’ and ‘Sequence Flow’. For the second factor alternative, i.e., change pattern support, the subjects were provided with the seven change patterns described in Section 2.3.2. Finally, a cognitive load survey serves to document and evaluate the mental effort for creating each modeling task using the different tools. The modeling assignments were the preflight and mortgage task (cf. Section 3.1.3).

3.3 Execution of Exploratory Study

The last phase of the exploratory study was the study execution which describes the operation of the study procedures (Basili et al., 1986), thus the preparation (cf. Section 3.3.1) and execution (cf. Section3.3.2) of this study.

3.3.1 Preparation

Study preparation implies testing the objects of the study in a pilot, with the purpose of validating that the chosen modeling tasks matched the optima-forma criteria (cf. Section 3.1.3), the understandability of the task description and the usability of CEP tool. Hence the level of difficulty, the representation of the tasks, the formalization phase and, finally, the task information were once again checked and evaluated.
The subject participating in the pilot was at similar experience level with the subjects (cf. Section 3.1.1). The user was requested to perform the modeling tasks and then answer some questions relevant to each task level of difficulty. Moreover, the subject was asked at the end to denote ambiguities and propose any necessary improvements concerning the experimental set up. The results of the pilot revealed that the problems during the experimental session were not unusual in any sense. Nevertheless, it was identified that the context of the two modeling processes should be revised and refined in order to be difficult enough, matching the users’ experience level, however not too farfetched. Additionally, further improvements regarding the change patterns were built in CEP and inconsistencies were eliminated.

3.3.2 Execution

After having assured that the target process occurs in optima forma (cf. Section 3.1.3), it is important to describe the setting of the study. The think aloud theory describes practical procedures (setting) to be applied in exploratory studies in order to get the subjects to think aloud (cf. Section 4.1). The main criterion for the setting is that it helps the subject to feel at ease (Van Someren et al., 1994). For this reason the exploratory study was conducted in a quiet room, with a bottle of water provided to each subject.

The duration of the study was between half an hour and one hour, thus it could be tiresome for the voice and throat of the subject. Moreover, the experimenter provided all the necessary information and answered any questions raised before starting with the exploratory study, to avoid as much interference as possible with the subject’s thought process during the study. The only interference occurred when the subject stopped talking or was talking too softly. Then, the experimenter prompted him or her to keep on talking aloud. Finally, all instruments were checked regularly during the study session to avoid any contingency risk.

The subjects, all students or assistants at the Innsbruck University, were contacted via email in mid July 2011 in order to arrange with each one of them the suitable date for the modeling sessions. The email informed them about the context as well as the place of the exploratory study. The overall study run was set from July to August 2011. The sub-study procedures were conducted in separated modeling sessions for each subject, at Campus Technik. Large samples were not critical for this study due to the richness of
Data collected via verbalizations (think aloud protocols). Data collection was performed anonymously, while participation in the study was voluntary. Overall, the verbal data obtained from the 12 subjects resulted to 24 protocols.

As already described, CEP was used to facilitate the conduction of the exploratory study. In particular, a basic process modeling editor was placed within CEP with the aim of recording each modeler’s sequence of actions together with the corresponding time stamp in an event log (Pinggera et al., 2012). At the same time, the overall modeling process consisting of the subjects verbalizing loud their thoughts and of their modeling activities was video recorded with their permission. As a result, the users’ interactions during the modeling process were recorded with the video camera and with CEP modeling editor. The outcome was a detailed picture of the modeling activities and the corresponding thoughts of the users, describing the creation of the process model step by step.

4 Data Analysis Procedure

This section describes the procedure that has been followed to analyze the data collected as a result of the exploratory study. Within the context of this thesis code protocols are considered as data “in the sense that coding involves procedures and coding schemes that are less objective than those for segmenting” (Van Someren et al., 1994). Accordingly, data analysis is the analysis of the code protocols (PPM instances), which examines and evaluates the influence of the factor and factor levels on the outcome of the exploratory study. Data analysis is conducted using a quantitative method (cf. Section 4.6) in order to address RQ1 (cf. Section 3.1.4) and obtain an in-depth understanding of the PPM when using change primitives and change patterns. The goal of RQ2 (cf. Section 3.1.4) is to complement findings obtained so far with a qualitative analysis of the challenges the designers faced when utilizing change patterns (cf. Section 4.7).

In the following, Section 4.1 introduces the think aloud method used for collecting verbal utterances of the participants' thinking processes in the context of this thesis. Then Section 4.2 provides the threads of using the think aloud method. Section 4.3 introduces the tools that were used for data analysis, i.e., Atlas.ti and ProM and Section
4.4 deals with data validation. The stages of the verbal protocol analysis are presented in Section 4.5, while Section 4.6 describes the encoding schema applied to the protocols for answering RQ1. Finally, grounded theory used for the qualitative protocol analysis is presented in Section 4.7 answering RQ2.

### 4.1 Think Aloud Method

Within the context of this thesis the think aloud method is used for the exploratory study, in order to discover what subjects were thinking during modeling, how they arrived at their models and to identify the difficulties they faced. The think aloud method is applied at the moment the subjects are requested to verbalize their thoughts while working on a model and after that time everything they say and do is recorded and used as data for analysis. The speech and writing result in spoken and written protocols. The recorded verbalizations are then transcribed, coded and analyzed. The method consists of (a) collecting think aloud protocols and (b) analyzing the protocols to obtain a model of the cognitive processes that take place during problem.

The think aloud method origins from psychological research and in particular from the introspection model, proposing that “one can observe events that take place in consciousness, more or less as one can observe events in the outside world” (Van Semeren, 1994). It is considered to be the standard method when instructing subjects to verbalize their thoughts (Ericsson and Simon, 1993). It is also known as *Verbal Protocol Analysis* (VPA), concurrent think aloud, which is a technique providing a “bias-free method of revealing that a persons in thinking when performing a task” (Hughes & Parkes, 2003).

The fundamental principle of the verbal protocol analysis technique is that any verbalization produced by a subject whilst problem-solving will directly reflect the contents of the subject’s working memory (Hughes & Parkes, 2003; Ericsson & Simon, 1993). Hughes and Parkes (2003) provide a very detailed literature review on the application of the VPA technique to software engineering. Further research domains that VPA has been applied to are cognitive science and psychology, artificial intelligence, human-computer interaction and computing. The appraisal of literature
revealed 26 studies (published in 15 journals and 6 conference proceedings), which have used the VPA technique.

The goal of this method is to identify changes in the knowledge during repeated problem-solving on a single task, or “differences in problem-solving abilities between people, differences in difficulty between tasks” or other factors affecting problem-solving (Van Someren et al., 1994). The resulting benefit of adopting this method is that it is a very direct method to gain insight into the knowledge and methods of human problem-solving, i.e., into a subject’s cognitive processes, and use this knowledge to address a research question. Thus, think aloud method helps avoiding ‘false’ information and getting direct data about the solution process that takes place when a subject solves a modeling task (Van Someren et al., 1994; Hughes & Parkes, 2003).

4.1.1 Alternative techniques
The think aloud is a good technique for capturing the actual thoughts of the subjects reliably, minimizing the possibility of producing “erroneous or distorted reporting” or inaccurate reports as in the case of interviews and questionnaires (Hughes & Parkes, 2003). Such retrospective techniques are considered to be inaccurate (reports) due to forgetfulness or selective report of thoughts and mental processes, providing false impressions and thus invalid data (Hughes & Parkes, 2003; Van Someren et al., 1994). Moreover, introspective techniques, where a subject is instructed to report at intermediate points during the task completion, are more “readable, than concurrent protocols but also more subject to memory errors and misinterpretation” (Van Someren et al., 1994). Other alternative techniques are dialogue observation and questions and prompting. However, both methods are inferior to the think aloud method. The disadvantage of the former technique is that not all tasks require dialogues and people incline to skip some thoughts in a dialogue situation, where a disadvantage of the latter is the constancy interruption during the problem-solving process, while “prompts that require interpretation will affect the problem-solving process” (Van Someren et al., 1994). An overview of alternative techniques is provided in Table 4.1.

---

7 Problem-solving means answering a question for which one does not directly have an answer available (Van Someren et al., 1994).
### Table 4.1: Overview of Alternative Techniques (adapted from Van Someren et al., 1994)

<table>
<thead>
<tr>
<th></th>
<th>Disturbance</th>
<th>Memory errors</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrospection</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Introspection</td>
<td>No</td>
<td>Little</td>
<td>Yes</td>
</tr>
<tr>
<td>Prompting</td>
<td>Yes</td>
<td>No</td>
<td>Little</td>
</tr>
<tr>
<td>Dialogue</td>
<td>Not applicable</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Structured techniques</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The main attributes that differentiate this method from other techniques for gathering data are:

- The think aloud method does not allow any interruptions or suggestive prompts or questions. The subjects are only being asked to verbalize his or her thought while concentrating on the task.
- The method does not interfere with the task performance and does not lead to much disturbance of the thought process.
- The collected data are very direct, as the user is not expected to provide an explanation of what he is doing.
- Experiments based on the think aloud method are straightforward for the subjects, since they are allowed to use their own language.

Summarizing, two key aspects deriving from Van Someren and collegeus (1994) are that the method avoids interpretation and explanations by the subject of his thoughts and actions, whilst it only supposes a very simple verbalization process and the verbal protocols are handled as data (free access to everyone), creating an **objective method**.

#### 4.1.2 Think Aloud Protocol

As aforementioned, the think aloud protocols are collected by asking subjects to solve one or more tasks while talking aloud\(^8\), stating directly what they think. The resulting

---

\(^8\) Thinking aloud during problem-solving means that the subject keeps on talking, speaks out loud whatever thoughts come to mind, while performing the task at hand (Van Someren et al., 1994).
protocols require substantial interpretation and analysis to realize their implications for process theories of problem-solving. The focal point of protocol analysis is to identify the processes of problem-solving or decision making. Chi (1997) supports that the process of solving a problem should match with the sequence of problem statements carried out by the designer. As a result, the validation of protocol analysis is depicted with the “degree of match between the sequence of protocol utterances and the sequence of states generated by the model” (Chi, 1997).

The VPA technique supports protocol analysis in answering the research question. The main phases of VPA are verbalizations, transcription, encoding and analysis of the codes. The person analyzing the protocols is responsible for structuring the resulted information. Within the context of this thesis, two methods of VPA were reviewed, proposed by Chi (1997) and Hughes and Parkes (2003). Chi introduces a method of coding and analyzing verbal data according to the following eight functional steps:

1. Reduce or sample the protocols. Data reduction is succeeded either by
   a. random sampling
   b. selecting a subset on the basis of some "non content" criterion
   c. coding first the entire set and then conducting a detailed coding on a selected subset

2. Segment the reduced or sampled protocols considering
   a. Granularity
   b. Correspondence
   c. the features used for segmenting
   d. searching rather than segmenting

3. Develop or choose a coding scheme

4. Operationalize evidence in the code protocols

5. Seek patterns in the mapped scheme

6. Interpret the patterns and its validity based on
a. the hypotheses being tested
b. the research questions
c. the researcher’s theoretical orientation.

7. Repeat the whole process, if needed

Another method describing the stages of verbal protocol analysis is suggested by Hughes and Parkes (2003). The authors claim that VPA is a simple technique hence it requires only few arrangements. An overview of the proposed technique is given in Figure 4.1, illustrating its main aspects. Chi’s version differs from this method with respect the number of the stage that the encoding schema is devised. The former supports that the process of devising an encoding schema should take place after a protocol is segmented and aggregate into episodes, while the latter expedites the process at stage 0.

The VPA technique used in this thesis for the analysis of the protocols is a combination of the two methods described above. The stages followed for the analysis of the collected protocols as well as the selected encoding schema is described extensively in Section 4.6.

### 4.2 Validity Threads and Risk Mitigation

Despite the benefits resulting from this method there are some factors threatening the validity (invalid data) and completeness of verbal data caused by the *disturbance of the cognitive process or interpretation by the subject* or *problems with working memory*.
The time needed for the subject to verbalize her thoughts while doing the tasks is more than the time required for doing the task silently however, there is no evidence that the additional task of verbalizing modifies the cognitive process, since the data are gathered directly with no delay (Ericsson & Simon, 1993). Van Someren et al. (1994) reported that thinking aloud does not add much to the effect heeded and evaluated that it is “inevitable in knowledge acquisition and experimental settings”. As a result there is no great impact on the thought process. Moreover, according to the authors no invalidity was found caused by (mis-) interpretation of the subjects, or evidence showing that think aloud protocols are inaccurate. Information resulting from verbalization provides accurate data, eliminating the need to trust in the object.

The thread to invalidity due to problems with working memory results from the variations in subject’s verbalization skills, as not all subjects are able to articulate their thoughts while doing a task. Some subjects have also difficulties in providing complete verbalizations, due to “substantial differences in the ease with which people verbalize their thoughts” (Van Someren et al., 1994; Hughes & Parkes, 2003). Subjects’ limited working memory capacity (cf. Section 2.1.2), which is affected by the mental effort required for solving a task, provides a plausible explanation to the differences in verbalization skills. As a result, subjects stop thinking aloud in case they are feeling difficulty or are under a high cognitive load (Ericsson & Simon, 1993). Another reason is the subject’s difficulty in concentrating on a problem-solving task (Van Someren et al., 1994).

The think aloud method was selected since it is an objective, straightforward method which treats verbal protocols as data, avoiding interpretation and explanations of the user’s thought process or interference with the task performance. In view of the fact that the literature research revealed that some people have difficulties to verbalize their thoughts, and in order to be consistent with the think aloud method, it was necessary to repetitive encourage the subjects during the study procedure to become more fluent, while concentrating on the task (Van Someren et al., 1994). The researcher gave subjects a short reminder to keep verbalizing loud their thoughts, prompting the subjects

\[\text{in the sense that people give incorrect information about the cognitive process concerned} \] (Van Someren et al., 1994)
to ‘think aloud’ or posed the question ‘what are you thinking of now?’, if there was a period of silence. Given that the think aloud method expects to get as much data as possible from as few experimental procedures as possible, the number of the 12 conducted exploratory studies was adequate.

Furthermore, as aforementioned, disturbance of the cognitive process, misinterpretation by the subject and problems with working memory are possible factors affecting the validity and completeness of verbal data. Consequently, a pilot was conducted to assert that the above criteria were met with the threats being eliminated and to assess the mental-effort needed (cf. Section 3.3.1). Moreover, the pilot validated the criteria of level of difficulty, task representation, formalization phase and task information, proving that the process occurred in ‘optima forma’ (cf. Section 3.1.3). All experimental procedures took place in a quiet room fully equipped, to avoid any interruption of the modelers’ thinking process.

4.3 Data Analysis Tools

In order to support the verbal protocol analysis the tools Atlas.ti (cf. Section 4.3.1) and ProM (cf. Section 4.3.2) were used. The former is a tool supporting segmentation and coding of the verbal protocols, converting them into analysis-ready data, i.e., code protocols. The latter is a framework supporting business process mining. ProM was used for the data analysis of the exploratory study, in order to understand the main process flows and identify differences between the two modeling tools.

4.3.1 Atlas.ti

Atlas.ti is an object-oriented tool supporting the VPA procedure in applying a “general-purpose coding scheme” and providing the means to segment and code textual data and to analyze the coded text (Hughes & Parkes, 2003). Atlas.ti is produced by Scientific Software Development, offering “similar facilities for graphical and audio data as well as textual data, and a networking feature to give a diagrammatic representation of systematic relationships between elements such as codes” (Hughes & Parkes, 2003). Its philosophy is based on four principles, Visualization, Integration, Serendipity, and Exploration, forming the acronym VISE and represents its emphasis on qualitative analysis. Atlas.ti was chosen as the most suitable tool for the ‘soft data’ analysis of the
PPM instances resulting from the exploratory study. Atlas.ti supported the knowledge transformation process, as illustrated in Figure 4.2, including the following steps:

- Symbol shows a relation to denominated object or activity (sigmatics).
- Data refers to symbols (syntactics). Data is recorded, classified and stored but unable to produce a specific meaning.
- Information is interpreted data, i.e., organized data conveying meaning or value to the recipient. Information depends on a person’s interpretation of signals (semantics).
- Knowledge shows a relation to user or application. Through inking or networking information to a person’s context or experience, knowledge is produced (pragmatics).

![Diagram of objects in data, information, and knowledge](adapted from Maier, 2009)

The think aloud protocols, constituting the primary (raw) data, were transformed with the help of Atlas into structured data generating useful information to be interpreted within the context of this exploratory study. The resulting information is described in the following chapters.

4.3.2 ProM

Process mining techniques are essential for the inspection of logs resulting from code protocols and for extracting information about processes (cf. Section 2.1). Process mining assumes that it is possible to record events such that (i) each event refers to an activity (i.e., a well-defined step in the process), (ii) each event refers to a case (i.e., a process instance), (iii) each event refers to a performer or originator (the person
executing or initiating the activity), and (iv) events have timestamps and are totally ordered (Van Dongen et al., 2005). A detailed analysis of process mining research is beyond the scope of this thesis, a more complete overview is presented in the work of Van der Aalst and Song (2004).

Process mining aims at analyzing the underlying processes extracting information from transaction logs, generated from workflow management systems Therefore, it provides techniques and tools for “discovering process, control, data, organizational, and social structures from event logs” (Van der Aalst & Song, 2004). Such a technique is Process Mining Framework (ProM), which supports business process mining to achieve “automatic construction of models explaining the behavior observed in the event log” and integrates different tools with diverse perspectives and mining techniques (Van der Aalst et al., 2007).

![Figure 4.3: Overview of ProM Framework (adapted from Van Dongen, 2005)](image)

ProM’s advantages lie in a completely plug-able environment for process mining, since it can be easily extended by adding plug-ins10 and in the interaction between plug-ins.

---

10 “A plug-in is basically the implementation of an algorithm that is of some use in the process mining area, where the implementation agrees with the framework” (Van Donge et al.)
Moreover, ProM is flexible regarding the input and output format, and allows the easy reuse of code during the implementation of new process mining ideas. An overview is provided in Figure 4.3, depicting the relationships among framework, process log format and plug-ins (Van Dongen et al., 2005). The framework supports five different types of plug-ins enabling process mining. The most important plug-ins are mining and analysis plug-in, where export, import and conversions facilitate save, open and conversion functionalities accordingly.

4.4 Data Validation

The experimental data was analyzed regarding its validity. In particular, subject's familiarity and tasks' level of difficulty were verified (cf. Section 4.4.1).

4.4.1 Validating the Experimental Subjects

The properties of subjects regarding the degree of expertise and the verbalization skills varied. Some participants had a background of IT studies, whereas others of business studies. In addition, most participants were familiar with change primitives, but not with change patterns. Nevertheless, in order to understand the modeling capabilities of the subjects, they were asked to rate themselves regarding their BPMN familiarity and mental effort, using a Likert scale with the values strongly disagree (1), disagree (2), somewhat agree (3), neutral (4), somewhat agree (5), agree (6) and strongly agree (7). The collected demographical data were then analyzed. Subjects’ familiarity with BPMN was measured by asking them how familiar consider themselves with BPMN. They reported a mean value slightly above Neutral (M=4.5). Moreover, they consider themselves somewhat confident in understanding BPMN models (M=5.3), however felt neutral competent in using the BPMN for process modeling (M=4.5). The results revealed a rather homogeneous group, with an average familiarity with BPMN.

Similarly, the subjects’ familiarity with the two modeling tasks was measured on the same Likert scale. For the preflight modeling task, subjects reported a mean value of Neutral (M=4), indicating their lack of domain knowledge. The same was strongly evidenced for the mortgage task, where the subjects’ familiarity was below Neutral (M=2.25). Both mean values indicate that “modelers could hardly rely on their prior domain knowledge for performing the modeling task” (Pinggera et al., 2012).
As already described in Section 3, it was important that the task were designed equally difficult to each other. Therefore, the mental effort required for completing each modeling task was evaluated. The subjects' mental effort was measured after each modeling task on a seven point Likert scale ranging from Very Low over Medium to Very High. For both tasks modelers reported an almost equal mean value (Preflight: M=3.5; Mortgage: M=3.6), which indicates that they perceived a Rather Low to Medium mental effort. These results show that the two process models were similar, hence allowing for comparing the two modeling tools, without affecting the outcome. Additionally, it could be confirmed that the tasks were not very difficult requiring high mental-effort, allowing the subjects to freely talk aloud without constraining their working memory capacity.

4.5 Verbal Protocol Analysis

The analysis of the think aloud protocols was executed via the verbal protocol analysis technique (cf. Section 4.1.2). The created method results from the combination of the methods introduced in Section 4.1.2. More in detail, this analysis relies mostly on the method proposed by Hughes and Parkes (2003). However it agrees with Chi (1997) who supports that the process of devising an encoding schema (stage 5) should be postponed until after the verbalization is segmented and aggregated into episodes (stage 3 and 4). The stages used for the VPA are listed in Figure 4.4.

![Figure 4.4: Stages of the Verbal Protocol Analysis](image-url)
Stage 1: Recording the Verbalization:

The verbalizations were recorded with video-tape as modelers’ actions contributed further data of interest. The recordings were very helpful for the coding phase, when the verbalizations where not understandable or when it was unclear what the subject meant or referred to. Thus, playing back the recordings provided the missing information.

According to the literature, the experimenter should decide whether to analyze all the collected protocols or only a sub-set, before starting with the transcription of the verbal data (Hughes & Parkes, 2003; Chi, 1997). The number of subjects utilized in the experiment was relative small, and therefore all 24 protocols collected were used in the next stages of the research as well as for the encoding. Experiments normally took between thirty minutes to one and a half hour to be completed; yet the experiment was not time-limited. Three levels of verbalization exist:

- Level 1: when information is reproduced directly, i.e., in the form it was acquired, without further processes necessary (Ericsson & Simon, 1980).
- Level 2: when it is needed to “describe or recode information into a verbal form” (Hughes & Parkes, 2003).
- Level 3: when the heeded information is modified. This means that the verbalization needs further processing in order to be analyzed (Ericsson & Simon, 1980; Hughes & Parkes, 2003).

The verbalizations occurred in this exploratory study were of level 1, as simple vocalization was possible without requiring further processes. The information was produced directly hence the verbalizations did not influence the level or structure of the subject’s processes (Ericsson & Simon, 1980).

Stage 2: Transcribing the Verbalization:

The transcription of the verbalization involves the creation of written documents from the verbal record. The action of transcribing was usually performed without any difficulty. In this stage the experimenter transcribed the verbalizations of each subjects in text on an excel data file. The modeling actions of the subjects were also included in
the text. The resulting excel file comprised the following four columns, the relative starting time, the relative ending time, the user name and the text. *Relative* time refers to the time since the opening of the editor, while the *timestamp* from starting to ending time refers to the duration of each verbalization phase. The phase begins (starting time) when the subject starts talking until it pauses (ending time). At the time the subject starts talking again, the next verbalization phase begins.

*Stage 3: Segmenting the Verbalization:*

The reason for segmenting according to Chi (1997) is that one has to determine what “constitutes a unit of analysis, analogous to a single trial in an experimental design”. As a result, each protocol is subdivided into a number of segments. Four issues worthy of consideration in segmenting are: (a) the grain size of the segment, (b) the correspondence of the grain size to the research questions, (c) the characteristics in the data used for segmenting, and (d) when it may not be necessary to segment. For this experiment, only the first three cases were considered for segmenting the protocols. With respect to grain size of a segment, it should be mentioned that *granularity* plays a key role for verbal data. Thus, a segment schema considered varying grain sizes, such as a complete thought, a word, a sentence a paragraph or an idea. Additionally, a segment scheme resulted from the art of designer’s interactions with the tool. Summarizing, the protocols were subdivided into segments with different grain size, according to the encoding schema.

*Stage 4: Aggregate Segments:*

Segment aggregation, as Hughes and Parkes (2003) support, is “relative straightforward, although encoder’s judgments are required if inferences are to be made from the given verbalization”. For example, the use of the short statement ‘okay’ might be only “a habitual phrase without consequence” or in contrary indicate the confirmation of quality of an object under examination. Furthermore, the encoder decides based on the context of the statements made, whether consecutive segments should be aggregated as an episode. Successive modeling segments for instance were aggregated as one episode.

*Stages 5 and 6: Devise encoding schema and encode the segments and episodes:*
**Coding** involves assigning labels to protocol episodes (and segments) according to the **coding scheme**. The result of applying the coding scheme to a protocol (raw data) is a **code protocol** (Van Someren et al., 1994). The following requirements of coding scheme allow objective coding of protocol fragments: Completeness, Justified, Grain size, Unambiguous, Context/Text Independent (Van Someren et al., 1994). 

**Completeness** means that the coding scheme should be clear and explicit and contain descriptions of all reasoning steps that might be expected to appear in the protocols.  

**Justified** coding scheme, implies that the elements and concepts introduced should result from the model and the verbalization theory. **Grain size** should match with the grain size of segments, while the aggregation of segments to episodes should be objective and match with the categories in the coding scheme. **Unambiguousness** underlines the need to preserve objectivity of the coding procedure. **Context Independent** means that the coding category should be independent of the context in which it appears. **Task Independency** should also be at hand, to avoid the task dictating the coding scheme.

The protocols should be encoded in accordance with a coding schema that captures the verbalization. A **coding schema** is a theoretically grounded model of the cognitive process and types of information involved in the activity under study (Kasper, 1998). The coding schema could be created either before or after the transcription and segmentation of the protocols. The codes should correspond to a formalism which is used to represent the knowledge, while the decisions for the appropriate codes and formalisms depend entirely on the task model, the researcher’s theoretical orientation, the hypotheses or questions being asked and the content domain (Chi, 1997; Hughes & Parkes, 2003).

Following the above considerations and in compliance with the research questions under test, the coding schema was applied with the help of Atlas.ti on the think aloud protocols. In order to address RQ1 and RQ2 two different coding schemata were devised, after the transcription and segmentation of the protocols and described in the following sections (cf. Section 4.6 and Section 4.7). The protocols were scanned to identify the vocabulary corresponding to the coding schema, examining the modeling
process and its phases. Encoding all protocols was accomplished with the coding style represented in Figure 4.5, using Atlas.ti. The protocol is illustrated at the left side of the document, while the corresponding codes are shown at the right side.

![Figure 4.5: Screenshot of Code Protocol in Atlas.ti](image)

**Stage 7: Check Reliability of Encoding:**

The consistency and reliability of the encoding should be assessed, typically by means of a second encoder (Hughes & Parkes, 2003). Additionally the coding process is subjective and requires careful attention to context (Willemain, 1995). In order to be confident in the reliability of the scheme’s application, a second encoder checked the encoding of the researcher applied to a set of protocols.

**Stage 8: Analyze the Code Patterns**

“The encode protocols are analyzed to answer the research question” (Hughes & Parkes, 2003). The number and type of analyses depends on the number and type of the research question. As already described, for answering RQ1 and RQ2, two coding schemata were created. The analysis of the coding schemata is supported by ProM and the resulting code patterns are depicted in Section 5.1 and Section 5.2 accordingly. Moreover, an example of code patterns analysis is provided in Section 4.6.3.
4.6 Coding Schema for Research Question 1: Change Patterns versus Change Primitives

This section describes the data analysis procedure applied to answer research question RQ1, which deals with the differences regarding the PPM depending on whether process designers use change primitives and change patterns.

4.6.1 Encoding Schema

The reference point for the encoding scheme related to RQ1 is the study of Pinggera and colleagues (2011), who suggest that a modeling process comprises a cycle of the three successive phases of comprehension, modeling, and reconciliation (cf. Section 2.2.1). This thesis goes a little further and extends the comprehension phase into four phases, namely comprehension understanding domain, comprehension understanding tool, comprehension validation and comprehension planning. These four phases together with modeling and reconciliation form the six codes including for the encoding of the think aloud protocols. An explanation of the codes and their identification criteria is given below, while examples of coding with the think aloud encoding schema are presented in Table 4.2:

1. Comprehension Understanding Domain (D): refers to verbalization parts showing that the subject tries to understand the modeling domain, i.e., the requirements to be modeled, either by reading out loud the process description or describing parts of the model that are already created.

2. Comprehension Understanding Tool (T): is used for encoding modeler’s verbalizations referring to the modeling tool and mostly to difficulties in understanding the tool. During this phase the users usually are trying out the available change primitives or patterns in order to find the right one.

3. Comprehension Validation (V): refers to verbalizations revealing that the modelers are checking the model, i.e., inspecting the so far created model or confirming the correctness of their modeling actions.
4. Comprehension Planning (P): applied when the subjects are planning the next steps or describing in detail their thoughts about how they are going to proceed with the next PPM part.

5. Modeling (M): captures the modeling actions of the subjects, which result from the information stored in their working memory and is acquired during the comprehension phase. Thus it captures the vocabulary corresponding to the modeler’s interactions to create or delete model elements such as activities or edges or to reverse (undo) a modeling action.

6. Reconciliation (R): the reconciliation phase comprises a sequence of modeler’s interactions in order to reorganize the process model or utilize its secondary notation. Thus the code captures subject’s verbalizations describing how she reorganizes the model, either through moving or renaming model elements. It also captures the modeler’s layouting actions.

<table>
<thead>
<tr>
<th>Comprehension Understanding Domain</th>
<th>“also das ist also eee contact ground (lesen).contact ground for taxi” “können wir jetzt in 2 Sachen interpretieren, entweder ist es check mortgage ein eigener Schritt oder man sagt es…”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comprehension Understanding Tool</td>
<td>“okay wie funktioniert das jetzt?”, “wie kann ich die beiden jetzt löschen?”</td>
</tr>
<tr>
<td>Comprehension Validation</td>
<td>“so check weather, check flight plan, check engine, check flight plan, na das war file the flight plan”, “des passt weil des aufjeden fall passiert und optional je nachdem…”</td>
</tr>
<tr>
<td>Comprehension Planning</td>
<td>“als nächstern brauchen wir ein conditional insert für dieses optionally choice”, “at first, start event”</td>
</tr>
<tr>
<td>Modeling</td>
<td>“also check weather (serial insert), check the weather”, “okay dann (activity) das nächstes activity evaluate results”</td>
</tr>
<tr>
<td>Reconciliation</td>
<td>“soll anders heißen file flight plan”, “okay schieben wir das ein bisschen dass man es lesen kann”</td>
</tr>
</tbody>
</table>

Table 4.2 Examples of Coding with Think aloud Encoding Schema
4.6.2 Example Coding

This section presents an example of devising an encoding schema (VPA stage 5) and encoding the segments and episodes (VPA stage 6). It provides a detailed analysis of the process of transforming the data collected from the experiment to significant outcomes. Particularly, the coding of protocols is enlightened with the help of a protocol generated by subject 05. The chosen protocol is taken from the second experimental unit, thus it comprises the transcript of the subject’s verbalization while modeling the mortgage process using the change pattern support. The subject was able to complete the modeling task in 10 minutes and 39 seconds. In order to express the encoding\(^{11}\) of the protocol in an understandable way, the subject’s actions with the corresponding codes are described below.

Subject 05 read aloud the short description first, wondered if the patterns are named adaptation or change patterns and continued with reading the first modeling requirement (D). Then the subject wanted to add a conditional insert (M), synchronously the subject read again the process description (D) and cancelled the modeling action the subject was about to do. Instead of a conditional insert the subject added a serial insert (M) and named the activity “check if customer has mortgage”. Then the subject confirmed the correctness of his model (V) “[00:01:42] gut”. Then the subject read again the modeling requirement (D) and decided to add now the conditional insert (M). Then the subject said that it should be better to read (D) once more the modeling requirement in order to understand if adding the conditional insert was the right thing to do: “[00:01:57]…schauen wir jetzt ob das überhaupt richtig ist…[00:02:09]” then read for 10sec and decided to continue with the modeling action and name the activity “register mortgage locally” (M). The subject inspected the outcome and commented that the modeling action fitted perfectly (V) “[00:02:22]…na past super!schön…[00:02:24]”. Then the subject updated the condition with “customer has no mortgage” (M) and moved the label to the right edge (R). Then the subject understood that this was the wrong condition (T) to update, moved the edge again (R) and finally deleted the wrong label and updated the right condition (M) while commenting that it is weird that it is not working properly (T) “[00:03:08] das ist komisch das des nicht richtig funktioniert

\(^{11}\) The exact code protocol can be found in Appendix III
The subject continued with a serial insert (M) which the activity named “inform headquarters”, shifted two activities (R) updated a condition (M) and moved some modeling elements (R). Then the subject read and interpreted the next modeling requirements (D), planned the next actions (P) and created an activity (M). Afterward the subject did again the same steps, thus read (D), planned (P) and created two activities (M). After the subject has read again the process description in order to name the activities (D) he moved the activities in order to layout his model (R). The subject read loud the next requirements (D), added another activity (M), read again, added a conditional insert, and updated a condition (M), while layouting once again the model (R). Next the subject performed the following actions, read again the process description (D), moved an edge (R), continued with reading (D), planned the next action based on the requirements (P), and created an activity based on this plan (M). Afterwards, the subject read further (D), created a serial insert (M) hoping that the activity will be placed in the right position but soon the subject realized (T) that it will not and for that reason the subject aborted the action in order to correctly add an activity (M) and read again the text to name the activity (D). The subject then started criticizing the tool (T), complaining that the subject was not feeling “free” to model. However, the subject continued reading with the intention to interpret the process description (D), created a conditional insert (M), read again in order to name it properly (D) and moved some elements (R). Thereafter, the subject was not sure if the subject had used the change patterns correctly (T), but the subject continued with modeling (M), reading (D) layouting (R) naming all the conditions (M) and again moved them to the right edges (R). Finally, the subject checked the model in order to validate it (V), the subject planned the next moves “und jetzt muss i gleich schauen ob i bei allem XORs ein dings hab, eine ehh a label habe” (P), moved some elements (R) and checked again (V). The subject looked again at the process description (D), checked the so far created model once more (V) and pressed the finish button.

4.6.3 Example of Code Patterns Analysis
This section describes how the experimental data (i.e., recorded verbalizations) are converted into code patterns (i.e., code protocols). The complete transformation process of the experimental data into data ready for analysis is illustrated in Figure 4.6. The first
step was the transcription of the verbalizations into protocols. Next step was the deployment of the encoding schema, using Atlas.ti (cf. Section 4.3.1), for coding the protocols. All code protocols were analyzed using the ProM tool (cf. Section 4.3.2). In order to import the data into ProM the protocols were divided into four different Hermeneutic units (HUs) based on the experimental unit (cf. Table 3.2) they belonged to. Thus, the four HUs, with each one of them including six protocols (n/2) were the following: (1) Pre-Flight task with BPMN tool, (2) Pre-Flight task with CP, (3) Mortgage task with BPMN and (4) Mortgage task with CP. Each HU was exported from Atlas.ti as an xml data file and transformed into an mxml log file, in order to be imported in ProM. Finally, the ProM 6.1 was applied to the data of each hermeneutic unit.

Four code patterns were created according to the four HUs of the think aloud encoding schema, providing data for change primitives and change patterns support (cf. Section 2.3).

In the following chapters the transformation process of the code patterns is described using the protocol of the Mortgage modeling task with change patterns (for short “Mortgage/CP”) generated by subject 05 as an example. Each code pattern is composed of the relative occurrences of the six phases (i.e., codes) identified, as well as of the number of transitions taking place. The relative occurrences (for short “occurrences”) result from the log events generated by ProM, depicting in terms of percentage the occurrence of each modeling phase during the experimental procedure. A transition is a pair of codes which run in sequence. At this point, it should be mentioned that in order to picture the code patterns, abbreviations for all codes were used as follows ‘D’ for Comprehension-Understanding Domain, ‘T’ for Comprehension-Understanding Tool, ‘V’ Comprehension-Validation, ‘P’ for Comprehension-Planning, ‘M’ for Modeling and ‘R’ for Reconciliation. Figure 4.7 shows the code pattern after devising the think aloud method.
aloud schema. More in detail, it depicts the occurrences resulting from ProM (cf. Figure 4.8) and the number of the transitions identified during the encoding phase. For this example the resulting subsequence appears: D-M-D-M-V-D which is translated into the following transitions: two DM transitions, one MD transition, one MV transition and one VD transition. Moreover, the analysis of this code pattern shows that modeling comprised 30.1%, while the reconciliation phase accounted for 17.4% of the exploratory study. Finally, the comprehension phases involved 28.5% Understanding the Domain, 9.5% Understanding the Tool, 7.9% Validation and 6.3% Planning.

![Figure 4.7: Example Code Pattern for Think aloud Schema](image)

<table>
<thead>
<tr>
<th>Model element</th>
<th>Event type</th>
<th>Occurrences (absolute)</th>
<th>Occurrences (relative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling</td>
<td>running</td>
<td>19</td>
<td>30.159%</td>
</tr>
<tr>
<td>Comprehension (understanding Domain)</td>
<td>running</td>
<td>18</td>
<td>28.571%</td>
</tr>
<tr>
<td>Reconciliation</td>
<td>running</td>
<td>11</td>
<td>17.46%</td>
</tr>
<tr>
<td>Comprehension (understanding Tool)</td>
<td>running</td>
<td>6</td>
<td>9.524%</td>
</tr>
<tr>
<td>Comprehension (Validation)</td>
<td>running</td>
<td>5</td>
<td>7.937%</td>
</tr>
<tr>
<td>Comprehension (Planning)</td>
<td>running</td>
<td>4</td>
<td>6.349%</td>
</tr>
</tbody>
</table>

![Figure 4.8: ProM Log Event for Mortgage / CP - Distribution of codes](image)
4.7 Analysis Procedure for Research Question 2: Challenges in Change Patterns Usage

This section describes the data analysis procedure applied to obtain an in-depth understanding of challenges during the PPM based on changes patterns, answering research question RQ2. A qualitative analysis was performed in an attempt to understand change patterns support from the participants’ perspective. In particular, Grounded Theory was used, which is “a specific methodology developed for the purpose of building theory from data” associated with qualitative research”, first introduced by Barney Glaser and Anselm Strauss in 1967 (Corbin & Strauss, 2008). Grounded theory facilitates the discovery of theory from data, since its founders’ support that the researcher should focus on developing a new theory grounded in empirical data (Dunne, 2011). According to the theory, data analysis implies coding raw data and transforming them into conceptual data, while the researcher interacts with data using thinking strategies called analytic tools (Corbin & Strauss, 2008).

The protocol analysis was based on constant comparisons, focusing on transcripts of the second (Mortgage/CP) and third (Preflight/CP) experimental unit. Thus, two HUs were created with the help of Atlas.ti, each HU including the corresponding group of transcripts. Along with Corbin & Strauss (2008), constant comparisons is one type of comparative analysis where the researcher compares each incident in the data with other incidents to mark similarities and differences as the researcher proceeds with the investigation.

In compliance with grounded theory, the transcripts (raw data) were reviewed several times to mark aspects (i.e., incidents that evident difficulties when using change patterns support) and to identify properties in order to allow for classification. Accordingly, the transcripts were examined once again comparing incident with incident, in order to identify similarities among users’ verbalizations when experiencing problems with the tool. As a result, conceptual similar incidents were grouped together under a higher-level descriptive concept and given the same code.

12 Deriving and developing concepts from data (Corbin & Strauss, 2008).
The authors denote the importance of constant comparisons since it “allows the researcher to differentiate one category/theme from another and to identify properties and dimensions specific to that category/theme” (Corbin & Strauss, 2008). In contrast to quantitative analysis used for VPA, in qualitative analysis the codes are not pre-specified but emerge from raw material, i.e., transcripts. The findings of the grounded analysis as well as the four categories identified are described in Section 5.2 in detail.
5 Findings

This section discusses the findings resulting from the data analysis. An exploratory study is conducted using a quantitative method of user research in order to achieve an extensive investigation of RQ1, in comparing change primitives versus change patterns support. In order to target RQ2, a qualitative method is used analyzing the challenges arising from change patterns. Therefore, Section 5.1 addresses RQ1, presenting the quantitative analysis of the code patterns. Section 5.2 describes the qualitative findings regarding RQ2.

5.1 Change Patterns vs. Change Primitives

This section addresses research question RQ1. In a quantitative research the researcher knows in advance what he/she is looking for, with the purpose to classify the outcomes and construct models in an attempt to describe what is observed. In this quantitative analysis the researcher focused on the aggregated experimental outcomes, with the goal to classify the protocols according to modeling phases in terms of code patterns. Consequently, each code pattern comprises data produced from the 12 process models with the use of ProM. The aggregated data of the 12 process models of each experimental unit form a modeling process.

5.1.1 Think Aloud Code Patterns

In this section the outcome of the think aloud encoding schema is discussed. The aggregated data for each experimental unit (EU) is presented in the following figures (cf. Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4) showing the relative occurrence of the different PPM phases (i.e., modeling (M), comprehension understanding domain (D), comprehension validation (V), comprehension planning (P), comprehension understanding tool (T) and reconciliation (R) ) for each PPM instance as well as the total number of transitions\textsuperscript{13} between phases. Transitions occurring less than five times are not depicted.

\textsuperscript{13} Transition is a pair of codes which run in a sequence
5.1.1.1 Preflight Modeling Task

The results for the EU1 (Pre-Flight/BPMN) depict that the investigated instances consist of 36.9% M, 30.5% D, 17.9% R, 7.7% P, 4% T, and 2.6% V. The analysis of the total number of transitions provides a more in depth view. The most frequently used transitions were MD (MD=75) and (DM=71), followed by MR and RM transitions (MR=RM=36). These findings imply that the users frequently switched between the process description to understand the requirements and the model externalizing the textual description. They had also often reorganized (i.e., reconcile) their process model during modeling, indicated by the high number of MR and RM phases. Figure 5.1 demonstrates that reconciliation occurs more often after modeling than (it does) after comprehension understanding domain (phase).

In addition the data shows that subjects spent some time on less frequently occurring comprehension activities. In particular, it is illustrated that subjects spent 7.7% on comprehension planning, while little time was spent on tool comprehension (4% T) and validation (2.6%), indicating confidence in handling change primitives as well as the modeling environment. Summarizing, it is obvious that in the first EU the dominant phases are modeling, comprehension-understanding domain and reconciliation.

Figure 5.1: Code Pattern of 1st Experimental Unit - Pre-Flight / BPMN

Figure 5.2 represents the code patterns of the EU2 (Pre-Flight/Change Pattern), where the modelers received the preflight task using change patterns support. The overall
modeling process consists of 29% modeling, 53.1% comprehension and 17.6% reconciliation. The most frequently comprehension activity was comprehension understanding domain (29%), followed by comprehension tool understanding and comprehension planning with 10.7% each and validation (3.8%). It should be pointed out, that the percentage of R is similar to the percentage of R in the EU1 code pattern (R=17.9%). In turn, more focus is put on comprehension phases (44.8% EU1 vs. 53.1% EU2) instead of modeling phases (36.9% vs. 29%).

The rather high percentage of T indicates that the users spent considerable time on tool understanding, suggesting that it was challenging for them to work with the change patterns. This is also supported by the comparably high percentage of P (10.7%), implying that modelers had to carefully plan their actions before continuing with modeling (PM=21) and/or reconciliation (PR=6). The users not encountered some issues in using the tool correctly (TM=23), but also in mapping the change patterns to the process requirements to be modeled (Domain). This is illustrated by the high number of TD (TD=10) and DT (DT=12) transitions. It is observed, that also for this code pattern the phases most frequently occurring phases are modeling, comprehension understanding domain, and reconciliation. Moreover, the most often occurring transitions were MD (49) and DM (42).

Figure 5.2: Code Pattern of 2nd Experimental Unit - Pre-Flight / Change Pattern
5.1.1.2 Mortgage Modeling Task

The results of the EU3 (Mortgage/Change Pattern), illustrated in Figure 5.3, suggest that modeling (M=30.2%) takes the highest proportion of the modeling process followed by comprehension domain (25.1%) and reconciliation (24.7%). However, of great interest is the fact that this pattern reports the highest reconciliation percentage of all patterns (R=24.7%), occurring almost as frequently as comprehension understanding domain (D=25.1%). The relatively low percentage of comprehension understanding domain (when compared to EU2) might be an indication that the mortgage domain as perceived as easier to understand than the preflight domain. The remaining comprehension activities show similar distributions when compared to EU2 (i.e., comprehension tool 9.6%, comprehension planning 7.4%, comprehension validation 2.7%). Like in EU2 data suggests that users had difficulties in tool usages, indicated by the relative high amount of the Tool Understanding (T=9.6%) as well as in the number of TM transitions (TM=29).

Figure 5.3: Code Pattern of 3rd Experimental Unit - Mortgage / Change Pattern

Summarizing, the dominant phases pointed out are modeling, comprehension understanding domain, and reconciliation, coinciding with the findings of the study of Pinggera et al. (2010) and in line with the results for EU1 and EU2. Nevertheless, modelers encountered difficulties in working with change patterns, as depicted by the high number of understanding tool occurrences.
The outcome of the EU4 (Mortgage/BPMN), illustrated in Figure 5.4, reflects that the subjects deal mainly with the modeling phase (40.4%) and the understanding of domain (30.9), paying less attention on improving the model’s optical appearance (R=15%). On the other hand, the number of transitions as well as the percentage of the Planning phase (P=10.6%) imply that the users tried to plan their actions before and after modeling (PM=18; MP=13). The modelers seem not to be highly involved with validating their model (V=1.4%). Moreover no major issues with tool understanding were observed (T=1.4%). Summarizing, modeling and understanding domain form the largest part of the modeling process for the EU4.

![Figure 5.4: Code Pattern of 4th Experimental Unit - Mortgage / BPMN](image)

**5.1.1.3 Comparison**

The representation of the outcomes of each experimental unit as code pattern provided an aggregated overview of the different PPM instances, based on the relative occurrence of each PPM phase and the corresponding transitions. Nevertheless, since the scope of this study is to compare change pattern support and change primitives support, this section deals with the assessment of these two factor alternatives. For this, the results are compared to each other in two ways. First, a comparison is made between the code patterns depicting the two different modeling tools used for the same tasks (i.e., EU 1 vs. EU 2 and EU 3 vs. EU 4). It follows the comparison of the code patterns capturing the same modeling tools (i.e., EU 1 vs. EU 3 and EU 2 vs. EU 4).
5.1.1.3.1 Change Primitives vs. Change Patterns

a. Modeling Task: Preflight

The first comparison is made between the two modeling tools on basis of the preflight modeling task (i.e., EU1 and EU2 are compared). The data illustrates that modelers using change primitives (EU1) could focus more on modeling when compared to change patterns ($M_1=36.9\%$ vs. $M_2=29\%$). Change patterns users, in turn, had a stronger focus on comprehension (overall $U_1=44.8\%$ vs. overall $U_2=53.1\%$).

When having a more detailed look at comprehension it could be seen that subjects using change primitives could predominantly focus on domain understanding followed by comprehension planning with little focus on tool understanding and validation. Change patterns usage, in turn, led to a smaller relative occurrence of domain understanding and emphasized tool understanding and validation more when compared to EU1. In particular, the high number of transitions TM ($TM_1=9$, $TM_2=23$), MT ($MT_1=10$, $MT_2=16$), TD ($TD_1<5$, $TD_2=10$) and DT ($DT_1<5$, $DT_2=12$) indicates that subjects had troubles using change patterns. TM and MT transitions point towards problems externalizing the internally created mental model (e.g., how to use a particular pattern), while TD and DT transitions point towards problems in mapping the textual description to the available patterns (i.e., pattern selection problems). Finally, EU2 suggests a higher relative occurrence of validation phases when compared to EU1 ($V_1=2.6\%$ vs. $V_2=3.8\%$).

b. Modeling Task: Mortgage

In the second comparison the two modeling tools are being compared on basis of the mortgage modeling task (i.e., EU3 and EU4 are compared). Once again, it is observed that modelers using change primitives (EU4) could focus more on modeling when compared to change patterns ($M_3=30.2\%$ vs. $M_4=40.4\%$). Change primitives, in turn, depict a high amount of reconciliation phase implying that reconciliation, in this case, is dependent on the tool support provided ($R_3=24.7\%$ vs. $R_4=15\%$). Both change patterns and change primitives user showed a similar focus on the overall model comprehension (overall $U_3=44.8\%$ vs. overall $U_4=44.3\%$).
Change patterns usage, led to a higher relative occurrence of tool understanding and validation when compared to EU4 (T₃=9.6% vs. T₄=1.4%, V₃=2.7% vs. V₄=1.4%). More in detail, the analysis of the comprehension phases, such as the high number of transitions TM (TM₃=29, TM₄<5), PT (PT₃=8, PT₄<5) and DT (DT₃=12, DT₄<5), implies that the subjects had troubles using change patterns. In particular, the TM and PT transitions point towards problems externalizing the internally planned or created mental model (e.g., how to use a particular pattern), while TD and DT transitions point towards uncertainty in mapping the textual description to the available patterns (i.e., pattern selection problems). Moreover, the high number of transitions MR (MR₃=66, MR₄=35), RM (RM₃=49, RM₄=23) and DR (DR₃=40, DR₄=11) suggests that modelers using change pattern support were mainly concerned with reorganizing their model which points towards troubles in mapping the textual description to the process model (i.e., how to (re-)organize the created process model). It could be possible that the modeling task in combination with the layouting feature integrated in the CP tool may had an impact on the higher number of reconciliation transitions (i.e., MR, RM and DR), depicted in EU3.

c. Conclusions

In this section the two modeling tools were assessed to attain information on how modelers perceived change patterns vs. chance primitives. For both modeling tasks, the modelers using change primitives were mainly involved with modeling, which suggests that applying change primitives was more straightforward for the modelers than change patterns. Change patterns users, in turn, had a stronger focus on overall comprehension, suggesting that comprehension is dependent on the tool support. Summarizing, the results depict that subjects had troubles using change patterns, in mapping the textual description to the available patterns, or to the process model and in externalizing the internally planned or created mental model.
5.1.1.3.2 Pre-Flight vs. Mortgage

a. Modeling Tool: Change Patterns

Until now the two different modeling tools are compared to each other based on the same modeling tasks. However, it is important to discuss the findings resulting from the comparison between the same modeling tools used in the different modeling tasks (i.e., mortgage and preflight). The first comparison is between the two modeling tasks when using change patterns (i.e., EU2 and EU3 are compared). Change patterns users creating the preflight process model (EU2) had a strong focus on comprehension (overall $U_2=53.1\%$ vs. overall $U_3=44.8\%$), especially on comprehension planning ($P_2=10.7\%$ vs. $P_3=7.4\%$). When creating the mortgage process model users could focus more on reconciliation when compared to preflight ($R_2=17.6\%$ vs. $R_3=24.7\%$). Both code patterns depict a similar amount of modeling phases which suggests that modeling is independent on the modeling task applied ($M_2=29\%$ vs. $M_3=30.2\%$).

In particular, the high number of transitions MD ($MD_2=49$, $MD_3=61$) and DM ($DM_2=42$, $DM_3=53$) suggest that the modelers probably experienced more difficulties in understanding the mortgage process description and the requirements to be modeled when compared to EU2. Additionally, the high number of transition RD ($RD_2=29$, $RD_3=41$), DR ($MD_2=22$, $MD_3=40$), MR ($MR_2=27$, $MR_3=66$) and RM ($RM_2=12$, $RM_3=49$) point towards troubles in mapping the textual description to the process model (i.e., how to (re-)organize the created process model) when compared to EU2. Finally, the data depicts that the subjects modeling the preflight process model could focus more on comprehension planning, and validation than the modelers of EU3.

b. Modeling Tool: Change Primitives

In the second comparison the two modeling tasks are being compared when utilizing change primitives (i.e., EU1 and EU4 are compared). The modelers creating the preflight process model focused slightly more on modeling when compared to EU4 ($M_1=36.9\%$ vs. $M_4=40.4\%$). Both code patterns depict a similar amount of comprehension and reconciliation phases which suggests that both modeling tasks were equally difficult (overall $U_1=44.8\%$ vs. overall $U_4=44.3\%$ and $R_1=17.9\%$ vs. $R_4=15\%$).
The number of transitions MP and PM (MP₄=13, PM₄=18), indicates that the user planned their actions before creating the mortgage model which suggests modelers’ insecurity in modeling the mortgage process model, when compared to EU1. On the other hand the high number of transitions MD, DM, RD and DR of EU4, indicates that the domain was more difficult to understand.

c. Conclusions

Summarizing the findings of the comparison of the two modeling tasks, it is depicted that the modelers experienced some issues in understanding the mortgage process description and the requirements to be modeled, or in mapping the textual description to the process model, which suggests that both tasks were at a similar difficulty level, since no great differences were identified.

5.2 Change Patterns Challenges

This section presents the obtained results of the data analysis and describes the problems faced by each modeler when dealing with change patterns. To address RQ2 only EU2 and EU3 were used, i.e., the experimental units belonging to factor level change pattern support. As basis for this comparative analysis the transcripts of preflight and mortgage process models were used (i.e., six transcripts for each experimental unit). For data analysis, grounded theory (cf. Section 4.7) is used to get the “inner experience of participants” with the aim of identifying low-level concepts and grouping them into higher-level concepts, i.e., categories (Corbin & Strauss, 2008). The analytic tool applied on the transcripts was constant comparisons, introduced in Section 4.7. The “first reading” of the think aloud protocols revealed that the subjects tend to behave in many cases in the same manner (Corbin & Strauss, 2008, p. 163). They often used the same expressions while modeling, raised the same questions, or made the same modeling mistakes during the experiment. Subsequent, these incidents were compared to each other for similarities and differences. Similarities were identified and classified for both, EU2 (preflight) and EU3 (mortgage). The coding was repeated until no new concepts could be derived. The following four categories were identified, Tool, Pattern, Domain, and Critic:
Tool

The tool category includes users’ verbalizations depicting problems in using the tool properly. Users were in many cases not sure how to add a specific pattern and therefore they applied the trial and error method. Modelers’ unfamiliarity with the tool often resulted in complaining or wondering why the tool was not working as they expected. An example of coding the tool category is “…parallel insert, did also not work…”\textsuperscript{14}.

Pattern

This category comprises the verbalizations of the users when trying out the different patterns in order to find the correct one. In this case they know what they want to model yet have difficulties in locating the correct or appropriate change pattern. Consequently, modelers make use of the undo command as well as the delete process fragment pattern. An example of coding the pattern category is “…conditional insert, this could be what I am looking for…”\textsuperscript{15}.

Domain

This category describes mistakes when mapping the domain (process description) to the tool. It illustrates modelers’ efforts in understanding the process requirements to be modeled while using the appropriate change pattern. An example of coding the domain category is “…let’s see whether this is correct (reading)…”\textsuperscript{16} which shows that the subject was trying to understand the textual description.

Critic

This category comprises users’ criticism on the usability of tool and proposals for tool improvement. An example of coding the critic category is “…I am not feeling free by modeling”\textsuperscript{17}.

\textsuperscript{14} “…parallel insert, (hat) also nicht funktioniert…”
\textsuperscript{15} “…conditional insert, genau das konnte sein was ich brauche…”
\textsuperscript{16} “…schauen wir ob das überhaupt richtig ist…”
\textsuperscript{17} “…aber frei fühlt man sich nicht bei modellieren…”
5.2.1 Change Pattern Support / Preflight

The transcript analysis for EU2 suggests that subjects consistently adopted similar expressions when experiencing difficulties with the change patterns.

5.2.1.1 Tool Category

The tool category captured 25 verbalizations from five out of six modelers. It was observed that when modelers were unable to add a change pattern they thought that the tool was not functioning “…it should be a conditional insert, but it does not work…”\(^{18}\), “---I just want to add here an activity, why it is not working? It should work…”\(^{19}\). Four out of six subjects were also wondering why the tool was not working exactly as they wished to, “…this is also not working…why not? This is not working the way I would like to…”\(^{20}\), “…nice this also did not work, why not?”\(^{21}\).

Moreover, when modelers were not sure how to add a change pattern they used the trial and error method or they were just clicking around the different elements, e.g., subjects were trying to conditional insert a process fragment “…let’s see how does it work (conditional insert)…”\(^{22}\), “…let’s try out once again the conditional insert…it also does not work…”\(^{23}\) in case it was the pattern they were looking for. They did the same with the serial insert pattern “…I would like here in between to add an activity, maybe a serial insert could work (serial insert), it is only a test…”\(^{24}\) as well as with the parallel insert “…parallel insert, also did not work, when I select an activity it also does not work. The last XOR connector (selecting)…”\(^{25}\). Furthermore, three out of six subjects faced difficulties regarding pattern combination when applying the parallel insert pattern, since they did not know that they should add an activity first (serial insert) and after selecting the newly inserted activity apply the parallel insert, e.g., “…let’s try out,
a serial insert, but I need a parallel insert, but let’s see first if this works…”26 and “…okay from this serial insert I am trying out now a parallel insert…”27.

5.2.1.2 Pattern Category

The following phenomena (16 verbalizations) appeared in the transcripts of five out of six modelers forming the Pattern category. It was not always straightforward for subjects to know which change pattern to use e.g., “…now I should find out how I can add an activity, (reading) serial insert no, condition insert also no, embed in conditional branch maybe…”28. Therefore, they were trying out the different patterns in order to find the most appropriate, “…ah exactly, it is probably embed in conditional branch, oh god no it is not…”29.

Five out of six subjects had to use the undo command, “…if I say I want here a conditional insert, oh no this is not working at all (conditional insert, undo)…”30, or delete a process fragment to restore their process model “…delete this fragment (delete process fragment), now I will try out a conditional insert (conditional insert)…”31, since they added the wrong pattern. A common error was the confusion of the rename command and the update condition pattern “… have selected the wrong element, wait a moment, let’s see if the other one is working, here you are, update condition it is not rename…”32.

5.2.1.3 Domain Category

Overall, the Domain category comprises eight verbalizations stemming from the transcripts of five out of six subjects. It was observed that subjects were striving to map the process requirements with the correct change pattern, “…however it is not clear if this is a parallel insert…but probably we are doing the run up checks…so I have

26 “… probieren wir a mal, na machma a serial insert, eigentlich will ich ein parallel insert machen aber schauen wir a mal wie das funktioniert…”
27 “…ok aus diesem serial insert versuche ich jetzt ein parallel insert…”
28 “…jetzt muss ich schauen wie man das macht eine Aktivität einzufügen, serial insert, es ist nicht, conditional insert ist auch nicht, embed in conditional branch vielleicht…”
29 “…ah genau embed in conditional branch ist es wahrscheinlich, achh gott das ist es nicht, oh man…”
30 “…wenn ich sage ich will hier ein conditional insert, aijai das funktioniert so überhaupt nicht…”
31 “…das fragment nochmal löschen diesmal probiere ich es mit einem conditional insert …”
32 “…das falsche eingeklickt, warten wir kurz, schauen ob das andere da… aua da, update condition ist nicht rename…”
to insert 2 activities (serial insert)…"³³. In this case modelers knew how to use the change patterns though they experienced issues in understanding the domain to be modeled feeling uncertain which pattern to use “…now I am adding a conditional insert, no a serial insert (serial insert, abort, reading)…”³⁴. Consequently, they had to read several times the process description “…probably another AND constructor (reading), so we are adding a parallel insert, or we could do a serial insert…”³⁵ and occasionally delete a process fragment “…now I did something wrong, because this step is not optional but…(delete process fragment), I delete once again this fragment (conditional insert)”³⁶.

5.2.1.4 Critic Category

The last category, Critic, captures one user’s verbalizations passing criticism on the tool “…this auto formatting is annoying…”³⁷, or making suggestions for improvement “…a copy function, which of course does not exist…”³⁸.

5.2.1.5 Summary

Overall, it was found that the 83.3 % of the subjects experienced problems when utilizing the change patterns tool (i.e., tool category) or selecting the appropriate change pattern (i.e., pattern category). As a result, modelers frequently used the trial and error method to understand the tool functionality, e.g., “…let’s try out once again the conditional insert…it also does not work” or to check out the different patterns in order to identify the most appropriate one. Finally, no great issues were identified regarding modelers understanding the process requirements (i.e., domain category) or criticizing the modeling tool (i.e., critic category).

³³ „allerding ist es unklar ob das parallel ist...hmmm aber wahrscheinlich machen wir die run up checks ...also 2 activities einzufügen..."
³⁴ „...als nächstes füge ich ein conditional insert, nein a serial insert ...”
³⁵ „...wahrscheinlich noch ein AND Konstruktor also wir machen ein parallel insert, oder wir machen serial insert...”
³⁶ „...jetzt habe ich einen Fehler gemacht weil der Schritt ist nicht optional sondern…das Fragment nochmal lösen...”
³⁷ „...dieses auto format nervt...”
³⁸ „...eine copy Funktion die natürlich nicht gibt...”
5.2.2 Change Pattern Support / Mortgage

The analysis of Mortgage/CP transcripts based on Grounded Theory provided considerably more data than the preflight protocols regarding users’ difficulties in using change patterns. Nevertheless, the process of open coding analysis resulted again into the same categories explained in Section 5.2 with no new aspects found.

5.2.2.1 Tool Category

A total of 37 verbalizations produced by all users compose the Tool category for the Mortgage/CP transcripts. The Mortgage/CP transcripts illustrated that users were often complaining about the tool not working correctly, “…(update condition) no mortgage (layouting) it is strange that it does not work correctly…”\(^{39}\), “…okay it does not seem to work, ah when it is green then it works? Really?”\(^{40}\). It was depicted that modelers (six out of six) tend to ask themselves questions when they experience troubles with the tool e.g., “…should I select it again? No! Now it works (parallel insert)…”\(^{41}\), “…(update condition) this is strange, why is this here? Abort…”\(^{42}\), “…(parallel insert, layouting) I cannot delete it, but it is supposed to work ,or not?…”\(^{43}\). Five out of six modelers had used the trial and error method in order to test if the outcome of a change pattern was what they were looking for, “…let’s see if I could do it, let’s try it out (conditional insert)…”\(^{44}\), “…I do not really know how this should be done, but I will just try it out, so like trial error (layouting)…”\(^{45}\). They were also trying out how to add, “…now I will try something out, I will abort it and say…no it does not work, I will put an embed in loop there, okay it does not work…”\(^{46}\) or delete a pattern “…I will insert a loop (insert loop) which now I have to delete, so I will better do delete process fragment, but it is not so easy as I thought (layouting)…”\(^{47}\). Two subjects experienced great issues in adding or deleting change patterns “…I will now select just the activity

\(^{39}\) “…no mortgage, das ist komisch das des nicht richtig funktioniert…”

\(^{40}\) “…okay geht irgendwie nicht, emm, a immer wenn es grün ist dann geht es? Echt?…”

\(^{41}\) “…muss ich wieder markieren? Nein! Jetzt geht’s…”

\(^{42}\) “…das ist strange warum steht das? Abbrechen…”

\(^{43}\) “…ich kann es aber nicht löschen, aber das muss es wohl gehen oder? …”

\(^{44}\) “…mal schauen ob ich es zusammenkrieg, testma…”

\(^{45}\) “…ich weiss jetzt nicht wie genau das da gemacht werden soll … Ich probiere es einfach durch, so trial error mäßig…”

\(^{46}\) “…jetzt probiere ich was, ich breche es ab und sag… geht halt nicht, embed in loop machen, zwischen da und da, okay geht halt nicht…”

\(^{47}\) “…eine loop eingeführt die jetzt irgendwie wegbekommen muss und zwar sage ich einfach am liebsten delete process fragment aber das geht nicht so einfach wie ich grad sehe…”
and the end element, it does not work, the XOR join, this is also not working…”48, showing the highest task completion time among the subjects (approximately 40 minutes). The main problems were again observed with parallel insert, “…I do not know, how, how can I add three activities in a parallel…”49, “…(parallel insert) is also not working…”50 and with update condition “…(update condition) has no mortgage (layouting), it is really strange that it does not work…”51. Overall, the subjects showed by uncertainty when using the tool “…(serial insert) now I am not sure if this will add it there, I hope not, indeed it is added there (abort)…”52, “… (parallel insert, layouting) but I cannot delete it, but it should be working, no?”53.

5.2.2.2 Pattern Category

For five out of six modelers it was challenging to select the correct pattern to use “…parallel insert it is not working, oh I need this here (conditional insert)…”54, often confusing the patterns with each other, “…it should be serial insert, eh I mean conditional insert…”55 The 13 verbalizations of the Pattern category also provide evidence for the subjects’ unfamiliarity with change patterns, “…embed in loop, no, embed in conditional branch?…”56, “…I should put a condition here, eh stop not parallel insert, but embed in conditional branch, no undo undo…”57 lacking knowledge regarding the semantics of some specific patterns, such as embed in conditional branch, “…but I cannot close it, ok what does it mean embed in conditional branch?…”58, as well as misinterpretation of rename activity and update condition, “…rename activity, oh this is not an activity, haha (update condition)…”59.

---

48 “…ich markiere jetzt einfach nun mal die Aktivität und das End Ereignis, geht immer noch nichts, das XOR join und des geht immer noch nicht…”
49 “…ich weiß nicht wie, wie ich drei Aktivitäten parallel become…”
50 “… parallel insert geht nicht…”
51 “…has no mortgage, das ist komisch das des nicht richtig funktioniert…”
52 “…jetzt bin ich i gespannt ob des nit da aini macht, hoffentlich nicht, doch macht er…”
53 “…ich kann es aber nicht löschen, aber das muss es wohl gehen oder?…”
54 “…parallel insert geht nicht, aaa ich brauche das hier…”
55 “…in Prinzip serial insert, eee conditional insert…”
56 “…embed in loop, no, embed in conditional branch?…”
57 “…eine condition einbauen, e stop nicht parallel sommes embed in conditional branch, nää undo undo undo…”
58 “…aber wie kriege ich des dass es zusammenschließt, ok was heisst das embed in conditional branch?…”
59 “…rename activity, ah das ist ja keine activity, haha…”
5.2.2.3 Domain Category

The Domain category captured a higher number of users’ verbalizations (17), when compared to the preflight model pointing towards problems in mapping the mortgage textual description to the modeling tool. Modelers were trying to understand the requirements to be modeled, for example to the modeling tool two subjects (subject nr. 5 and nr. 6) it was unclear from the process description if they should add a serial or a conditional insert, “…because I have difficulties, because now I do not know if I should first add a serial insert and then the other one, to connect something with a decision, or just add a conditional insert…”60 “…this means we add here an insert, ja conditional (conditional insert), let’s see if this is right (reading), I think I should add in front a serial insert (reading)…”61 In order to overcome the encountered difficulties subject nr. 6 had to draw first part of the process model on a piece of paper in an attempt to understand the modeling domain, “…so I have to say I have troubles in doing it afterwards, the problem is that I do not know how the process should look like, okay I will try something different now, I will draw it first…”62 Moreover, the subject denoted that it would be easier if he had read the whole process description before starting with modeling, “…Does it make sense? Because here the check, which means I will do it (undo) like that (conditional insert, abort), so it would have been really practice if I have read the process description first…”63 Overall, it is observed once again that when modelers face difficulties in understanding the modeling domain, they repeatedly read the process description and tend to delete process fragments, e.g., “…(reading) I will just remove this (delete process fragment), so here it should not be a loop but a

---

60 “…weil ich komme schon in Schwierigkeiten weil jetzt nicht weiß ob ich jetzt erst ein serial insert machen soll und dann quasi des, darauf, irgendwas mit einer Entscheidung verbinden soll, oder gleich ein conditional insert…”
61 “…d.h. wir machen da a insert ja, conditional schauen wir ob das überhaupt richtig ist, ich glaube ich muss davor ein serial insert machen…”
62 “…also ich muss sagen ich tue da ein bisschen schwer mit diese, im Nachhinein des zu machen, das Problem ist ich weiß wie der Prozess aussehen sollte, ok dann versuche ich es jetzt mal ganz anders ich zeichne es wieder auf so…”
63 “…ee macht es dann sinn? Weil hier ist der check, d.h. ich mach es mal so also wär mal praktisch erstmals alles durchgelesen zu haben…”
condition…” or use the undo command “…okay, once again another way, I will start with this check mortgage first(undo command x2, delete process fragment)…”

5.2.2.4 Critic Category

Three out of six modelers criticized the tool’s functionality, with two of them stressing that change patterns restrict free modeling, making the user feel that he is only allowed to model in a pre-specified way “…it worked, but you do not feel free by modeling, I always have to think that I am able to do only what is in there, otherwise I cannot do anything…” or with pre-specified ‘building blocks’ which lead the user in trying out all the available patterns without active thinking “…so this point was really difficult for me, because I could not add single process steps by hand, but I had predefined blocks which were somehow translated to some expressions, this is missing, therefore I just started trying out and testing without actually thinking what it should really be put in there…”.

5.2.2.5 Summary

Summarizing, the results show that all subjects experienced, at least once, difficulties in utilizing change patterns support when creating the mortgage process. At least five out of six modelers reported problems with applying the tool (i.e., tool category) or understanding the change patterns (i.e., pattern category) as well as the modeling domain (i.e., domain category), while only one subject found to be familiar with change patterns. The findings suggest that understanding the modeling domain and mapping the tool to the textual description was challenging for the mortgage task when compared to the preflight task. Finally, the modelers criticized the functionality of the modeling tool, suggesting difficulties in utilizing change patterns (i.e., critic category).

---

64 “…ich mach es jetzt einfach auch noch weg also da dahinter muss jetzt kein loop sein sondern a condition…”
65 “…ok nochmal ganz anderer Ansatz, ich fange mit diesem check mortgage an…”
66 “…Geworden aber frei fühlt man sich nit bei modellieren da denke ich mir immer ich kann nur das machen was da drinnen steht, da weil kann ich sonst gar nix machen…”
67 “… also in dem Punkt habe ich mich wirklich schwer getan, wenn ich nicht per Hand irgendwie einzelne process-schritte einbauen kann, sondern wenn ich quasi vorgegeben Bausteine habe, die irgendwie nur im irgendswelchem Worten ausgedruckt sind, dann fällt das einfach dieses aus, wobei ich bin es einfach in ausprobieren und testen reingegangen ohne wirklich zu überlegen was müsste jetzt dahin…”
6 Conclusion and Limitations

6.1 Summary

This exploratory study seeks to compare change primitives and change patterns and their impact on the PPM (RQ1). Moreover, the challenges process designers are facing when creating process models using change patterns is investigated (RQ2), since little is known about the PPM when utilizing change patterns. In order to address the research questions, two modeling sessions were conducted. In the following paragraphs, the major findings for each research question are briefly outlined.

RQ1 aims at comparing the process of creating a formal process model using change primitives vs. change patterns support from an informal description. Based on the results of the quantitative analysis the following conclusions can be drawn. The analysis of using change primitives suggests that modelers are mainly involved in creating the process models in a straightforward series of modeling steps following the process description, without being engaged in validating their model or planning their actions. The modelers also seem to be more comfortable in using change primitives when they are asked to create a challenging process model.

In case of change patterns, the quantitative analysis suggests that modelers were mainly focused on modeling, however not in a continuous sequence of interaction. Their modeling steps were often interrupted, in order to refer again to the process description and improve the process model’s optical appearance. Furthermore, a considerable amount of time was invested into planning their modeling actions. The challenges faced by modelers when creating more complex process models were higher due to the structural restrictions imposed by changes patterns.

Overall, for both modeling tools a similar amount of reconciliation phases was depicted in the experimental units, implying that reconciliation is independent on the tool support provided. A more in depth analysis of the comprehension phases of change primitives pointed towards modelers’ concerned mainly with domain comprehension and utilizing their time for understanding the textual description. For change patterns, in turn, the modelers’ time invested in domain comprehension was constraint by the effort needed for tool understanding and validation, indicating the issues they faced by utilizing
change patterns. As a result, it is proposed that change patterns are considerably more difficult to use, especially for inexperienced modelers, compared to change primitives.

The problems faced by the process designers are addressed by RQ2 and summarized into four categories, tool, pattern, domain and critic. When faced with more challenging process structures, a higher amount of verbalizations illustrating modelers’ problems in using change patterns or understanding the process description was identified.

The research goal of this exploratory research was fulfilled as a comparison between change primitives and change patterns was achieved. Further, this thesis raises awareness regarding the improvement of the modeling approach based on change patterns and confirms the findings of Weber et al. (2013) that the creation of process models utilizing change patterns impacts the PPM. The topic of investigating the PPM when utilizing change patterns has the potential for promising future research.

6.2 Limitations

The research is characterized by some limitations. While the modelers participating in this study were familiar with change primitives they were rather inexperienced with change patterns support. Therefore, the question can be raised whether with more experienced modelers the same observed differences among the two modeling tools would have occurred. Another possible thread regarding generalization of the results of this exploratory study is the small sample size (12 participants). This limitation could be mitigated to some extent by applying method triangulation, i.e., analysis of code patterns and qualitative analysis based on grounded theory. Another limitation could be the use of only two different modeling tasks in this exploratory study. In addition, differences on the model characteristics may affect the generalization of results.

6.3 Future Work

Future work could replicate the study with more experienced modelers or within another environment, instead of CEP, supporting the PPM. Similarly, a higher number of modeling tasks could be used. Further, additional perspectives on the PPM should be explored. For instance, additional phases under the consideration of time i.e., the time consumed for each modeling phase during the creations of the process model, could be
exploited for identifying challenging aspects during the creation of process models. Finally, another aspect under investigation could be recurring challenges faced by modelers during the PPM using additional change patterns.
Appendix

Appendix I: Instructions

Experience the Impact of Change Patterns on the Modeling Process

You are participating in a modeling session where you will work on two modeling tasks. You will experience the impact of change patterns versus change primitives. Moreover, you will be exposed to the Think aloud method, as you are requested to talk aloud during the whole modeling session.

Think Aloud

Think aloud method is used to identify changes in the knowledge during repeated problem-solving on a single task. Thus it can be used to investigate differences in problem-solving abilities between people, differences in difficulty between tasks, effects of instruction and other factors that have an effect on problem-solving.

Think aloud is a method where people are asked to work on a model and instructed to think aloud. What they say is recorded and used as data for analysis of the design process. This is a very direct method to gain insight in the knowledge and methods of human problem-solving.

- Therefore, you are expected to talk aloud, while working on the two modeling tasks. You should keep on talking; speak out loud whatever thoughts come to your mind during the whole experiment phase, especially while performing the modeling task.
- You are allowed to use your own language.
- No interruptions or suggestive prompts or questions are allowed as you are encouraged to give a concurrent account of your thoughts and to avoid interpretation or explanation of what you are doing; you just have to concentrate on the task.
Change pattern

Change patterns reduce the complexity of process changes and raise the level for expressing changes by providing abstractions which are above the level of single node and edge operations.

You will be provided with the following change patterns:

- **Serial Insert**: a new process fragment X is inserted between two activity sets
- **Delete Process Fragment**: a process fragment X is deleted from the process model
- **Conditional Insert**: a new process fragment X is inserted between two activity sets with additional condition –XOR-
- **Embed in Loop**: a loop construct is added to the process model which surrounds an existing process fragment
- **Parallel Insert**: a new process fragment X is inserted between two activity sets without additional condition –AND-
- **Embed in Conditional Branch**: a process fragment is embedded into a conditional branch such that it is only executed if a particular condition is met
- **Update Condition**: a transition condition in the process model is updated
- **Rename Activity**: an activity is renamed
- **Undo**: the last action is reversed
Instructions

During the modeling session you will work on two modeling tasks. The modeling session will be structured in the following way:

- Demographic Survey
- Modeling Tool Tutorial
- Modeling Task 1: Pre-Flight Process
- Modeling Tool Tutorial
- Modeling Task 2: Mortgage Process
- Concluding Survey

Please be aware that we will confront you with difficult models, requiring quite a mental effort to create. Please do not feel discouraged by these models, but rather understand the modeling tasks as challenges!

Please consider the following hints while answering the questions of the Demographic Survey:

- You must answer all questions in order to proceed unless stated otherwise.
- In answering the questions within the questionnaire, please provide responses only based on your own perceptions or beliefs.

Before every modeling task you will be presented with an id for your model. Please write the id in the corresponding box next to the task description.

To this end, we will present you with two modeling assignments that are approximately of equal difficulty. However, for one exercise you will be given change pattern support, whereas for the other one, you will have workflow patterns, as you already know.

- Please keep in mind that you should think aloud during the whole modeling session, speak out loud and clear whatever thoughts come to your mind while performing the modeling task.
Exemplary Process Model

Below you can find an exemplary process model created using a subset of BPMN (Business Process Modeling Notation); it illustrates all available constructs for modeling (start- and end events, activities, AND- and XOR gateways as well as sequence flows).
Process Model 1: Pre-Flight Process

Short Description

The process describes the steps to be conducted by an airplane’s crew before take-off. If you are finished with modeling, please use the Finish-Modeling-Button on the top left to proceed to Modeling Task 2.

Please keep in mind that you should think aloud during the whole modeling session, speak out loud and clear whatever thoughts come to your mind while performing the modeling task.

Process Description

In the following, the pre-flight process for conducting a general aviation flight is described. First, the pilot has to check the weather. Optionally, the pilot can then file a flight plan. This is followed by a preflight inspection phase of the airplane, where the pilot checks the engine as well as the fuselage of the airplane. Both activities can be conducted independently of each other. For large airports, the pilot calls Clearance Delivery to get the engine start clearance. If an airport has a tower, the pilot has to contact Ground to get taxi clearance; otherwise the pilot has to announce taxiing himself/herself. These activities are followed by taxiing to the run-up area and performing the run-up checks to ensure that the airplane is ready for flight. If the airport has a tower, the tower is contacted to get take-off clearance, otherwise take-off intentions have to be announced. Finally, the pre take-off process is completed with the take-off of the airplane.
Process Model 2: Mortgage Process

Short Description

In the following, the verification of the process of a bank handling a customer’s mortgage request is described. For this modeling task it is required to use change patterns. If you are finished with modeling, please use the Finish-Modeling-Button on the top left to proceed.

Please keep in mind that you should think aloud during the whole modeling session, speak out loud and clear whatever thoughts come to your mind while performing the modeling task.

Process Description

As first step, the bank checks whether the customer has already a mortgage. If the customer has no mortgage yet, the mortgage application is registered locally. Otherwise, if the customer already has a single mortgage, the headquarters need to be informed.

Afterwards, the bank performs the following checks in parallel:

- the mortgage must not exceed 80% of the property's value
- the applicant is currently employed
- the applicant is not internally listed for low payment moral

After all checks have been performed, the bank evaluates the results. If one of the checks turns out negative, the application is rejected and closed. Otherwise, general information about the application is registered. Subsequently, the bank analyzes the mortgage in detail: If the mortgage is below € 1.000.000, confirmation by a single person is sufficient. For mortgages equal or larger than € 1.000.000, supervisor approval is needed additionally.
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM</td>
<td>Business Process Modeling</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Modeling Notation</td>
</tr>
<tr>
<td>CEP</td>
<td>Cheetah Experimental Platform</td>
</tr>
<tr>
<td>CP</td>
<td>Change Pattern</td>
</tr>
<tr>
<td>D</td>
<td>Comprehension Understanding Domain</td>
</tr>
<tr>
<td>EU</td>
<td>Experimental Unit</td>
</tr>
<tr>
<td>HUs</td>
<td>Hermeneutic Units</td>
</tr>
<tr>
<td>LTM</td>
<td>Long-term Memory</td>
</tr>
<tr>
<td>M</td>
<td>Modeling</td>
</tr>
<tr>
<td>P</td>
<td>Comprehension Planning</td>
</tr>
<tr>
<td>PAISs</td>
<td>Process-aware Information Systems</td>
</tr>
<tr>
<td>PPM</td>
<td>Process of Process Modeling</td>
</tr>
<tr>
<td>ProM</td>
<td>Process Mining Framework</td>
</tr>
<tr>
<td>R</td>
<td>Reconciliation</td>
</tr>
<tr>
<td>RQ</td>
<td>Research Question</td>
</tr>
<tr>
<td>T</td>
<td>Comprehension Understanding Tool</td>
</tr>
<tr>
<td>V</td>
<td>Comprehension Validation</td>
</tr>
<tr>
<td>VPA</td>
<td>Verbal Protocol Analysis</td>
</tr>
<tr>
<td>WM</td>
<td>Working Memory</td>
</tr>
</tbody>
</table>
Bibliography


