Configuration vs. Adaptation for Business Process Variant Maintenance: An Empirical Study

Markus Döhring

SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany

Hajo A. Reijers

Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Sergey Smirnov

SAP AG, Am Schimmersfeld 5, 40880 Ratingen, Germany

Abstract

Many approaches for process variant management employ a reference model for deriving a target variant either using configuration or adaptation mechanisms. What is missing at this stage is an empirical insight into their relative strengths and weaknesses. Our paper addresses this gap. We selected C-YAWL and vBPMN for a comparative, empirical user study. Both approaches center on a reference process, but provide different types of configuration and adaptation mechanisms as well as modularization support. Along with this aspect, we investigate the effect of model complexity and professional level on human process variant modeling performance. Given unlimited processing time, we could not show that complexity or the participant's professional level significantly impacts the task success rate or user contentment. Yet, an effect of model complexity can be noted on the execution speed for typical variant maintenance tasks like the insertion and deletion of process steps. For each of the performance measures of success rate, user contentment and execution speed, vBPMN performs significantly better than C-YAWL. We argue that this is due to vBPMN's advanced modularization support in terms of pattern-based process adaptations to construct process variants. These insights are valuable for advancing existing modeling approaches and selecting between them.

Keywords: process modeling, process variants, process configuration, process adaptation, user experiment, model maintainability

1. Introduction

The ability to rapidly tailor a process to changing business requirements is among the top drivers of companies to employ Business Process Management (BPM) technology [1, p. 3][2]. In this context, it is often the case that new business requirements have to be taken into account over time. Such requirements may supplement existing ones, leading to the need for a slightly different behavior of the business process as executed previously. This motivates an efficient *process variant* modeling approach. For example, a sales order process that is run by a global company enforces the execution of a liquidity check at the start of the process for customers in Asia only, while for European customers this step is skipped. At some point in time, the company may want to execute a liquidity check for distinct European countries, but only at the end of the process. This necessitates the introduction of an additional process variant.

According to [3][4, p.4], variants of a process model are defined as "similar-but-different" from each other, i.e. they have at least one feature in common and one feature in which they differ. In practice, however, this definition of the term *variants* tends to be problematic, since one often finds at least one commonality or invariant [5] between two objects. It seems more practical to adopt a definition as from [5], where it is clearly stated that the delta between two objects should be small compared to their commonalities. For this paper, we consider structural characteristics like sequencing or branching behavior in the process graph as relevant features [6].

A typical pitfall that should be avoided when modeling process variants notably relates to the creation of redundancy by applying a copy-and-paste (multi-model) approach [7]. In such an approach, an existing process model is cloned and tailored to the new business requirement. Since there is no shared part list for the (loosely) corresponding process models, it is very hard to enforce global changes if the number of variants is high. One example would be the insertion of a new task that should be executed within all process variants, requiring the manual tailoring of each single process model.

Furthermore, as business requirements change over time, maintenance operations on the distinct process variants need to take place. In this respect, maintenance operations may, for example, relate to the variant-specific insertion, skipping or re-routing of process steps [8]. Two important characteristics which determine model *maintainability* are *understandability* and *modifiability*. This means that to properly maintain a process model, a user is not only required to correctly understand the process model, but must also be enabled to properly modify it according to a specified adjustment task. For this paper we subsume the terms *understandability* and *modifiability* under *maintainability*. The reason is that a lot of work related to software measurement exists, which considers understandability as an influencing factor for maintainability [9–13].

In order to address the challenges we described above, a broad variety of modeling approaches for process variants has emerged in recent years, e.g. [7, 14–23]. For these approaches, a comprehensive survey which classifies them according to multiple feature dimensions was conducted in [24]. While we do not claim these feature dimensions to be exhaustive, five of the most commonly addressed dimensions in scientific literature are explained in the following:

Variant Construction Direction. Process variants can be constructed and maintained using generally different strategies [4, 7, 18], including process *configuration* and *adaptation*.
When using process *configuration*, typically the first step is to create a reference process which comprises the behavior of all considered variants. From this all-embracing reference process model, a variant can be derived by eliminating elements which are not relevant for the given context. This corresponds to one way of *process model abstraction* [25].
When using process *adaptation*, the reference process is not necessarily constructed as the superset of all variants. Instead, a set of change operations [8] as, for example, the insertion, deletion, conditional-skipping or loop-embedding of tasks is defined. An appropriate reference model can then be selected by minimizing the change operations, potentially considering variant usage frequency, which need to be applied to the reference process for obtaining the required process variant [26]. Since these change operations can be perceived as extensions to the reference process, this strategy is strongly related to the concept of *inheritance* in process models [27].

Modularization Support. Modularity is usually referred to as a system property, which states that the system is composed of smaller subsystems. These subsystems in turn are independently manageable and function together as a whole [28, 29]. Decomposition is referred to as a stricter subconcept of modularity, in which modules need to be designed such that the interdependencies to other modules are minimized. Conveying this to process variants, modularization manifests itself primarily in variable regions spanning multiple process elements, which can be subject to change as a whole. Modularization may also manifest itself in reusable process fragments or respectively change macros which apply complex modifications on the reference process, as for example the wrapping of a process fragment into a loop construct or timeout exception handler.

Runtime Variant Construction. In some cases it is necessary to alter an instantiated process during runtime to a variant which had not been considered before starting it [8]. The ability to construct new variants at runtime, for example, refers to the inserting a new task, such as a special approval task.

Data-Flow and Resource Variability. Besides the modeling of control-flow variants, variability in processes can also relate to many other perspectives [30]. Most prominent are the data-flow and the resource perspective [31, p. 2]. Data-flow variants for example specify different types of objects to be passed within a static control-flow, while resource assignment variants specify different processors like clerks or computers for a process task.

Existing approaches for process variant management concentrate on different parts of the above dimensions. Most of them feed their claim to support the maintainability of process variant models by presenting case studies, which provide an impression on how a process variant model would be realized using the respective approach. No quantitative empirical evidence, however, exists on the actual benefits and drawbacks of the distinct approaches for the human process variant modeler, for instance regarding the maintainability of the process model. Moreover, only few insights are reported regarding the scalability of the approaches, i.e. what happens if the variant model gets complex and needs to be maintained over time.
Since control-flow is considered as the essential perspective in process modeling [31, p. 2] and very limited empirical prior work exists on process variant modeling, in this work we first focus on the two control-flow related feature dimensions described before, i.e. variant construction direction and modularization support. We leave the examination of other feature dimensions to future work. For the variant construction direction, effects on maintainability have not yet been thoroughly examined. This dimension is, however, recognized as the main classification criterion for approaches to tailor reference models [4]. For modularization support, a general positive effect on understandability has already been established empirically in [28]; moreover, modularization is described as a subcharacteristic of maintainability in the ISO 25010 standard [32].

The key contribution of this paper, to the best of our knowledge, is the first comparative user study on the effects of (1) process complexity, (2) the professional level, and (3) the type of the selected process variant modeling approach on variant maintenance tasks. Complexity concerns for example the number of nodes and their interconnectedness in a process (variant) model. The professional level concerns the experiences of the participant, distinguishing e.g. students from senior professionals. The particular difference of process variant maintenance tasks compared to regular process maintenance tasks is that usually the whole set of variants has to be taken into account. For example when manipulating the reference process model, the effect on all variants which are derived from it needs to be considered. We expect that the insights delivered by our work support the further development of existing modeling approaches. Also, the selection between these for real-life variant management is facilitated, because the benefits and drawbacks of distinct concept constructs for variant modeling can be estimated better based on empirical results.

This paper is structured as follows. In Section 2, we develop and specify the propositions which will be investigated to add upon the existing body of empirical studies on process model maintainability, with a specific focus on variant management. The two competitor process variant modeling approaches which are employed within the user study are introduced in Section 3. The process variant scenarios and their corresponding realization with the two approaches are discussed in Section 4. The setup and the results of the study are presented in Section 5. Related work is discussed in Section 6, while Section 7 concludes this paper.

2. Scope and Propositions

There is a body of recent studies dealing with the examination of understandability [28, 33–35] and maintainability of process models [10, 11, 36–38]. Some of the work also examines the differences between imperative and declarative process modeling languages regarding understandability and maintainability. The focus of our work is clearly on process variant modeling using mainly imperative languages [39, p. 560]. The studies largely agree in their findings upon the general negative effect of process model complexity factors like the type and number of nodes or the connectivity level between nodes on understandability and maintainability factors like maintenance task execution speed, success rate or user contentment. A corresponding detailed correlation analysis relating structural complexity factors to the effectiveness, efficiency and subjective perception of process model understanding and modification tasks is provided in [11]. We assume that the interrelation of process model complexity and maintainability can also be conveyed to process variant management and correspondingly formulate the following set of propositions:

P1: *Process model complexity decreases the success rate for process variant maintenance tasks.*

P2: *Process model complexity decreases the execution speed for process variant maintenance tasks.*

P3: *Process model complexity impairs the subjective perception (convenience and ease-of-use) for process variant maintenance tasks.*

The influence of personal factors like the participant's professional background have not been intensively researched so far [33]. Comparative experiments on process model understanding with students and professionals in [33], however, suggest that at least on a coarse-granular level, both groups perform similarly. For our study, we analogously summarize this aspect as the participant's professional level, i.e. whether the person is a student or a senior. Based on the cited work, we assume no positive or negative effect caused by the professional level and correspondingly formulate the following set of propositions:

P4: *The professional level does not impact the success rate for process variant maintenance tasks.*

P5: *The professional level does not impact the execution speed for process variant maintenance tasks.*

P6: *The professional level does not impact the subjective perception (convenience and ease-of-use) for process variant maintenance tasks.*

For the scope of this study, we furthermore consider approach characteristics in terms of variant construction direction and modularization support as influential factors for process variant maintainability as introduced in Section 1. The two dimensions are highlighted in light-gray in Table 1 together with a selection of representative process variant management approaches. We also include the additional feature dimensions described in Section 1 which are out of scope for this paper in dark-gray. We exemplarily categorized ABIS [19], AgentWork [16], AO4BPMN [20], C-EPC [14, 30], C-MPSG [23], C-YAWL [15], Design by Selection [17], Multi-Perspective Variants [22, 40], PESOA [18], Provop [7] and vBPMN [21, 24] and according to the two introduced dimensions. For a categorization along the modularization support dimension, we distinguish between approaches which only allow to define variability at a single element level and approaches which apply variability mechanism to multiple elements at once. These element sets can be statically defined as single-entry, single-exit (SESE) [41–43] segments within the process model graph. Alternatively, they can be dynamically determined for example based on a query taking into account structural or node label characteristics.

C-EPC and C-YAWL are configuration extensions for Event-Driven Process Chains (EPC) and Yet Another Workflow Language (YAWL) respectively. The extensions allow for example to assign variant constraints on the outgoing paths of gateway in a reference process model. If the constraints are not satisfied for a specific data context, these paths are omitted from the reference model to construct a specific process variant. The Multi-Perspective Variant approach works similarly, but relies on a tree-like representation of the process structure to apply configuration mechanisms.

AgentWork, vBPMN, AO4BPMN and Provop allow for the rule-based application of predefined change operations to a process instance at runtime. In contrast to AgentWork, vBPMN, AO4BPMN and Provop allow for the definition of modular higher-level change patterns which comprise multiple low-level change operations. The Design by Selection approach employs BPMN-Q [17] as a query mechanism to extract process fragments from a parent graph and embed them within a process variant. From the viewpoint of the variant, however, it is always only a single placeholder element which can be substituted with the queried fragment. Therefore we categorized it as a "single element-based" approach for process variant construction. The same holds for the variability points used by the PESOA approach, which combines variability mechanism based on restriction (configuration) and mechanisms based on combination (extension) to model process variants. ABIS allows do define reusable process fragments with multiple "docking nodes" and to weave them with a reference process for the construction of variants.

Finally, the Configurable Module-Based Process Structure Graph (C-MPSG) approach modularizes a reference process graph into a tree-like structure and allows for the configuration-based extraction of a subtree to form a process variant.

For our user study on process variant maintenance, we are especially interested in examining potential differences along

Table 1: Characteristics of Approaches for Process Variant Modeling (A=Adaptation, C=Configuration, F=Fragment-Based, S=Single-Element-Based)

	ABIS [19]	AgentWork [16]	AO4BPMN [20]	C-EPC [14, 30]	C-MPSG [23]	C-YAWL [15]	Design by Selection [17]	Multi-Perspective Variants [22, 40]	PESOA [18]	PROVOP [7]	vBPMN [21, 24]
Control-Flow Construction	A	A	A	C	C	C	A	C	A	A	A
Modularization Support	F	F	F	S	F	S	S	S	S	F	F
Runtime Variant Construction	☐	☑	☐	☐	☐	☐	☐	☐	☑	☑	☑
Data-Flow Variability	☐	☑	☐	☑	☐	☐	☑	☑	☑	☐	☑
Resource Perspective Variability	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐

the two selected dimensions (control-flow construction direction and modularization support) for tasks dealing with the adjustment or extension of a variant model. Since a full comparative user evaluation of representative approaches for all possible feature combinations is hardly feasible, we selected two polar process variant modeling approaches: vBPMN as a fragment-based process adaptation approach and C-YAWL as a single element-based process configuration approach. In analogy to the process model complexity factor, we correspondingly assume that the type of the chosen variant management approach influences the execution of typical process variant maintenance tasks. As we lack corresponding empirical evidence from existing work, we cannot make any assumptions on which of the approaches will perform better. Consequently, we formulate the following set of propositions:

P7: *There is a difference in the success rate for process variant maintenance tasks between using an adaptation approach with fragment support (vBPMN) and a configuration approach without fragment support (C-YAWL).*

P8: *There is a difference in the execution speed for process variant maintenance tasks between an adaptation approach with fragment support (vBPMN) and a configuration approach without fragment support (C-YAWL).*

P9: *There is a difference in the subjective perception (convenience and ease-of-use) for process variant maintenance tasks between an adaptation approach with fragment support (vBPMN) and a configuration approach without fragment support (C-YAWL).*

In our study, we are concerned with evaluating which strategy performs better when the process variants need to be adjusted or extended over time due to changed business requirements. It is not within the scope of our study to examine the effort required to construct the initial variant model.

3. Selected Approaches for Process Configuration

In the following, we describe the two selected process variant modeling approaches for our comparative user study. C-YAWL and vBPMN are introduced together with their basic conceptual components and an application example in terms of a repair process.

3.1. C-YAWL as a Representative Approach for Single-Element Process Configuration

C-YAWL [15] extends the YAWL [44] process modeling language, which is based on Petri nets [45], with mechanisms for process configuration.

3.1.1. Setting Port Configurations for Different Variants

C-YAWL uses *ports* within a process model as variability points. A port corresponds to an incoming or outgoing sequence flow of a YAWL process step. As illustrated in Figure 1, *blocking* and *hiding* are introduced by C-YAWL as the main two configuration concepts which can be applied to *ports*. A blocked port means that for this particular variant, no token can be received resp. emitted via this port. When setting a port to hidden, it means that for this particular variant a token which is received at this port is directly forwarded to the outgoing part of the process step. The action which normally would be conducted when executing the process step is correspondingly skipped. Hiding consequentially only makes sense to be applied to incoming sequence flows.

3.1.2. Deriving Process Variants from a Reference Model

The definition of a process variant in C-YAWL corresponds to a set of port configurations applied on a reference model. This means, that the reference model must already contain all possible elements (nodes and sequence flows) which occur in any variant. The upper part of Figure 2 shows a corresponding example repair reference process model in C-YAWL. After the problem analysis and spare parts ordering steps, the actual repair is conducted. In parallel, an advertisement is sent out to the customer to encourage him to buy a new product instead of repairing the old one. In some cases, if the repair is not finished after 1 week, a notification is sent to the customer. This waiting and notification sequence is canceled after the "Perform Repair" step has finished execution, which is realized by the succeeding dummy step which has the lowest branch in its cancellation set.

Depending on a context factor [46] like *country*, it could for instance be the case that in the USA, an advertisement should be sent and the customer should be notified after 1 week, while in Germany these steps should not take place. A corresponding

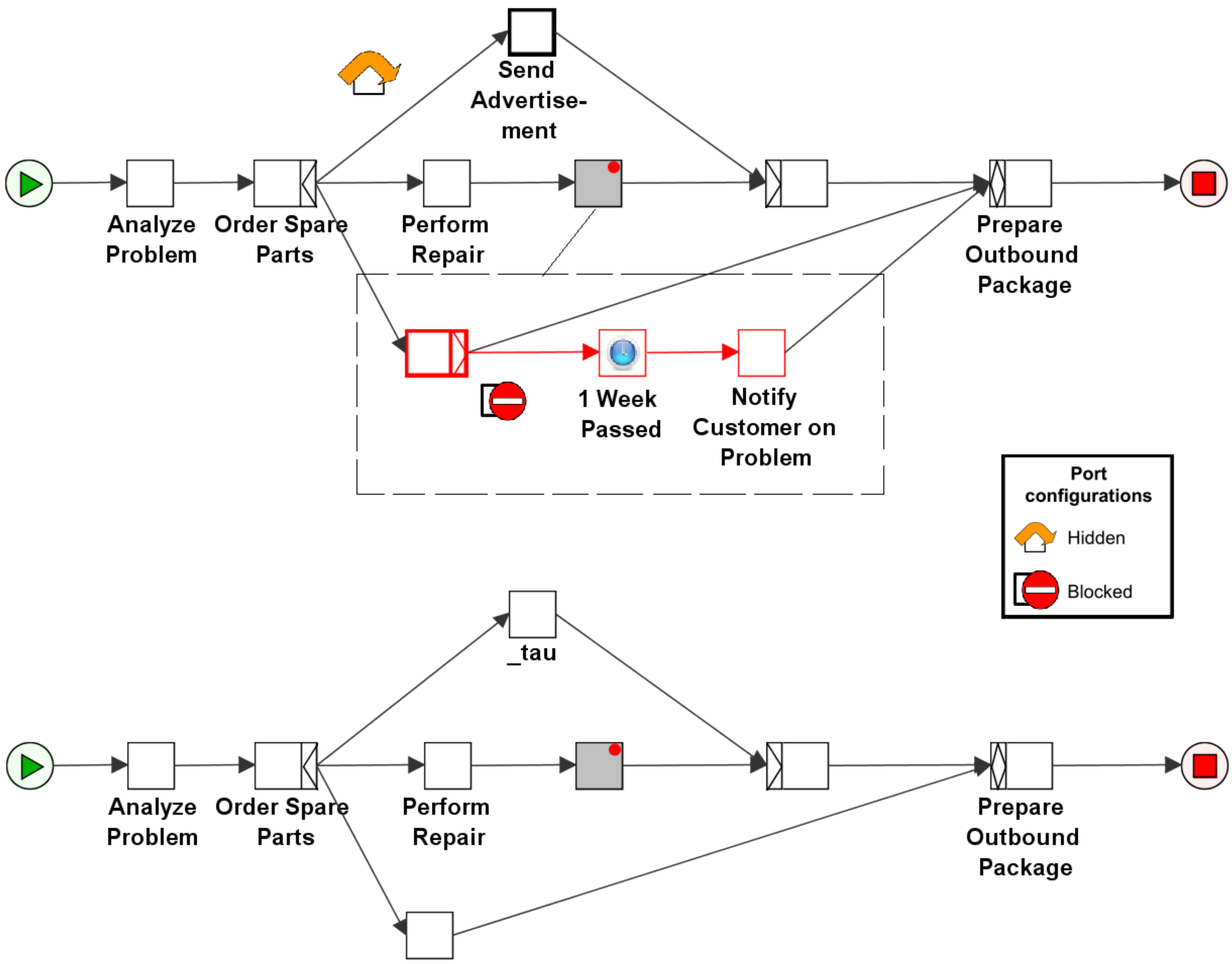


Figure 2: Derivation of a Process Variant from a C-YAWL Reference Process by Blocking and Hiding Ports

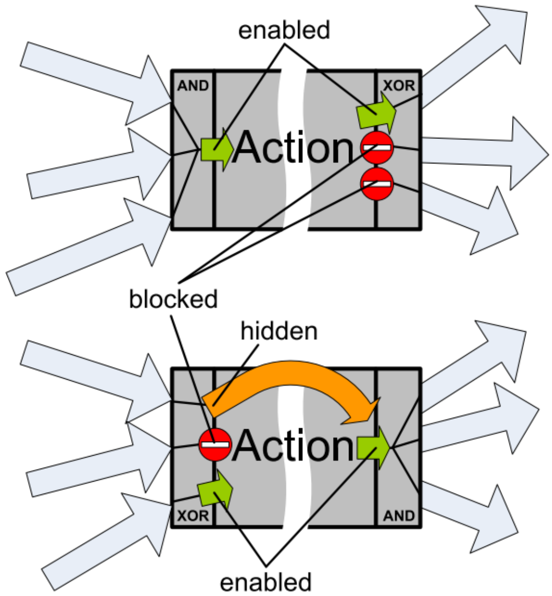


Figure 1: Blocking and Hiding Ports in C-YAWL [15]

variant of the reference process from Figure 2 for Germany can be derived by *hiding* the input port of the "Send Advertisement" step and setting the output port of the exclusive gateway leading to the waiting and notification branch to *blocked*. The resulting derived process graph is shown in the lower part of Figure 2. The process now comprises a purely sequential execution of the four main process steps, not considering the sending of an advertisement or the notification of the customer after 1 week.

For some process variant scenarios, it might be necessary not only to consider one single context factor like *country*, but also other context factors like *customer type*, *order value* or *order priority* for variant construction. In order to deal with process variants which depend on multiple context factors, C-YAWL proposes a questionnaire approach [47]. Users are guided through the questionnaire by a set of constraints, for example preventing the posing of a question which can be answered by the result of an already completed question. The port configurations are then set based on logical constraints on the set of answers. In our study, we do not focus on the end user who actually customizes the more or less fixed and spoonfed variant model, but on the process modeler whose task is to maintain the overall process variants and adjust them on a structural level according to changing business requirements. Therefore, we consider such a questionnaire approach as a loosely coupled add-on for end-users, which is not subject to our evaluation.

3.2. vBPMN as a Representative Approach for Fragment-Based Process Adaptation

In earlier research contributions [21, 48–51], we introduced *variant BPMN* (vBPMN) as an extension of the standard BPMN2 metamodel [52] to address requirements related to process variant management. In the following, a brief introduction to vBPMN is provided based on the repair process example.

The vBPMN approach consists of three main conceptual components: a facility to mark variant (adaptive) segments in a reference process, a catalogue of reusable adaptation patterns and a set of adaptation rules specifying where and for which business situation the patterns have to be applied. The conceptual components are explained in the following.

3.2.1. Adaptive Process Segments

As described in earlier work as a "pocket of flexibility" [53] or "adjustment point" [54], it is often desirable to have a clear separation of static parts of a process and those parts which may be adapted at design- or runtime. Therefore vBPMN introduces two new node types to indicate the start Ⓘ and end Ⓘ of such an "adaptive segment" within a regular BPMN process definition. An adaptive segment has to be single-entry, single-exit (SESE) structured to facilitate the use of adaptation patterns, which are explained in the next paragraph. The rest of the process, however, i.e. those parts which do not reside within an adaptive segment, do not need to be SESE structured. In practice, the SESE structuring of adaptive segments tends to be a mild restriction. According to [55], in a case study it has been found that 95% of process models from different domains were SESE-structured or could be transformed into a SESE-structured representation.

An example for the demarcation of an adaptive segment is provided in the left upper part of Figure 3. It shows the basic repair process, for which the "Perform Repair" task can be subject to variant-specific adaptations.

3.2.2. Adaptation Patterns

The structural adaptations which can take place for an adaptive segment are provided in an extensible pattern catalogue [49]. Many systems realize change patterns [8] (e.g. insertion or skipping of tasks), exception-handling patterns [56] (e.g. interruption and restart of running process steps) or time-constraint patterns [57] (e.g. a time window during which a step can be executed). In contrast to them, vBPMN does not define an additional notation and semantics for the realization of these patterns within the process before or during execution. Instead, vBPMN relies on the process modeling language itself to specify the adaptation behavior. The two main advantages of this characteristic are that patterns are self-explaining and that they can be arbitrarily modified and extended. Figure 3 contains two of such patterns under the plus signs. A special characteristic of an adaptation pattern is that it always contains a placeholder for the underlying adaptive segment which it is applied to. By these means, multiple patterns can be conveniently nested and combined. For instance, the first pattern (*Timed Message*) in Figure 3 sends a notification after a specified time while the adaptive segment is running. The second pattern (*Insert Parallel*) corresponds to the basic parallel-insert pattern of the additional process step "Send Advertisement".

3.2.3. Adaptation Rules

For variant construction, the connection between the values of data-context variables and process tailoring operations needs to be established. This is achieved by formulating adaptation rules in an event-condition-action (ECA) format. The event

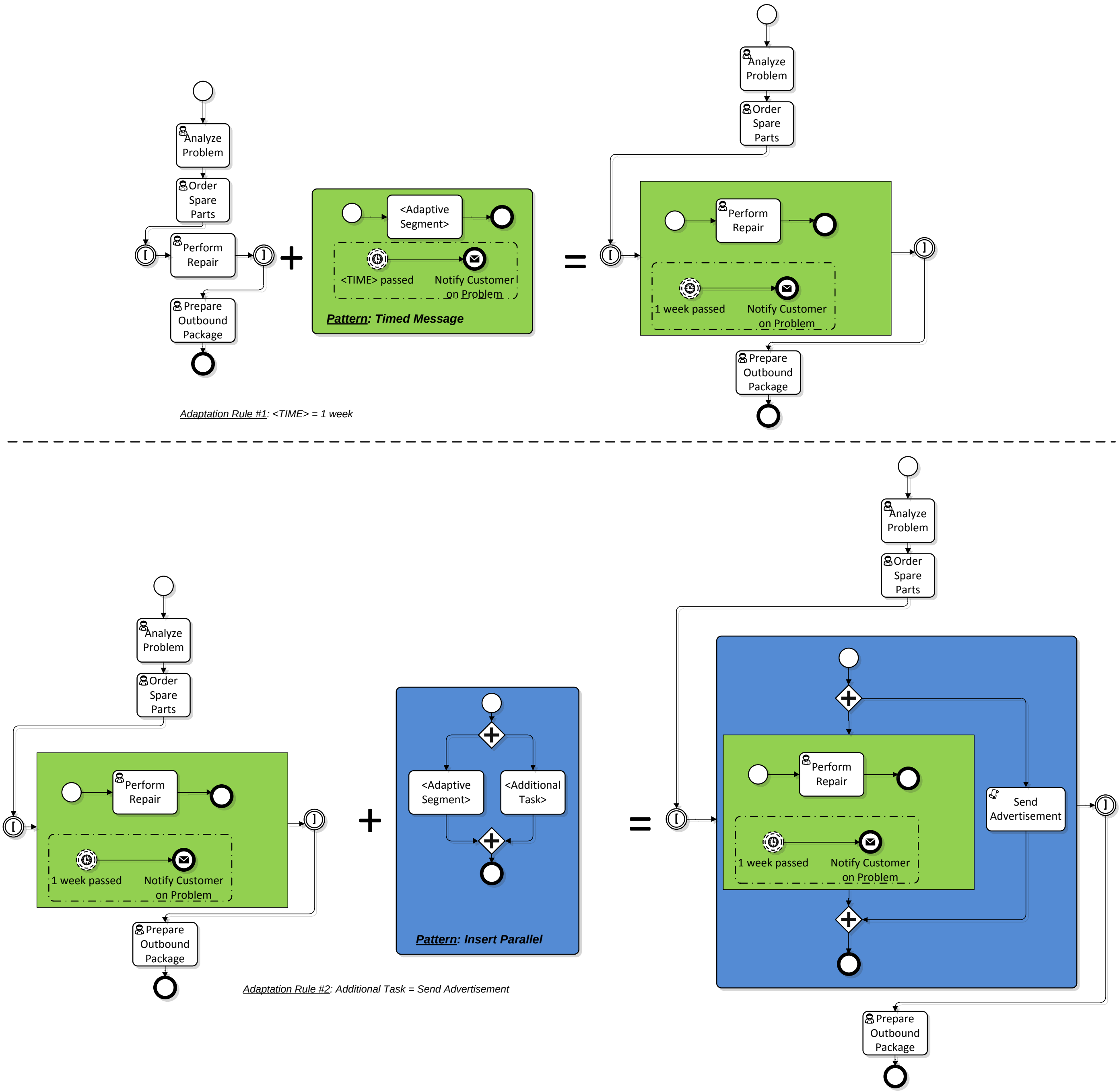


Figure 3: Pattern-Based Adaptation of a Process Model in vBPMN

of such an ECA rule corresponds to the entry event of an adaptive segment. The conditions constitute value restrictions on context variables. Finally, the actions contain parameterized adaptation patterns from the catalogue. An abstract syntax, where * stands for 0-n repetitions, can be defined as follows:

ON *entry-event*
IF *<data-context>*
THEN APPLY [*<pattern((parameter = value)*)>*]*

Each time a token within a process instance enters an adaptive segment, the context variables are evaluated and the segment potentially becomes subject to immediate adaptations before continuing through the segment. Below, the two adaptation rules for the repair process example are provided in a textual form. In Figure 3, it is shown how the corresponding two parameterized patterns are applied to the adaptive segment by wrapping them around it as extensions:

```
Adaptation Rule #1:
ON performRepair_entry
IF country=USA AND customerStatus=high
THEN APPLY timedMessage(time=1 Week)

Adaptation Rule #2:
ON performRepair_entry
IF country=USA
THEN APPLY insert_parallel(additionalTask='SendAdvertisement')
```

For our user study, we will not take into account adaptation rules with complex composite context conditions. This means that each adaptation rule has only one context factor, which uniquely assigns the rule to a distinct process variant.

4. Description of Scenarios and Variant Model Realizations

In order to keep our work comparable to the existing publications on state-of-the-art process configuration, we chose the travel process presented in [15] and the unborn child registration process as presented in [58] as the two main objects for the user study. These processes have already been realized in C-YAWL as a reference case study to demonstrate the overall approach, which facilities the establishment of a comparative experiment. To keep the complexity of our user study on a manageable level, we only examine the maintenance performance on process variants which depend on one single context factor, namely the institution or organizational unit which runs the particular process variant. While we only summarize the most relevant aspects of the realization of the two scenarios in C-YAWL and vBPMN, the full set of models for the user study can be found in the Appendix of this paper.

4.1. A Simple Scenario: Travel Process Variants

The travel process mainly consists of four phases: receiving an order, conducting different bookings, payment, and document delivery. There are two variants of the travel process, one for online booking and one for booking at an on-site travel agency. They differ, for example, w.r.t. whether a reduction card is booked, whether the booking can be canceled, whether cash payment is possible, and whether it is possible to collect the documents personally.

Realization in C-YAWL. Figure 4 shows the realization of the travel process from [15] within the C-YAWL editor[1]. By its nature, the C-YAWL reference model contains the superset of variants, i.e. all flow elements which can occur in any of the variants. We changed the original reference model to replace OR-JOINs with behaviorally equivalent constructs using parallel and exclusive gateways. We avoided OR-JOINs in the experiment, since this construct has complex non-local semantics and it is advised to design understandable process models free of OR-JOINs [59]. Against this background, we did not expect participants to be familiar with it, while most of them were aware of parallel and exclusive branching.

Figure 4 also shows how the incoming port for the "Cancel" step is set to *blocked* for the online variant of the travel process. The remaining port configurations for other tasks within the model are not immediately visible in the C-YAWL tool and need to be shown on request per element.

Realization using vBPMN. The vBPMN realization of the same travel process variant for online booking is shown in Figure 5. The main tab in the center of the vBPMN editor shows the reference process. Note that this model is smaller than the C-YAWL model in terms of the number of nodes and sequence flows. The vBPMN model is not the superset of variants, but has been constructed according to the minimum edit distance. In the lower left of Figure 5, the set of adaptation rules per variant is shown. In the lower right, the rules can be independently activated or deactivated. Figure 5 shows how a SKIP pattern, i.e the bypassing of the original adaptive segment, is applied on the "Book Reduction Card" task to realize the online booking variant. The correspondence of adaptations to structural modifications in the graph is visually highlighted by a matching color, which is especially helpful if the effect of multiple adaptation applications at once needs to be understood by the modeler. The adaptation patterns can be managed separately by a corresponding tab within the same editor. To create an adaptation rule, the user drags and drops an item from the list of adaptation patterns (not shown in the figure) onto an opening node for and adaptive segment within the reference process. Potentially required additional parameters like a task which should be inserted in parallel can then be specified in a popup window.

4.2. A Complex Scenario: Municipality Process Variants

As described in [58, 60], large parts of processes in public administration are driven by legislation. In many countries, there are laws prescribing how a name can be determined and registered for a newborn child. In the Netherlands, a name can even be registered for a child yet to be born. For such procedures, the NVVB (Nederlandse Vereniging Voor Burgerzaken) provides reference process models[2], describing the "best-practice"

[1] Version 2.2, obtained in March 2012 from http://www.processconfiguration.com/download.html.

[2] http://www.nvvb.nl/producten-en-projecten/werkprocessen

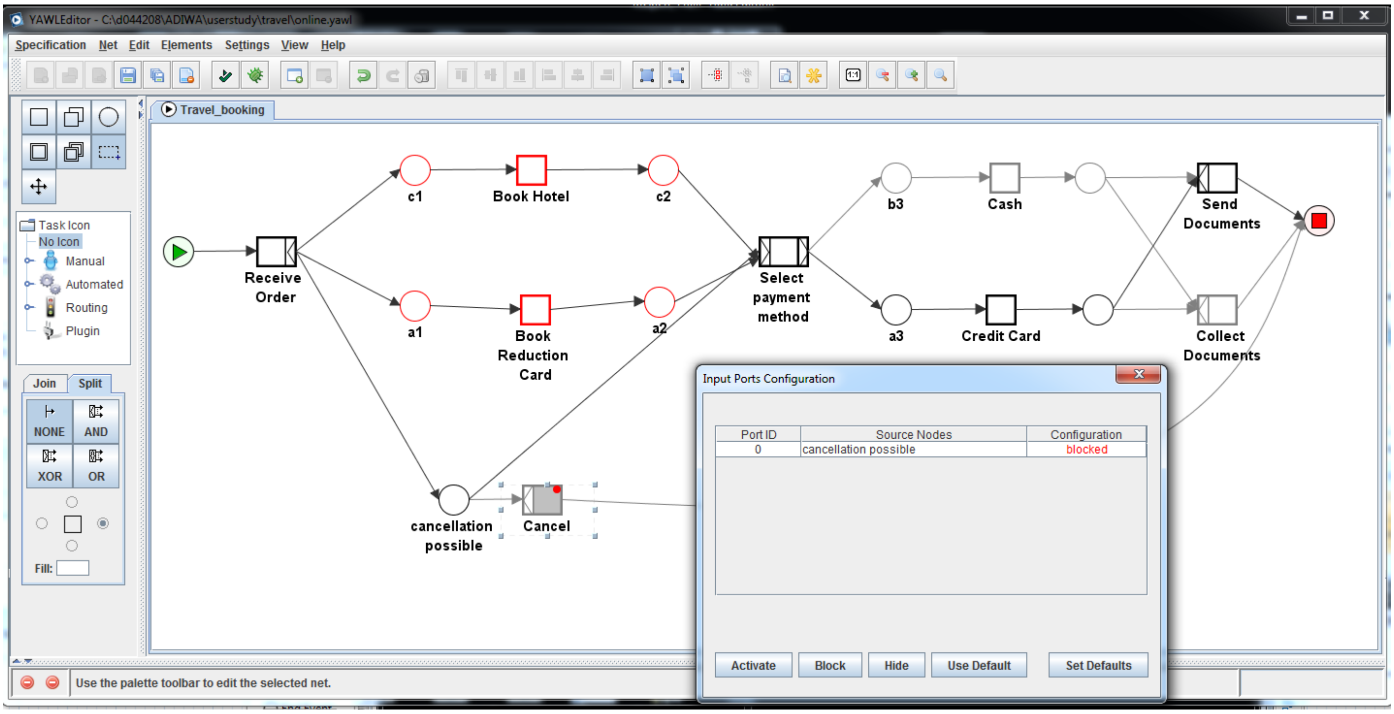


Figure 4: Travel Reference Process with Configuration for Online Booking in the C-YAWL Editor

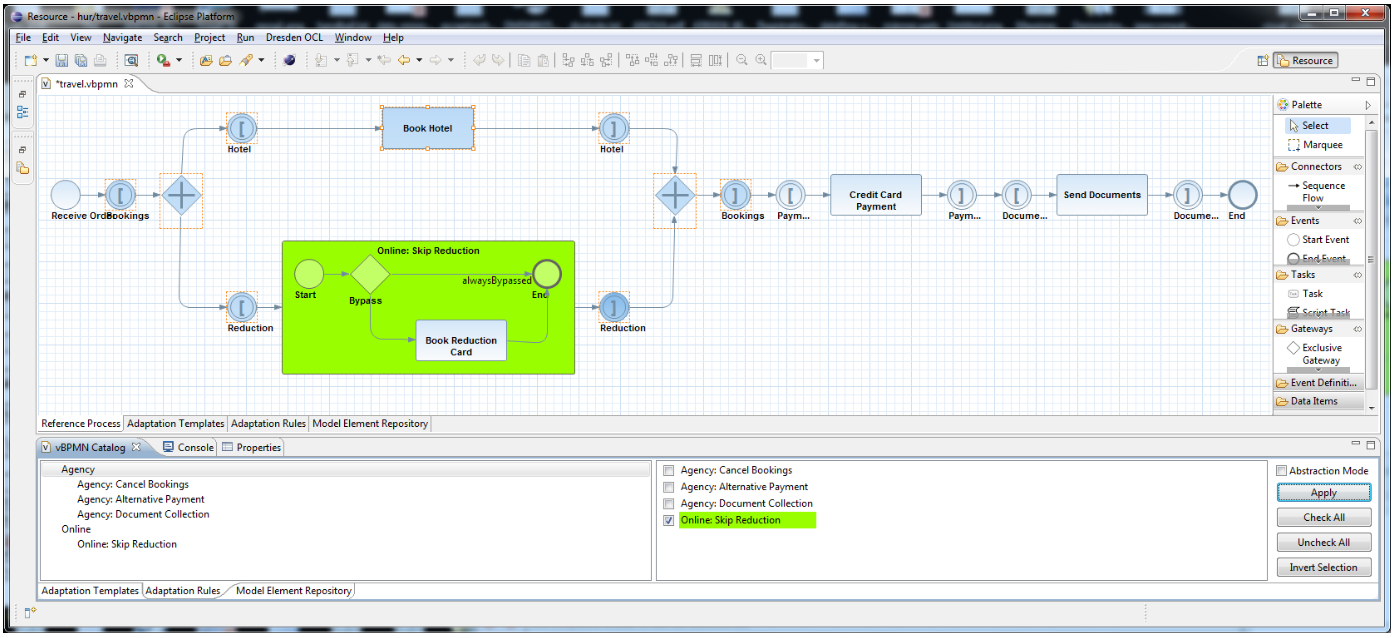


Figure 5: Travel Reference Process with Adaptaion for Onlibne Booking in the vBPMN Editor

for how the procedure should be executed. Despite the extensive national regulations which have to be considered for the registration process, the procedures grant some freedom to local municipalities for the concrete embodiment of the process. Figure 6 shows four different variants of the unborn child registration process, each defined by a different Dutch municipality. The processes are shown in their original form as reported in [58] using the Protos process modeling language [61]. It is not necessary at this point to fully understand the notational elements or the complete business logic of the variants. However, one can see that all of the variants contain process steps related to checking whether the requester is authorized to perform the name choice. These are executed in different phases of the variants. The concerned steps are encircled in Figure 6. Other steps, like a nationality-specific name choice, occur only in some variants. Some of the steps, like handing over the final copy of the registration document, are located in a fixed position for all variants.

When analyzing the models on a structural level, it is possible to identify nested behavioral blocks as shown in Figure 7. The four variants are displayed from bottom to top, the process flow is displayed from left to right. For example, within the "First Child Check" block, it is determined whether there is a previous child for which a last name has already been chosen. In that case the last name is also assigned to the unborn child after it has been ensured that at least 1 of the parents has appeared in person at the municipality. If it is indeed the first child of the two parents, one proceeds with the variant-specific name choice procedure. This procedure in turn is for instance succeeded by an optional determination of nationality in municipality 2, while nationality is not taken into account in municipality 1 at all. Correspondingly, the process variants for the municipality could also be constructed by separating them into six phases (from identity determination to the handover of documents) and combining (i.e. nesting) different variant aspects at distinct phases of the process. Especially phases two and five consist of variable nested components.

For simplification purposes of our user study, we only take three of the four variants from Figure 6 into account. In particular, we exclude the simplest variant, shown on the lower right of Figure 6.

Realization using C-YAWL. In the following, we denote by $X \rightarrow Y$ a sequence flow from process node X to process node Y, where $X = Y$ may be possible. In order to construct a configurable process model for the child registration variants in C-YAWL, the authors of [58] created a single process model per municipality and superimposed them to obtain a resulting aggregated model. That means, for example, that if a sequence flow $A \rightarrow B$ exists only for municipality 1 and a sequence flow $A \rightarrow C$ exists only for municipality 2, the final C-YAWL model would contain an XOR split after step A with sequence flows to both B and C. The port configurations would be set correspondingly, which means $A \rightarrow B$ would be blocked for municipality 2 and $A \rightarrow C$ would be blocked for municipality 1. This approach is clearly single element/node oriented, which means that there is no exploitation of higher-level structural patterns within the process. Variability is de-facto being modeled node-by-node. In Figure 8, the resulting C-YAWL model that we also employed for our user study is depicted. Please note that this is a conceptual view on the overall C-YAWL model provided by the authors of this paper; the model is not visualized this way in the C-YAWL editor. Municipality-specific blocked port configurations are visually indicated by a "stop-sign", the other ports are generally enabled. Interestingly, the authors of [58] also examined other process variants within a municipality case study and point out that "*the process of acknowledging an unborn child is the **simplest**, i.e. the [...] other combined process models include both more steps and more arcs*".

Realization using vBPMN. For building a reference process model in vBPMN together with pattern-based adaptation rules for variant construction, we take into account that the registration process variants can be constructed from nested building blocks as already shown in Figure 7, p. 12. The general observation that there are particular phases and *patterns* in the behavioral variability of the child registration process has already been recognized by A. Hallerbach in her dissertation on Provop [62]. Accordingly as illustrated in Figure 9, we can identify five vBPMN patterns which are reused at least two times, plus the SKIP pattern. Please note that this is a conceptual view on the overall vBPMN model provided by the authors of this paper; the model is not visualized this way in the vBPMN editor. The dashed links indicate for which variant(s) a pattern is applied and if necessary in which order, contained within rounded brackets.
One can see that compared to the C-YAWL model, there is an additional layer of indirection for process variant construction introduced by the adaptation rules and patterns. This additional layer serves to decrease the degree of interlinkage between distinct nodes and modularizes the process into coherent parts. The question which remains to be examined is whether this trade-off is beneficial for process variant maintenance.

5. Comparative User Study

In the following, we first explain the setup of the conducted user study on C-YAWL and vBPMN. Then the quantitative results are presented. Next, we provide an interpretation of the results in order to derive recommendations for process variant modeling in general and for the two approaches in particular. Finally, we briefly discuss the limitations of this study and its findings.

5.1. Setup of the Study

For the design of our experiment, we have been following the practices recommended by [63]. Before finalizing the setup of the user study, we conducted two test runs with students, who did not take part in the final evaluation, and iteratively improved the setup. Changes to the setup, for instance, consisted of the removal or simplification of over-complicated tasks and the adaptation of explanatory text paragraphs which contained ambiguities. The final setup of the user study is as follows:

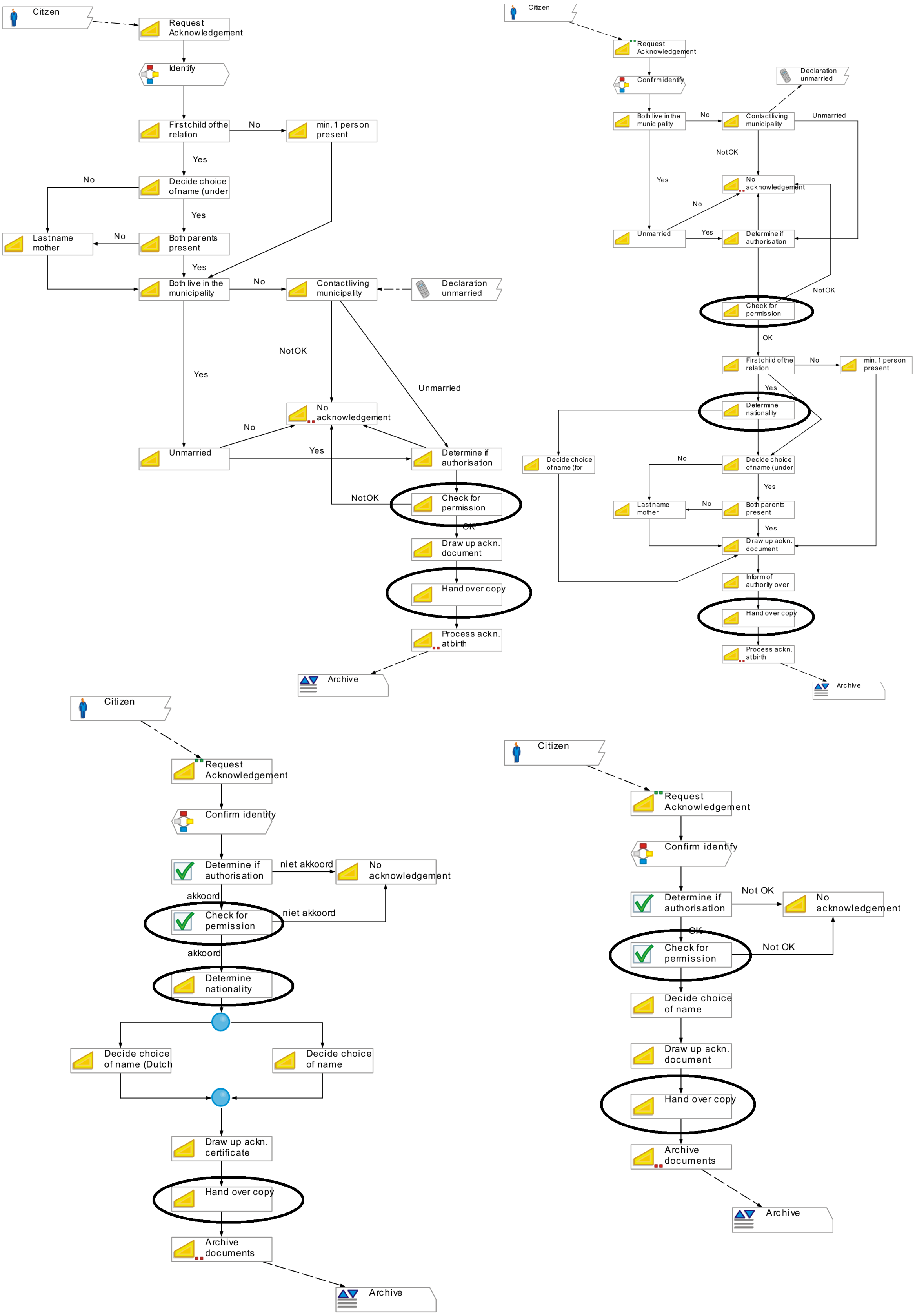


Figure 6: Registration Processes for an Unborn Child in 4 Dutch Municipalities [58]

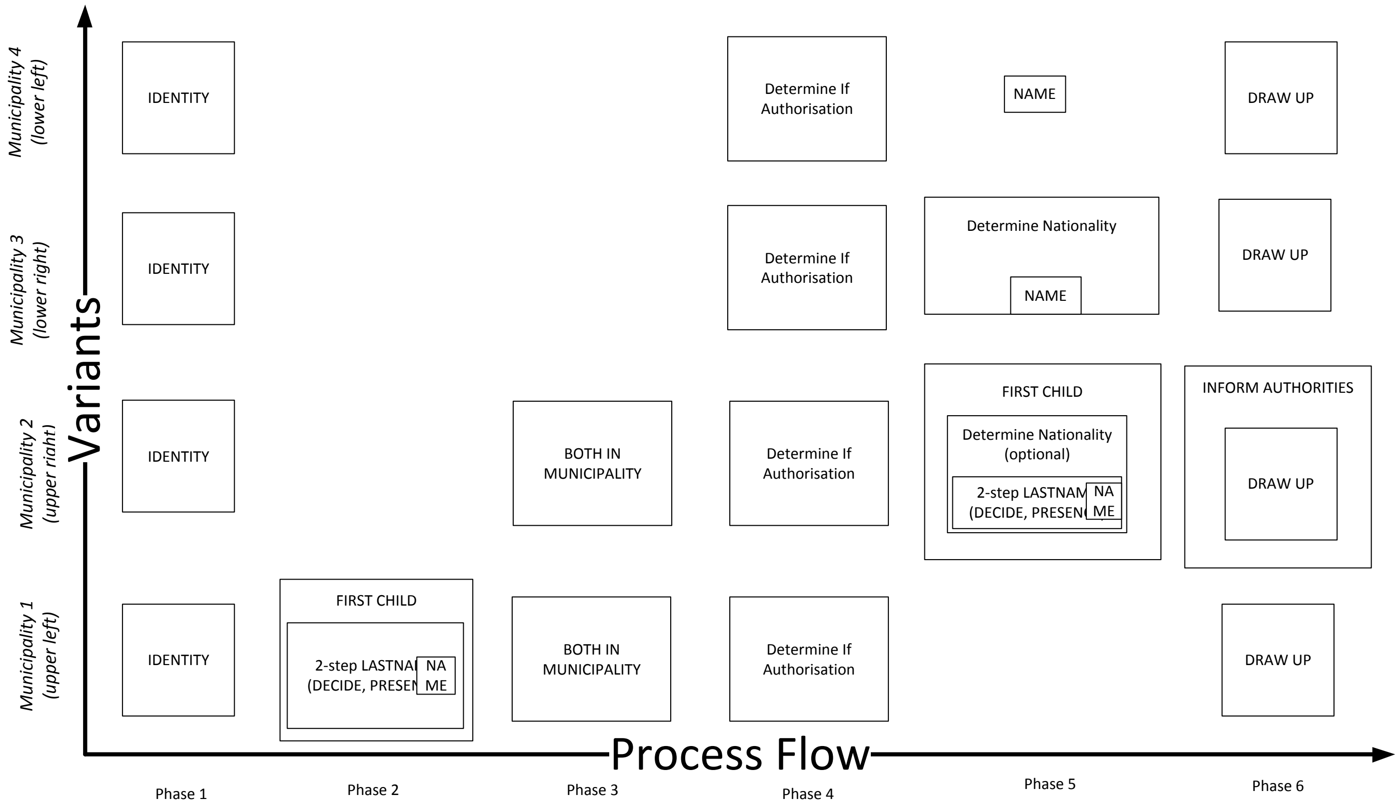


Figure 7: Nested Block Structure of the Unborn Child Registration Process Variants

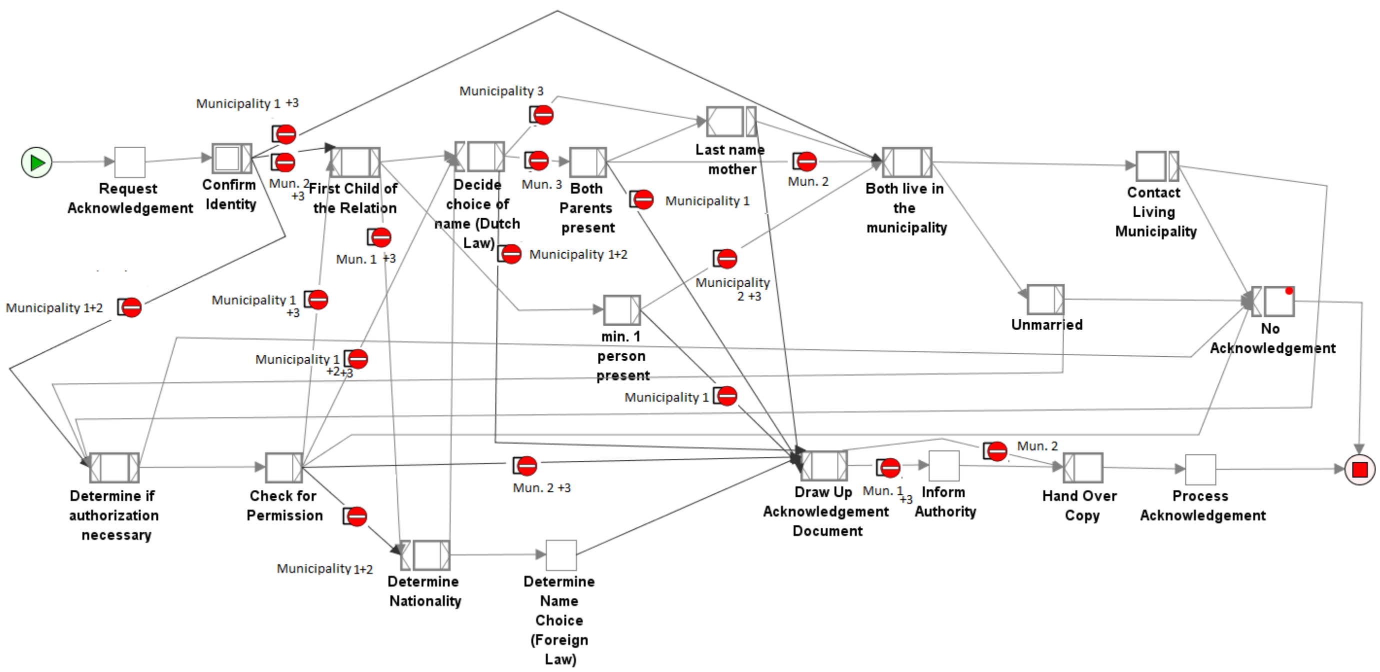


Figure 8: Aggregated Conceptual Visualization of Municipality Process Variants in C-YAWL

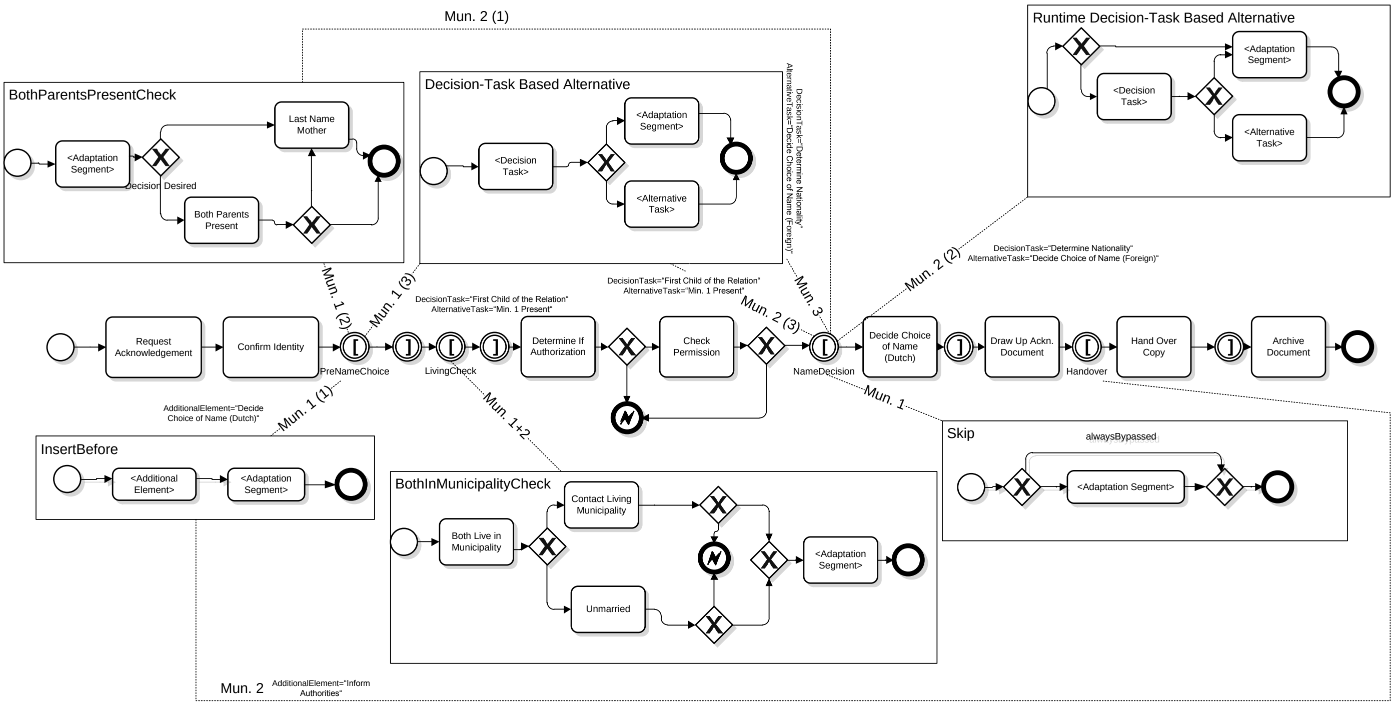


Figure 9: Aggregated Conceptual Visualization of Municipality Process Variants in vBPMN

Subjects. A number of 14 participants were involved in the execution of the experiment. Some descriptive statistics on them are provided in Table 2. Note that one participant stated he has examined "many" process models, but indicated 0 years of modeling experience, which explains the deviant minimum values. For our experiment, it was important to gather knowledgeable people with at least some process modeling skills. The group of participants therefore was composed of BPM researchers, BPM software developers, and BPM consultants from SAP, Software AG and the German Research Center for Artificial Intelligence (DFKI). Besides experienced business process analysts and developers, we also included students in the experiment who already had at least some basic experience in process modeling. As the major incentives, the participants were told to get the chance to learn about advanced concepts and tools for process variant modeling. Moreover, they were allowed on demand to compare their results against others after they had completed the study. Participants who were unfamiliar with process modeling were not considered in order to prevent confusion and frustration with the partly challenging tasks during the experiment, which would have biased or invalidated the results.

Objects. The simple travel process variant model and the complex child registration process variant model realized in vBPMN and C-YAWL each represented the objects of the experiment. In other words, two process models in each language were designed.

Tasks. We designed a number of variant-specific maintenance tasks, including model understanding as well as model modification tasks which were to be conducted by all participants. The different types of understanding tasks can be derived by taking the prior work of [64] as a reference. The work contains comprehension questions of four classes (*order*, *concurrency*, *repetition*, *exclusiveness*) to empirically examine and measure human comprehension of process models. Matching the question classes against the selected travel and municipality process model yields that they do not contain any repetition (or loop) behavior. For each of the other three classes, we included at least one question.
The different types of modification tasks can be derived by taking the prior work of [8] as a reference. The work lists typical patterns of tailoring control-flow to changed business requirements, whereas the most basic and first four patterns comprise the insertion, deletion, movement, and replacement of process fragments. For each of these four patterns, a modification task was included in our experimental setup.

An example of an understanding task for the child registration variant model is provided below. The complete set of tasks is contained in the Appendix of this paper.

```
FOR MUNICIPALITY 1, ARE ''LAST NAME MOTHER'' AND
''MIN 1 PRESENT'' MUTUALLY EXCLUSIVE, I.E. THEY
CANNOT BOTH BE EXECUTED WITHIN A PROCESS?

[ ] yes
[ ] no
```

Factors and Factor Levels. We consider four binary factors for our experiment setup. *ProfessionalLevel* captures the seniority level of the participants according to Table 2. We distinguish senior-level and student-level participants. Senior-level participants include post-docs and industry employees. Student-level participants include students up to PhD candidates. *IsModeling*

Table 2: Descriptive Statistics for Participants of the Experiment

	Age	Academic Degree (1=Bachelor; 2=Master; 3=PhD)	Process Modeling Experience in Years	Process Models Read	Process Models Modified	Process Models Modified	Average Number of Nodes in Modeled Processes	Professional Level (0=Student; 1=Senior)
min	24.0	1.00	0.0	5	0	1	10.0	0
max	51.0	3.00	15.0	200	100	100	50.0	1
median	32.0	2.00	2.5	20	10	10	19.0	0.5
mean	33.6	2.21	3.9	49	22	25	20.0	0.5
std.dev	7.6	0.68	4.0	54	34	32	9.9	0.5

indicates whether the task is an understanding or a modification task. *IsComplex* captures whether the simple travel or the complex municipality process has been used. *ExecutionTool* indicates whether C-YAWL or vBPMN has been used.

Tool Support. One can model business processes with pen and paper or leverage the support of a specialized software application. Within this study we followed the latter approach and prescribed the participants to use software tools for business process modeling. A possible negative aspect of this design decision is that tool properties, like usability and functional features, may impact the experiment outcome. To avoid such effects we instructed the participants to use only the comparable features of the tools. Furthermore, a decisive aspect for this choice is the high complexity of the tasks to be executed by the participants: the proposed model variance management tasks are hardly manageable with the pen and paper approach.

Response Variables. For each task, after the participant confirmed that he completely understood the question, we measured the time in seconds (*sValue*) until the participant indicated that he had finished or given up.

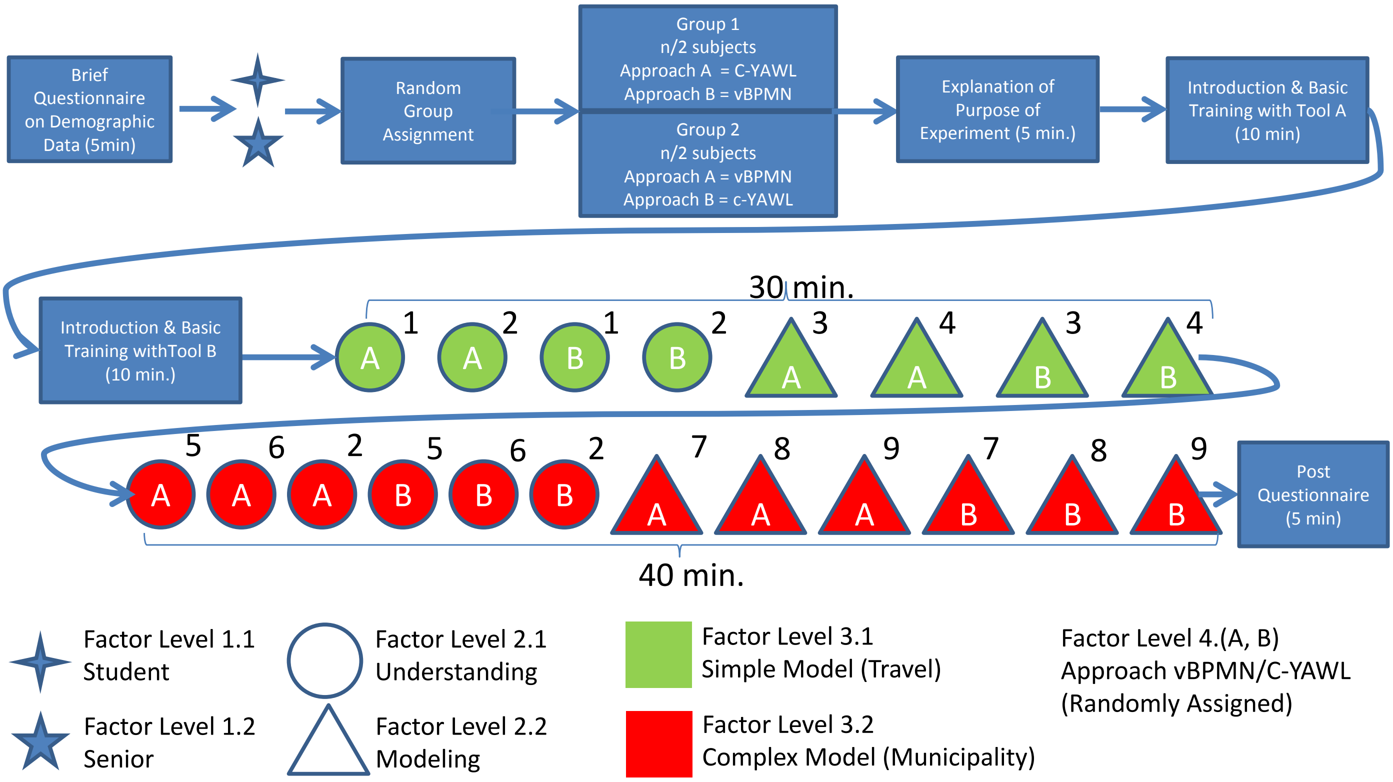
We also judged whether the task was correctly processed by the participant or not, indicated by the *success* variable. Errors, for instance, resulting from tool bugs or typos were not considered as errors and the participant was correspondingly advised to correct it. After all, we are more interested in evaluating the conceptual differences between C-YAWL and vBPMN. For each question, we also asked the participant to rank the *ConceptConvenience* and *easiness* for the particular task as formulated below. To understand the difference between the two response variables, consider the fact that even if a participant has understood a concept it may be hard for him to put it into practice, or vice versa. Examples for the two question types are provided below:

```
How confident did you feel when processing this task, i.e.\
was it clear to you how to achieve the task?

Not at all -> 1[ ]  2[ ]  3[ ]   4[ ]   5[ ] <- absolutely


How easy was the processing of this task for you, i.e.\
could you easily and quickly realize what you wanted to achieve?

Not at all -> 1[ ]  2[ ]  3[ ]   4[ ]   5[ ] <- absolutely
```

Experimental Design. The experiment design is illustrated in Figure 10, while the independent and dependent variables which are captured are additionally summarized in Table 3. At first, participants had to answer a brief questionnaire to gather demographic data and data related to their experience in process modeling. Then, they were randomly assigned to a group, which determined the task processing tool (A/B) as C-YAWL or vBPMN. After a general introduction to the purpose and scope of the experiment, an introduction and a neutral training task was provided for approach A. After the participant has confirmed a basic understanding of approach A, the same procedure was repeated for approach B. Nobody was instructed on the origin of the approaches and the instruction was conducted in a neutral tone.
Having completed the introductory part for both approaches, the participants proceeded with the execution of the actual maintainability evaluation tasks. Altogether, there were 20 tasks; 8 tasks were conducted on the simple model and 12 tasks on the complex model. Tasks were paired by a particular task type as shown in the legend on the bottom of Figure 10. Correspondingly, each task was executed with one of the tools exactly once by each participant; the tasks were completely *balanced* w.r.t *isModeling* and *executionTool*. We chose the alternating order of tool assignments to tasks as a compromise between learning effects and "first seen tool bias". For each task, we determined the response variables *sValue*, *success*, *Concept Convenience* and *easiness* as described above. In addition, we allowed the participants to provide qualitative feedback after the accomplishment of each evaluation task.
After all evaluation tasks were completed, the participants had to process an ex-post survey. This survey contained six statements on aspects of the respective process variant management approach in terms of modularization support, model understanding, model manipulation, subjective perception of the approach, practical value of the approach, and usability of the tool implementation. The participants had to rank the applicability of the statements on C-YAWL and vBPMN each on a scale from 1-5. The exact verbalization of all questions can be found in the Appendix.

5.2. Analysis

Since 14 users conducted 20 tasks, we have a number of 280 measurement points in our data set. All statistics described in the following have been computed using the R language and



1: Check whether task can be executed for given variant
2: Check in which variants (multiple) a task can be executed
3: Skip a task A and add a task B for a given variant
4: Create a modified variant from an existing one:

5: Check for mutual exclusion of two tasks in all variants
6: Check for task order of A and B in multiple variants
7: Remove a task from all variants.
8: Insert a new variant-specific task.
9: Re-route an existing task within a variant.

Figure 10: Experiment Design for Comparing Process Variant Modeling

Table 3: Summary of Independent and Dependent Variables captured per Task

Factors (Independent Variables)	Responses (Dependent Variable)
Professional Level {*Senior, Expert*}	Success (*boolean*)
Is Modeling {*Understanding, Modeling*}	Concept Convenience (*1-5 scale*)
Is Complex {*Simple, Complex*}	Easiness (*1-5 scale*)
Execution Tool {*C-YAWL, vBPMN*}	Time sValue (*seconds*)

environment for statistical computing [65]. In order to determine which type of statistical test we had to apply for each response variable, we first set out to establish the distribution of the response variables. The Kolmogorov-Smirnov test was used to check how well their fit was with the normality distribution [66], except for *success*. After all, a binary response variable cannot be expected to be normally distributed [67]. It turned out that a normal distributions could also not be assumed for *sValue*, *Concept Convenience* or *Easiness*. For *sValue*, we checked the log-transformed values on normality. Indeed, since *log10(sValue)* can be assumed to be normally distributed, it is appropriate to use this variable in the further statistical analysis[3] [68, 69]. While a Levene test is commonly needed to check for equal variances [70], it does not need to be performed here since our experiment is almost completely balanced [71, p. 382].

Given the lack of normality of three out of four response variables, the popular and common analysis of variance (ANOVA) [66] was not considered for these. Only for *sValue*, it was apparent that a multi-way repeated measures ANOVA would be suitable. For each combination of the other three response variables with each of the four dependent variables, we compute an averaged score per subject. We re-checked the normality distribution assumption for each of the resulting groups. This time, we used the Shapiro-Wilk test instead of the Kolmogorov-Smirnov test, because for small sample sizes (< 50) it is considered more appropriate [72, p. 84], [73, p. 147]. Following [74], if a normal distribution can be assumed, we used the parametric t-test to check whether there are significant differences between the groups of subject-averaged values. We used a paired test for within-subject independent variables and an unpaired test for between-subject independent variables.
If a normal distribution cannot be assumed, we used the non-parametric paired Wilcoxon test for within-subject independent variables and the non-parametric unpaired Mann-Whitney test for between-subject independent variables.
On the p-values of the three simultaneous independent tests, we applied a Bonferroni correction. This is a conservative method used to counteract the problem of multiple comparisons [73, p. 144]. We will only report on the corrected values in the following.

The outcomes of our statistical analysis are displayed in Table 4. As can be seen, the p-values for the *success* rate as well as the *easiness* depending on the *ExecutionTool* are significant for an alpha level of 0.05. The same holds for *Concept Convenience* and *easiness* depending on the *task type*. The alpha level of 0.05 means that we assume a 95% confidence level to call results statistically significant. All other p-values are not significant. Boxplots for the twelve constellations are provided in the Appendix of this paper.

The results of the performed ANOVA on the log-transformed processing time in seconds log10(sValue) on all 280 data points are provided in Table 5. The factor *ProfessionalLevel* does not have any significant impact. Each of the factors *IsComplex*, *IsModeling*, *ExecutionTool*, however, shows a significant impact on the processing time. Furthermore, the ANOVA allows to discover interactions between dependent variables. We can see that there is an interaction between the *ExecutionTool* and *IsModeling*, as well as between *ExecutionTool*, *IsModeling* and *ProfessionalLevel*. This effect is illustrated in Figure 11.

5.3. Discussion

5.3.1. Interpretation of Quantitative Results

The support for our propositions based on the statistical results are summarized in Table 6. The results are discussed in the following.

Table 6: Support for Propositions based on Statistical Results

Proposition	Supporting Dependencies	Statistical Support
P1	Success~IsComplex	No
P2	sValue~IsComplex	Yes
P3	ConceptConvenience~IsComplex Easiness IsComplex	No
P4	Success~ProfessionalLevel	No
P5	sValue~ProfessionalLevel	No
P6	ConceptConvenience~ProfessionalLevel Easiness~ProfessionalLevel	No
P7	Success~ExecutionTool	Yes
P8	sValue~ExecutionTool	Yes
P9	ConceptConvenience~ExecutionTool Easiness~ExecutionTool	Partial

The results we obtained from the statistical tests for the response variables *Success*, *Concept Convenience* and *Easiness* can be interpreted as follows:

- **Success ~ IsComplex (Proposition P1)**: Contradictory to our expectations, the participants did not *per se* run into more errors when dealing with variant management tasks on the complex model compared to the simple model. This result may need a deeper investigation as future work. Note that we did not set an upper boundary regarding how much time a user may consume for processing a task. In case of a time threshold, it seems plausible that there is a chance for this response variable to differ more intensely depending on the complexity level.
- **Success ~ IsModeling**: Interestingly, the participants did not produce significantly more errors for modeling tasks than for understanding tasks. In analogy to *Success ~ IsComplex*, this finding should be further investigated by an experimental setup with restricted processing times.
- **Success/ConceptConvenience/Easiness ~ ProfessionalLevel (Propositions P4 and P6)**: We could not find

[3] See Appendix of this paper for a graphical illustration of the approximated normal distribution for the histogram of log-transformed second-values.

Table 4: Averaged Response Variables and Pairwise Significance of Group Difference

Dependent Variable	Success								Concept Convenience								Easiness							
Factor	Professional Level		IsModelling		IsComplex		**Execution Tool**		Professional Level		**IsModelling**		IsComplex		Execution Tool		Professional Level		**IsModelling**		IsComplex		**Execution Tool**	
Factor Level	Student	Senior	Understanding	Modeling	Simple	Complex	C-YAWL	vBPMN	Student	Senior	Understanding	Modeling	Simple	Complex	C-YAWL	vBPMN	Student	Senior	Understanding	Modeling	Simple	Complex	C-YAWL	vBPMN
Average	0.850	0.829	0.907	0.771	0.857	0.827	0.757	0.921	4.093	4.250	4.379	3.964	4.080	4.232	3.971	4.371	3.650	3.864	4.129	3.386	3.786	3.738	3.171	4.343
Statistical Test	Mann-Whitney		Wilcoxon		Wilcoxon		**Wilcoxon**		Mann-Whitney		**Wilcoxon**		Paired T		Paired T		Unpaired T		**Paired T**		Paired T		**Wilcoxon**	
p-value after Bonferroni	1.0000		0.2176		1.0000		**0.0480**		1.0000		**0.0444**		1.0000		0.3032		1.0000		**0.0003**		1.0000		**0.0252**	

Table 5: ANOVA Results for Processing Time log10(sValue)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ProfessionalLevel	1	0.24	0.24	0.89	0.3645
Residuals	12	3.28	0.27		
ExecutionTool	1	1.31	1.31	19.94	**0.0008**
ExecutionTool:ProfessionalLevel	1	0.13	0.13	2.01	0.1817
Residuals	12	0.79	0.07		
IsModelinng	1	7.75	7.75	249.14	**0.0000**
IsModelinng:ProfessionalLevel	1	0.11	0.11	3.65	0.0802
Residuals	12	0.37	0.03		
IsComplex	1	1.60	1.60	29.97	**0.0001**
IsComplex:ProfessionalLevel	1	0.00	0.00	0.00	0.9486
Residuals	12	0.64	0.05		
ExecutionTool:IsModelinng	1	0.27	0.27	5.37	**0.0390**
ExecutionTool:IsModelinng:ProfessionalLevel	1	0.26	0.26	5.27	**0.0405**
Residuals	12	0.60	0.05		
ExecutionTool:IsComplex	1	0.06	0.06	2.05	0.1776
ExecutionTool:IsComplex:ProfessionalLevel	1	0.02	0.02	0.60	0.4540
Residuals	12	0.38	0.03		
IsModelinng:IsComplex	1	0.06	0.06	1.44	0.2537
IsModelinng:IsComplex:ProfessionalLevel	1	0.02	0.02	0.47	0.5040
Residuals	12	0.47	0.04		
ExecutionTool:IsModelinng:IsComplex	1	0.08	0.08	1.62	0.2273
ExecutionTool:IsModelinng:IsComplex:ProfessionalLevel	1	0.02	0.02	0.35	0.5661
Residuals	12	0.57	0.05		
Residuals	168	6.60	0.04		

statistical evidence that the professional level of a participants impacts any of the response variable. This may be due to the fact that we capture this factor on a relatively coarse-granular level within our statistical pipeline. For example, students may have obtained more training on modeling formalisms, while seniors might be able to compensate this with more practical experience in this area. It would be useful to set up an experimental design with more detailed factor levels in this respect, as for example "years of modeling experience" or "amount of training on formalisms obtained". This is subject to future work.

- **Success ~ ExecutionTool (Proposition P7)**: Supporting our proposition P7, the effectiveness of a participant on dealing with variant management tasks significantly depends on the chosen approach and tooling. As we will investigate in more detail in the remainder of this section, one main source of error for tasks processed using the C-YAWL configuration approach is arguably the complicated (re)allocation of port configurations, for example to insert a new variant-specific process step. Structural changes of the reference process are always necessary in such a case in C-YAWL, which may entail unintended side-effects on other variants. In vBPMN, such changes can to a large extent be realized using variant-specific high-level adaptations, which do not impact other variants.

- **ConceptConvenience ~ IsModeling and Easiness ~ IsModeling**: The participants generally felt more convenient with the provided modeling concepts and perceived their assignments as easier processable when they had to deal with understanding tasks. This can be explained by the increased difficulty of modeling tasks in general, which implicitly contain understanding tasks as well.

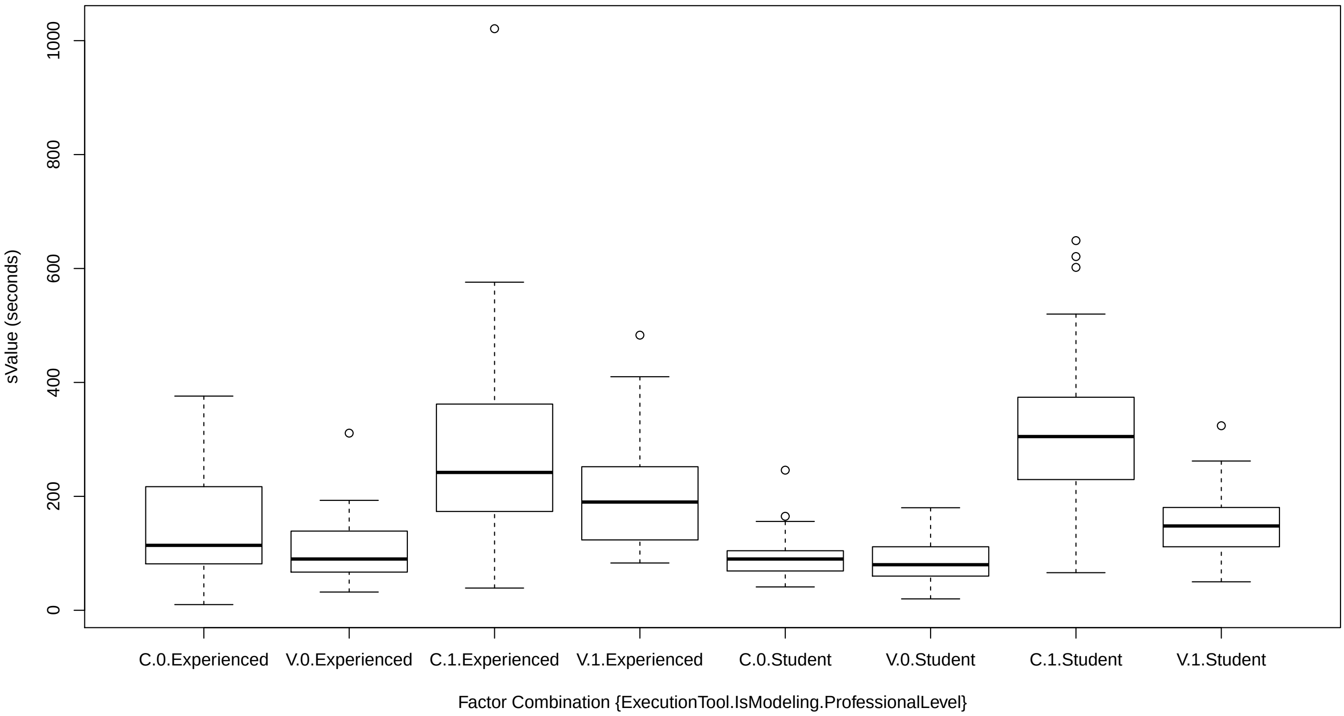


Figure 11: Boxplots for Combinations of Execution-Tool, Task Type and Professional Level against Execution Time

- **ConceptConvenience ~ IsComplex (Proposition P3)**: The non-significance of this p-value can be related to the observation that the steps for solving the variant management tasks are generally similar for simple and for complex models; they may be (much) more difficult and complicated to execute. This is, however, not captured by this response variable.

- **ConceptConvenience ~ ExecutionTool (Proposition P9)**: The non-significance of this p-value may be due to the fact that the participants generally understood both approaches and generally knew how to achieve the corresponding task using one or the other approach. This is something we could explicitly confirm during the ex-post interviews. Therefore, it is fair to conclude that both approaches appear equally challenging to learn and to understand for the participants.

- **Easiness ~ IsComplex (Proposition P3)**: Our proposition P3 seems to be rejected by the fact that the easiness of variant management task execution also does not depend on the complexity of the model. This issue also needs further investigation. After all, it may be explained by learning effects due to our experiment setup, where the participant first executed all tasks on the simple model and then all tasks on the complex model (which we deliberately did to prevent frustration of participants). It seems reasonable to assume that the error rate for participants generally drops during the experiment due to learning effects. Therefore, the error rate for tasks on the complex model is lower than it would be if the participants would not have run through the set of "simple" tasks at the start.

- **Easiness ~ ExecutionTool (Proposition P9)**: This response variable captures how *easily* a participant can put the steps for solving a specific task into practice. While this is clearly a combination of concept design and tool implementation for variant management, we can see a positive tendency towards vBPMN here. One potential explanation (also discussed later in more detail) is the more explicit definition of deviations from the "normal" process in terms of adaptation patterns and adaptation rules. As we checked in a follow-up discussion with our participants, there may be a more natural correspondence to the business requirement at hand than the relatively technical and very fine-granular C-YAWL port configurations.

The results in Table 5, which we obtained from the ANOVA on the response variable *sValue*, can be interpreted as follows:

- **sValue ~ isComplex (Proposition P2) and sValue ~ isModeling** For *IsComplex* and *IsModeling*, the significant differences in processing times make sense since we did not set a time threshold (as explained above). The participants took the time they needed to accomplish their task as accurately as possible, since it is more difficult and time-consuming to edit a model than to merely browse over it.

- **ProfessionalLevel ~ isModeling (Proposition P5)** We did not find significant differences in processing times for the two levels of professional expertise. The explanation, as already stated, could be due the the coarse granularity of the independent variable.

- **sValue ~ ExecutionTool (Proposition P8)** The results also support our proposition regarding the impact of the type of modeling approach on process variant maintenance speed by the significant p-value for *ExecutionTool*. We can even see that there is a two-way interaction between *ExecutionTool* and *IsModeling* as well as a three-way interaction between *ExecutionTool*, *IsModeling* and *ProfessionalLevel*. To explain the two-way interaction, we first examine the four "*Experienced*"-related boxplots in Figure 11 for the combinations of *ExecutionTool*(C,V) and *IsModeling*(0,1). It can be recognized that the modeling tasks in C-YAWL are especially time-consuming, as the corresponding boxplot is shifted upwards compared to the others. The same visual statement can be made about the four "*Student*"-related boxplots. An explanation may be that when extending one variant in C-YAWL, the impact on all other variants needs to be taken into account. This, for example, results in a manual synchronization of the model files and their port configurations by the participant. The three-way interaction now tells us that the general two-way interaction described above is more apparent in one of the groups (students and seniors). Visually, we can see in Figure 11 that the times required for modeling tasks performed in C-YAWL by students stand out even more from the other student tasks than this is the case for the experts. One observation we made throughout our user study was that seniors generally take more time to think about their actions, while students made their decisions earlier even though risking errors in order to continue with the next task.

In order to find out the critical types of evaluation tasks, i.e. those which are mainly responsible for the significant differences in the response variable groups, we compute for each task-type (see Figure 10) the relation of the averaged response variables as *((responseVariableVBPMN/ responseVariableCYAWL)-1)*. The result is shown in Figure 12. We can see that for tasks of type "multiexecutioncheck" (i.e. it has to be answered in which distinct variants a particular activity or situation can occur), there are no big differences between vBPMN and C-YAWL; for C-YAWL there is even a slight increase in success probability when answering such questions. For "routeexistingtask" however, the success rate for C-YAWL drops by an additional 1.75 of the rate in vBPMN (i.e. the success rate in vBPMN amounts to 275% of that in C-YAWL). Below, an exemplary task description is provided:

```
ONLY FOR MUNICIPALITY 3, 'INFORM AUTHORITY'
SHOULD BE EXECUTED DIRECTLY AFTER 'CONFIRM IDENTITY',
THEN IT SHOULD BE PROCEEDED AS BEFORE
```

While for the vBPMN model (see Figure 9) this task simply corresponds to an additional application of the INSERT pattern for municipality 3, in C-YAWL (see Figure 8) participants usually (correctly) tried to reuse the existing "inform authority task". This, however, requires the insertion of a sequence flow from "confirm identity" to "inform authority" and from "inform authority to "determine if authorization", since this is the path which the process for municipality 3 would follow. But then, the path from "confirm identity" to "determine if authorization necessary" needs to be set to blocked for municipality 3, while the two newly inserted sequence flows need to be blocked for municipality 1 and 2. Understandably, many participants found it challenging to accomplish this.

5.3.2. *Interpretation of Direct Participant Feedback*

As mentioned before, in the context of our ex-post survey within our user evaluation, we also confronted the participants with six statements. The statements target aspects of the respective process variant management approach in terms of modularization support, model understanding, model manipulation, subjective perception of the approach, practical value of the approach and usability of the tool implementation. The degree of applicability of these statements for each C-YAWL and vBPMN were to be ranked by the participants on a scale from 1 (does not apply at all) to 5 (fully applies). The results are visualized as barplots in Figure 13. One can clearly see that the average values for all statements on vBPMN are ranked at least one degree better than on C-YAWL. This suggests that participants will generally favor vBPMN over C-YAWL if they had to choose an approach for real-world process variant management.

In order to find out *why* specific tasks could be performed especially well or poorly, or which circumstances had a special impact on the subjective perception of the respective approach, we gathered qualitative feedback from each participant. This was done both after each task had been processed as well as in the context of our ex-post survey (see the Appendix of this paper). As could be expected, the provided feedback contains issues on the conceptual level as well as issues on the tooling level. The results are summarized in Table 7. The table is separated into sectors for positive and negative statements concerning vBPMN or C-YAWL each, while for each statement the number of participants (out of 14) which supported this statement is quantified in brackets.

A generally desirable feature for both approaches was a text search to quickly spot tasks within the models to be able to process a task within the experiment. We deliberately did not provide such a feature, as we thought it would have prevented participants from really mentally dealing with the distinct concepts for process variant modeling using either configuration or adaptation techniques. The risk would have been higher that participants "mindlessly" clicked around just to finish the task they are confronted with. For a professional process variant management tool, however, it seems that such a feature is crucial to achieve user satisfaction.

Improvements frequently requested by the participants consisted, for instance, of a tidier user interface or a navigable "where used list" for patterns in vBPMN, i.e. to start from a pattern and check in which adaptation rules and consequently

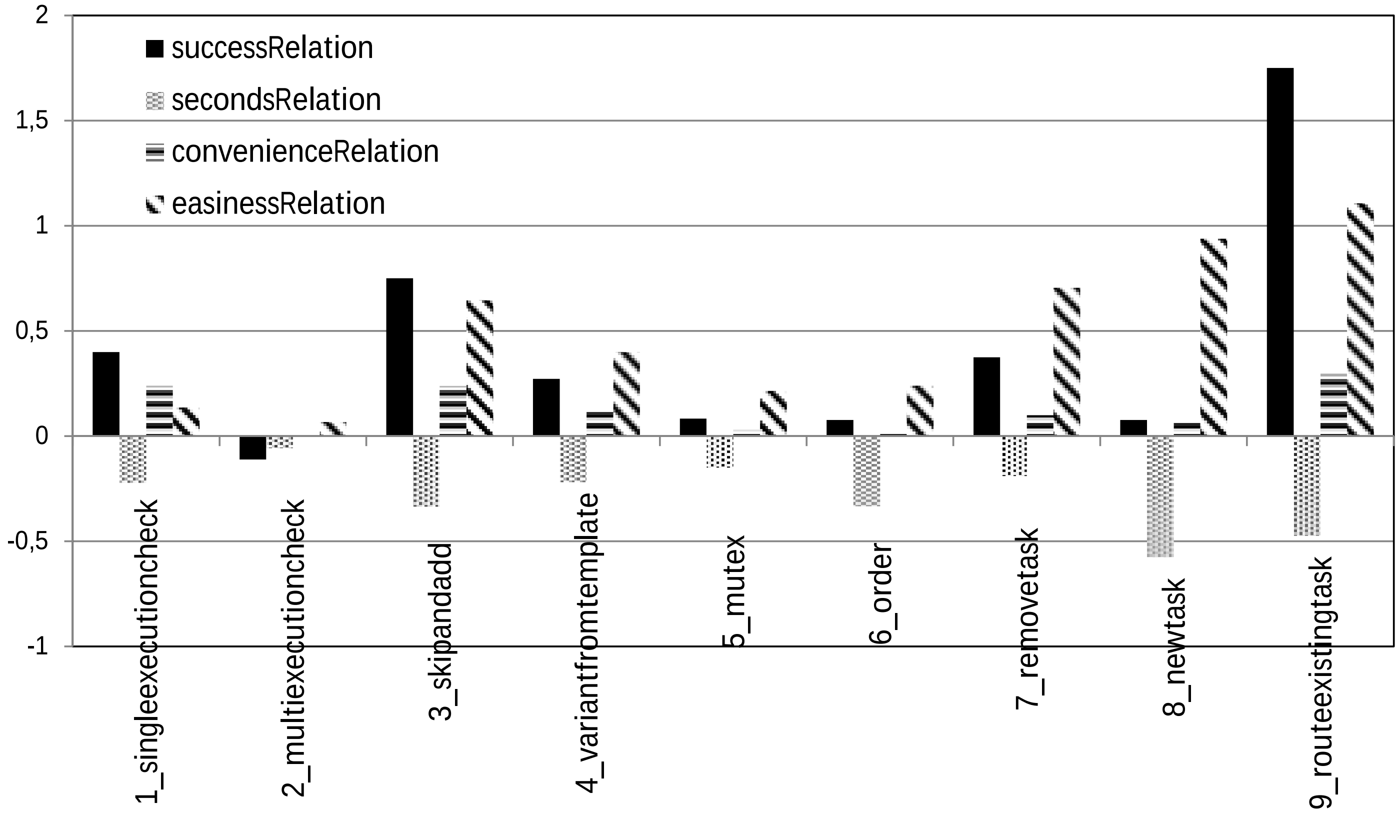


Figure 12: Relation of Averaged Response Variables for C-YAWL and vBPMN Grouped by Task Types

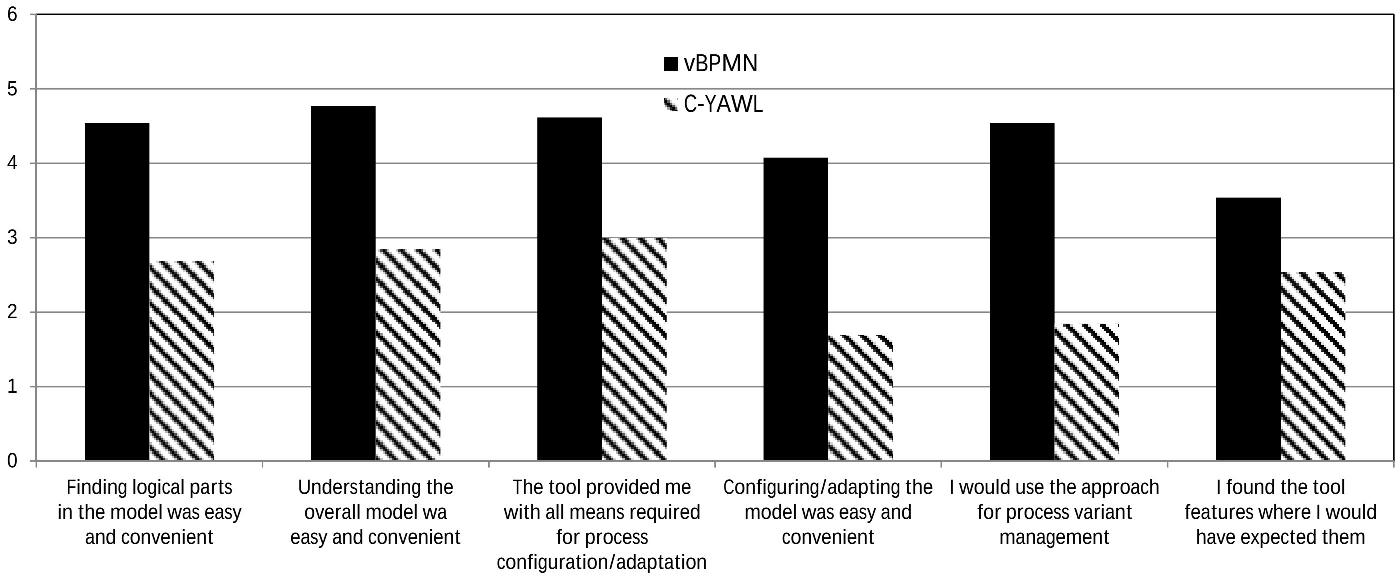


Figure 13: Averaged Results of Ex-Post Survey

on which parts of the reference process they are used. Moreover, a better visualization of the overall port configuration in a C-YAWL model was considered highly desirable. Such feature improvements for both tools seem to be straight-forwardly realizable with a reasonable amount of development effort.

Some participants requested improvements that, especially regarding C-YAWL, raise questions not only on the tool implementation, but also on the conceptual level. One of these issues concerns the integrated management and synchronization of the reference process with all of its variants. The C-YAWL editor maintains a reference process and one set of port configurations in one file. Unfortunately, it is currently not possible to administrate multiple port configurations together with the corresponding reference process within one file. This means that when structurally changing a reference process, the change needs to be manually propagated to all C-YAWL files containing the respective reference process. An approach for propagating changes throughout aligned business process models as for instance proposed in [75] may serve as a starting point for this issue. A proper "migration" concept for C-YAWL port configuration, however, still needs to be developed. In vBPMN, the required propagation mechanisms are conceptually realized by splitting the process logic into a reference process, adaptation patterns and adaptation rules. When changing the reference process or an adaptation patterns, all variants resulting from the adaptation rules are automatically updated.

5.3.3. Suggested Concept Improvements

Given the favorable impact of modularization concepts, an interesting question is whether C-YAWL could be extended with these. Based on the experiences we gained throughout the conduct of the experiments and the creation of the C-YAWL models, we can formulate the following two propositions in this direction:

- In general, the use of subprocesses is considered beneficial for an improved understandability of process models [28]. For example, the tasks "both live in municipality", "unmarried", "contact living municipality" and "no acknowlegement" are not subject to any variability. As such, they could be extracted to a subprocess and could be represented by a single task in the original municipality reference process. In this respect, the existing *worklet* extension [76] for YAWL may be of relevance. A worklet can basically be considered as a modular subprocess, which can be dynamically invoked throughout the course of a process instance according to rules on the process data context. This mechanism can also be employed for process variant management and can generally be combined with C-YAWL. However, it has not yet been investigated how, for example, the nested structure of the municipality process as outlined in Figure 7 would be realized using such an integrated approach. A major issue is that the worklets (=subprocesses) which can be dynamically selected are not combinable, i.e. they are mutually exclusive. This means in the worst case, one has to model one subprocess per variant, which contradicts the original intention of process variant modeling. A potential resolution may consist in the recursive nesting of worklets, i.e. a worklet subprocess may call another worklet again. By these means, combinable variant aspects like "additional task" and "timeout" can be combined as demonstrated for vBPMN in Figure 8. More extensive work on recursive subprocess selection to achieve variability and also runtime flexibility in workflow management systems can be found in [51]

- Instead of letting the modeler only deal with fine-granular port configurations when changing the overall variant model, it should be considered to introduce modular higher-level change operations as discussed in [8] for C-YAWL as well. One example concerns the tedious procedure discussed before to create a new process variant which executes an existing task from the C-YAWL reference process in a different phase of the process. A hypothetical C-YAWL change macro like "*insert variant specific task*" for a selected transition in the reference model could at least in some cases insert the required sequence flows and (re)set the required port configurations automatically. In vBPMN, such change macros are available in the form of adaptation patterns.

5.3.4. Limitations

Internal Validity. Regarding the interpretation of "true" drivers behind our observations, there are some validity threats which have to be considered. The experiment was conducted over a relatively long timeframe (2 hours) compared to other studies in the area of process modeling. It seems likely that the condition of subjects may change over time, not only negatively w.r.t. typical fatigue effects, but also positively in terms of developing a better understanding for process variant modeling. This means that tasks which are processed at a later stage of the experiment are in fact processed with a slightly different background of the subject, constituting a considerable learning effect. As for this work, however, we are mainly interested in finding differences between C-YAWL and vBPMN. This threat to validity is, therefore, mitigated by the fact that the more complicated tasks are conducted at a later stage of the experiment for both tools in an alternating manner across the two control groups.

Furthermore, layouting and other visual factors [77] which we did not explicitly include in our study may have had an impact on the results. We consider the thorough examination of interdependencies between an extended set of factors (including visual aspects) and process variant maintainability as future work.

Next, as in most other studies in this field, a realistic threat may consist of a subconscious bias of the experiment designers towards a specific result. We addressed this issue by mainly relying on case studies as already provided by the designers of C-YAWL instead of designing artificial scenarios from scratch. During the conduct of the experiment, we ensured that each participant disposed of the same level of expertise regarding process variant modeling in C-YAWL and vBPMN by providing

Table 7: Benefits and Room for Improvement for C-YAWL and vBPMN

	vBPMN	C-YAWL
Appreciated Features	• Complex adaptations for variants can be quickly and easily realized by the (re)use of patterns (7). • It is possible to retain a good overview on variability across distinct variants; variability is guided by "adaptive segments" and modular patterns to separate "default" and "special" cases (7). • Adaptation rules are more intuitive for non-technical modelers than Petri net blocking/hiding; there is an explicit relation of the business reason for adaptation and its impact on the process model (4). • The extensibility of the overall variant model is very high (2).	• It is possible to quickly and easily derive a variant from the reference process **IF** the proper port configuration set already exists. (4) • C-YAWL provides a better overview for simple/small process models, as the means for variability do not need to be explored first as in vBPMN (1). • With C-YAWL, it may be easier to realize OR-split variants than with vBPMN, as it is directly possible to configure arbitrary combinations of allowed outgoing paths in one step, which is not possible in vBPMN.(1)
Requested Improvements	• Text search and highlighting of matching tasks is highly desired (6). • The Eclipse-based user interface is partly confusing and should be tidied up (4). • To cope with the additional layer of indirection (rule-based pattern application), a "where used list" for patterns should be introduced to improve the overall overview. It is sometimes not clear what to search or put into the reference process and what in the patterns (4). • It would be desirable to have a more systematic structuring and browsability of the patterns to find the "right" one for the required variant adaptation at hand (1). • The application of multiple adaptation patterns in the vBPMN editor is a bit slow and should be accelerated (1).	• Text search and highlighting of matching tasks is highly desired (6). • There has to be an integrated management and synchronization of the reference process and multiple port configuration sets which constitute the variants. The impact and compatibility of structural modifications with the reference process and with existing port configuration needs to be made more transparent (9). • For a single variant, there should be a better overview on port configurations on the overall process level; currently ports can only be inspected and modified task by task (4). • There should be more guidance for how to use port configurations (e.g. when to use input ports, when to use output ports, when to use blocking, when to use hiding). Associated with this issue, support for realizing higher-level change operations on the reference process structure like "insert task" should be provided also on port configuration level, for example automatically conveying port configurations to the new task (3).

neutral introductory tasks, which were not considered for the experimental results.
We finally acknowledge that the mixed evaluation of concepts and tools might bias the results of the experiments due to, explicitly, latent user interface preferences of the participants. From a pragmatic perspective, however, it is neither possible nor considered valuable to conduct an experiment on process variant maintenance for complex model just by using "pen and paper". Moreover, in practice, the value of concepts for tool-based process modeling may be doubted as long as no corresponding manifestation in software has been achieved as a proof of concept. Usability is mostly driven by a combination of modeling concept and tool implementation [78].

External Validity. Regarding the portability of our results to other domains in terms of, for example, users or process models, we acknowledge that similar to the majority of user studies in the area of business process modeling, the amount of subjects (users) and objects (the amount of examined process models) is relatively small in statistical terms. Due to the fact that process variant modeling requires an advanced process modeling skill set, acquiring a larger number of participants for a corresponding experiment is far from an easy target. We, however, mitigated this issue by the fact that each subject conducted a relatively large (20) number of tasks, leading to 280 measurement points based on which statistical analyses can be conducted.
In addition, with respect to the generalization of the results to other approaches sharing similar features, we acknowledge that only one representative approach including a dedicated tool was evaluated. It can be assumed that a sufficient degree of result generalizability can only be achieved having analyzed a multitude of different approaches and tools per category.
Furthermore, the composition of participant groups might have been influenced by personal background factors which we did not explicitly capture. For example, there are many uncaptured factors which could have had an impact on the evaluation outcomes. Examples relate to the amount of training on modeling-related skills, a-priori experiences with software tools of a particular type, passed course of studies or the current job type. For future work, this issue needs to be more deeply investigated.
Finally, the generalizability of selected process models for process variant modeling may have its limitations. We tried to maximize the portability of results by including a complex variant model resulting from a real-world case study (child registration in Dutch municipalities). Process variants in other domains, however, may be of a different nature.

6. Related Work

The general approach of reference process modeling and using either adaptation or configuration for variant construction is extensively discussed in [4, 79]. The work of [80] judges the quality of a reference model by its *generality*, *usability*, *flexibility*, *completeness* and *understandability*, taking into account that factors may also negatively influence each other. A concrete approach to find a suitable reference model from existing variants is proposed in [81]. The approach tries to merge the distinct variants for example in such a way that the change edit distance to each variant is minimized.

Since process variant modeling is mainly concerned with realizing context-dependent *deviations* between distinct process models, the general field of *process flexibility* is very related to this work. These context-dependent deviations can be imposed at design time, which is equal to process variant modeling as examined by our user study, or at runtime. A full discussion of distinct approaches to model and execute flexible processs is out of scope at this point; we instead refer to corresponding surveys like [82–85]. General frameworks for judging different variability features of process modeling languages and execution systems like guidance or granularity are presented in [84, 86, 87], while a concrete scenario-based evaluation of differences in process variant modeling using four major approaches (C-EPC, Rich BPMN, Provop and YAWL/Worklets) is presented in [88].
Initial insights on the user perception of configurable Event-Driven Process Chains (C-EPC), which are similar to C-YAWL, are provided in [89]. Students were provided with a configurable model and a tool for freely exploring it. They then had to rate the conceptual support of cEPC and the tool support for variant configuration. The authors identify "*an area of improvement as the conceptual support towards configuration consequences is deemed not yet sufficient*"[89]. We have confirmed this issue by our empirical study, as participants found it difficult in C-YAWL to estimate the overall effect of port configurations on the resulting variants (see Table 7).
In [90], C-EPC and Provop (which is similar to vBPMN) are compared on a qualitative level regarding their support of process variant understanding. Their results are complementary to our work, since the authors of [90] explicitly investigate which concepts from cognitive psychology (external memory, abstraction, split-attention effect) affect the human understanding. Although being restricted to a *qualitative* discussion of *understandability*, our empirical results match with their findings in terms of that "*unlike C-EPC, in Provop Boolean expressions are always expressed in terms of context variables. These variables provide semantics to the change options, helping the model reader to understand the intent of the options. [...] we argue that for small models, C-EPC presumably is easier to understand, as all the information is integrated and hence in contrast to Provop no split-attention effect can be expected. However, when model size increases,models may quickly become too complex resulting in an overload for the model reader, especially when there are many relationships between alternative modeling elements.*"[90]

Besides the different kinds of imperative process variant modeling approaches discussed in Section 2, the surveys also contain references to the genre of declarative modeling approaches [91] to realize process variants. For example as proposed by [92], building upon the idea of "pockets of flexibility" [53], variable regions in a process can be defined which contain a loose set of process steps and a set of constraints on these steps. As long as the constraints are not violated, process variants emerge by arbitrarily executing the steps within the variable region.

To examine the general differences in maintaining imperative or declarative process models (but not with a focus on variant management), several user studies have been conducted [10, 35, 36, 93]. One finding presented in [36] is that the realizability of a maintenance operation for an imperative or declarative process depends on its type in terms of whether it is a sequential (relating to process step ordering) or circumstantial (relating to data dependencies) change.

There is also a variety of dedicated BPMN-related user studies. In [37], empirical evidence is presented supporting the hypothesis that the usability (including maintainability) of UML Activity Diagrams is equal to that of BPMN for process modeling. In [38], a task-based comparison of BPMN to Event-Driven Process Chains (EPC) is conducted. The authors examine the overall process modeling skills of a participant having been trained in one language only. They find that being trained in a specific language has little impact on the overall understanding of process models.

Although we did not explicitly quantify complexity aspects (e.g. number of nodes) and modularization aspects (e.g. number and reuse of components) for our user study, it is strongly related to existing work in the field of process complexity metrics [94–97].

For "flat" process models, different process metrics as for example "number of nodes" or "average incoming sequenc flows" have been examined according to their impact on understandability and modifiability tasks [98]. The work of [11, 99] contains a model for predicting modeling errors based on particular characteristics of a BPMN process model. The work of [100] similarly predicts errors in the EPCs of the SAP Reference Model [101], providing findings in terms of error patterns like: "*a higher number of XOR/OR-splits and AND joins in an EPC increases the error probability*"[100].

The authors of [28] examined how decomposing a flat complex process modeling into hierarchically aligned modules supports process understanding. In [102], the hypothesis that higher structuredness leads to fewer modeling errors is supported by quantifying the *structuredness* metrics for a set of a process model and comparing it against the occurrence rate of modeling errors int the same set. In [103] valid reasons for violating structuredness in a process model are discussed, while general process modeling guidelines to reduce modeling errors are contained in the work of [59].

A broader overview on the factors which influence process model comprehension, for example also considering task labeling or background knowledge, is supplied in [33, 34].

Related to process modularization are techniques for process abstraction, which for example try to identify coherent segments in a process model which can be replaced by a representative element and/or be extracted to subprocesses [25, 104, 105]. Such abstraction or process syntax modification [106, 107] mechanisms are for example required when maintaining and refactoring a large collection of process models (which are potentially variants of each other) within a repository [108–110].

To the best of our knowledge however, there is no dedicated work which empirically examines the differences in modeling languages based on a realistic process variant scenario. Normally, tasks in contemporary user studies on process modeling consist only in applying changes to one existing model, not taking its existing variants into account as in the setup of our user experiment.

7. Conclusion

This paper addressed the existing lack of empirical insights into the effects of process model complexity and the type of variant management approach on the maintainability of process variants. For the different types of process variant modeling approaches, we considered two dimensions as especially relevant: their modularization support and the construction direction of process variants. Accordingly, we selected vBPMN as a reference process adaptation (extension) approach with modularization support and C-YAWL as a reference process reduction (configuration) approach without modularization support. Building upon existing case studies on process variants, we realized a simple as well as a complex process variant model for each approach.

Based on the created models and the available tools for C-YAWL and vBPMN, we carried out a controlled randomized experiment. Each participant had to execute a particular sequence of variant maintenance (including understanding and modification) tasks using both of the approaches. We measured the error rate and speed, as well as the subjective concept understanding and perceived easiness for each task. The findings can be summarized as follows:

- Given unlimited time, process model complexity does not significantly impact the modeler's success rate for process variant maintenance tasks. This can be explained by the fact that variant understanding or modification is usually executed on a rather localized part of the process; a complete understanding of it may not be required.

- Process model complexity significantly and negatively impacts the speed of process variant maintainability. This seems intuitively correct. For example, it is harder to spot the proper set of model elements required for the processing of a process variant understanding task in a model containing many nodes, arcs or indirections (e.g. subprocess layers).

- Process model complexity does not significantly impact the subjective perception of process variant maintainability. The explanation is similar to (1.)

- The professional level of a participant does not have a significant impact on success rate, speed or subjective perception. However, we designed the corresponding binary independent variable relatively coarse-granular, such that a finer-granular experimental setup by decomposing the variable may yield other results.

- vBPMN performs significantly better than C-YAWL regarding the success rate of process variant understanding and modification tasks. An explanation we offer is that keeping an overview on port configurations across multiple variants is error-prone, especially when changing the reference model.

- vBPMN performs significantly better than C-YAWL regarding the execution speed of process maintenance tasks. One of the main drivers seems to be the better support of vBPMN for the modularization and reuse of variant aspects as adaptation patterns. These partly abstract from low-level change operations on the reference model, which are required in C-YAWL.

- vBPMN partially performs significantly better than C-YAWL regarding the subjective perception of process variant maintainability by human modelers. While the convenience of working with either approach does not seem to significantly differ, participants ranked the ease of use for vBPMN significantly higher. They generally found it more natural to work with adaptation patterns and adaptation rules for process variant construction than with low-level port configurations.

Our qualitative analysis of the participants feedback indicated that proper modularization support is crucial for process variant management. There is an outspoken preference for high-level change patterns on process models over fine-granular configurations operations like port blocking or hiding for variant construction. Moreover, it has been recognized that the proper propagation of C-YAWL port configurations is not trivial, even for simple changes to the reference model. For an approach like vBPMN relying on adaptation patterns, changes to the reference model only need to be considered if variation points are moved or deleted.

The above insights are valuable, since they provide directions for further developing existing approaches and to guide end-users in the selection of these for their daily work.

Multiple opportunities for future research remain, based on our contributions. First, we only examined the maintenance of existing process variant models and not their creation from scratch. This might also be a decision criterion for or against the selection of a specific approach. Furthermore, a larger participant group and a richer set of factors, like visual aspects of the process models, detailed backgrounds of the participants or additional perspectives on process models will be targeted. Finally, the participants of our study were granted an unlimited amount of time for processing their tasks. It would be highly interesting to see how the error rate for tasks would be affected by setting time thresholds to put different degrees of pressure on the participants.

Acknowledgments

We thank Patrick Fischer for methodological support with the statistical evaluation of our study results. This work has partly been funded by the German Ministry for Research and Education under grant no 01ISO7009 ("INFOSTROM").

References

[1] F. Schönthaler, G. Vossen, A. Oberweis, T. Karle, Business Processes for Business Communities - Modeling Languages, Methods, Tools, Springer, 2012.

[2] C. Richardson, C. Moore, C. Le Clair, R. Vitti, The Forrester Wave: Business Process Management Suites , Q3 2010, Technical Report, Forrester, 2010.

[3] M. Dalgarno, D. Beuche, Variant Management, in: 3rd British Computer Society Configuration Management Specialist Group Conference, 1, Oxford.

[4] J. Becker, P. Delfmann, Reference Modeling - Efficient Information Systems Design Through Reuse of Information Models, Physica-Verlag, 2007.

[5] R. Conradi, B. Westfechtel, Version models for software configuration management, ACM Computing Surveys 30 (1998) 232–282.

[6] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, J. Mendling, Similarity of business process models: Metrics and evaluation, Information Systems 36 (2011) 498–516.

[7] A. Hallerbach, T. Bauer, M. Reichert, Capturing variability in business process models: the Provop approach, Journal of Software Maintenance and Evolution 22 (2009) 519–546.

[8] B. Weber, M. Reichert, S. Rinderle-Ma, Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems, Data & Knowledge Engineering 66 (2008) 438–466.

[9] M. Genero, E. Manso, A. Visaggio, G. Canfora, M. Piattini, Building measure-based prediction models for UML class diagram maintainability, Empirical Software Engineering 12 (2007) 517–549.

[10] D. Fahland, J. Mendling, H. A. Reijers, B. Weber, M. Weidlich, S. Zugal, Declarative versus imperative process modeling languages: the issue of maintainability, in: Business Process Management (BPM'09) Workshops, Springer, Ulm, 2010, pp. 477–488.

[11] E. Rolon, L. Sanchez-Gonzalez, F. Garcia, F. Ruiz, M. Piattini, D. Caivano, G. Visaggio, Prediction Models for BPMN Usability and Maintainability, in: IEEE Conference on Commerce and Enterprise Computing (CEC'09), IEEE, 2009, pp. 383–390.

[12] L. C. Briand, C. Bunse, J. W. Daly, A Controlled Experiment for Evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs, IEEE Transactions on Software Engineering 27 (2001) 513–530.

[13] R. Harrison, S. Counsell, R. Nithi, Experimental assessment of the effect of inheritance on the maintainability of object-oriented systems, Journal of Systems and Software 52 (2000) 137–179.

[14] M. Rosemann, W. M. P. van der Aalst, A configurable reference modelling language, Information Systems 32 (2007) 1—-23.

[15] F. Gottschalk, W. M. P. van der Aalst, M. H. Jansen-Vullers, M. La Rosa, Configurable Workflow Models, International Journal of Cooperative Information Systems (IJCIS) 17 (2008) 177–221.

[16] R. Müller, U. Greiner, E. Rahm, AgentWork: a workflow system supporting rule-based workflow adaptation, Data & Knowledge Engineering 51 (2004) 223–256.

[17] A. Awad, S. Sakr, M. Kunze, M. Weske, Design by Selection : A Reuse-Based Approach for Business Process Modeling, in: 30th International Conference on Conceptual Modeling (ER'11), Springer, Belgium, 2011, pp. 332–345.

[18] A. Schnieders, F. Puhlmann, Variability mechanisms in e-business process families, in: 9th International Conference on Business Information Systems (BIS'06), volume 85, GI e.V., Klagenfurt, 2006, pp. 583–601.

[19] M. Weidmann, F. Kötter, M. Kintz, D. Schleicher, R. Mietzner, Adaptive Business Process Modeling in the Internet of Services (ABIS), in: Sixth International Conference on Internet and Web Applications and Services (ICIW), pp. 29–34.

[20] A. Charfi, H. Müller, M. Mezini, Aspect-Oriented Business Process Modeling with AO4BPMN, in: ECMFA, Springer, Paris, 2010, pp. 48–61.

[21] M. Döhring, B. Zimmermann, vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules, in: 16th International Conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'11), Springer, London, 2011, pp. 332—-341.

[22] S. Meerkamm, Staged configuration of multi-perspectives variants based on a generic data model, in: 2nd International Workshop on Reuse in Business Process Management (rBPM'11), Springer, 2011, pp. 326–337.

[23] M. Rastrepkina, Managing Variability in Process Models by Structural Decomposition, in: 2nd International Workshop on Business Process Modeling Notation (BPMN'10), Springer, Potsdam, 2010, pp. 106–113.

[24] M. Döhring, Handling Variants and Adaptation along the Life Cycle of Event-Aware Workflows, Dissertation, University of Jena, 2013.

[25] S. Smirnov, H. A. Reijers, M. Weske, T. Nugteren, Business process model abstraction: a definition, catalog, and survey, Distributed and Parallel Databases 30 (2012) 63—-99.

[26] C. Li, M. Reichert, A. Wombacher, Discovering Reference Models by Mining Process Variants Using a Heuristic Approach, in: 7th International Conference on Business Process Management (BPM'09), Springer, Ulm, 2009, pp. 344–362.

[27] W. M. P. van der Aalst, T. Basten, Inheritance of workflows: an approach to tackling problems related to change, Theoretical Computer Science 270 (2002) 125—-203.

[28] H. A. Reijers, J. Mendling, R. Dijkman, Human and automatic modularizations of process models to enhance their comprehension, Information Systems 36 (2011) 881–897.

[29] R. Langlois, Modularity in technology and organization, Journal of Economic Behavior and Organization 49 (2002) 19–37.

[30] M. La Rosa, M. Dumas, A. Ter Hofstede, J. Mendling, Configurable multi-perspective business process models, Information Systems 36 (2011) 313–340.

[31] B. Kiepuszewski, A. H. M. ter Hofstede, W. M. P. van der Aalst, Fundamentals of control flow in workflows, Acta Inf. 39 (2003) 143–209.

[32] ISO/IEC 25010:2011 9126-1.2: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, Technical Report, 2011.

[33] H. A. Reijers, J. Mendling, A Study Into the Factors That Influence the Understandability of Business Process Models, IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans 41 (2011) 449–462.

[34] J. Mendling, M. Strembeck, J. Recker, Factors of process model comprehension - Findings from a series of experiments, Decision Support Systems 53 (2012) 195–206.

[35] D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich, S. Zugal, Declarative versus imperative process modeling languages: The issue of understandability, in: 14th International Conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'11), Springer, Amsterdam, 2009, pp. 353–366.

[36] M. Weidlich, S. Zugal, J. Pinggera, D. Fahland, B. Weber, H. Reijers, J. Mendling, The impact of sequential and circumstantial changes on process models, in: 1st International Workshop on Empirical Research in Process-Oriented Information Systems (ER-POIS'10), CEUR-WS, Hammamet, 2010, pp. 43–54.

[37] D. Birkmeier, S. Kloeckner, S. Overhage, An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users, in: 18th European Conference on Information Systems (ECIS'10), Pretoria.

[38] J. Recker, A. Dreiling, Does It Matter Which Process Modelling Language We Teach or Use? - An Experimental Study on Understanding Process Modelling Languages without Formal Education, in: 18th Australasian Conference on Information Systems (ACIS'07), Toowomba, pp. 356–366.

[39] A. H. M. ter Hofstede, W. M. P. van der Aalst, M. Adams, N. Russell, Modern Business Process Automation - YAWL and its Support Environment, Springer, 2010.

[40] S. Meerkamm, Configuration of Multi-Perspectives Variants, in: 1st International Workshop on Reuse in Business Process Management (rBPM'10), Springer, Hoboken, 2010, pp. 277–288.

[41] J. Vanhatalo, H. Völzer, F. Leymann, Faster and more focused control-flow analysis for business process models through SESE decomposition, in: 5th International Conference Service-Oriented Computing (ICSOC'07), Springer, Vienna, 2007, pp. 43–55.

[42] O. Kopp, D. Martin, D. Wutke, F. Leymann, The Difference Between Graph-Based and Block-Structured Business Process Modelling Languages, Enterprise Modelling and Information Systems Architecture 4 (2009) 3–25.

[43] O. Kopp, H. Eberle, F. Leymann, T. Unger, The Subprocess Spectrum, in: Business Process and Service Science (BPSC'10), GI e.V., 2010, pp. 269—-279.

[44] N. Russell, A. H. M. ter Hofstede, W. M. P. van der Aalst, YAWL: Yet Another Workflow Language (Revised version), Information Systems 30 (2005) 245–275.

[45] C.-A. Petri, Kommunikation mit Automaten, Dissertation, University of Darmstadt, 1962.

[46] M. Rosemann, J. C. Recker, C. Flender, Contextualization of Business Processes, International Journal of Business Process Integration and Management 3 (2008) 47–60.

[47] M. La Rosa, W. M. van der Aalst, M. Dumas, A. H. M. ter Hofstede, Questionnaire-based variability modeling for system configuration, Software and System Modeling 8 (2008) 251—-274.

[48] M. Döhring, B. Zimmermann, E. Godehardt, Extended Workflow Flexibility using Rule-Based Adaptation Patterns with Eventing Semantics, in: Informatik2010 Service Science, GI e.V., Leipzig, 2010, pp. 216—-226.

[49] M. Döhring, B. Zimmermann, L. Karg, Flexible Workflows at Design- and Runtime using BPMN2 Adaptation Patterns, in: 14th International Conference on Business Information Systems (BIS'11), Springer, Poznan, 2011, pp. 25–36.

[50] M. Döhring, S. Heublein, Anomalies in Rule-Adapted Workflows - A Taxonomy and Solutions for vBPMN, in: 16th European Conference on Software Maintenance and Reengineering (CSMR'12), IEEE, Szeged, 2012, pp. 117–126.

[51] M. Döhring, A. Schulz, I. Galkin, Emulating Runtime Workflow Adaptation and Aspect Weaving by Recursive Rule-Based Sub-Process Selection - A Model Transformation Approach, in: 16th IEEE International Enterprise Distributed Object Computing Conference (EDOC'12), IEEE, London, 2012, pp. 133–142.

[52] OMG, Business Process Model and Notation (BPMN) Version 2.0 - 2011-01-03, Technical Report January, Object Management Group, 2011.

[53] S. Sadiq, W. Sadiq, M. Orlowska, in: 20th International Conference on Conceptual Modeling (ER'01.

[54] A. Hallerbach, T. Bauer, M. Reichert, Configuration and management of process variants, International Handbook on Business Process Management, 1 (2010) 237–255.

[55] C. Li, M. Reichert, A. Wombacher, Mining Business Process Variants: Challenges, Scenarios, Algorithms, Data & Knowledge Engineering 70 (2011) 409–434.

[56] N. Russell, W. M. P. van der Aalst, A. ter Hofstede, Workflow Exception Patterns, in: 18th International Conference on Advanced Information Systems Engineering (CAiSE'06), Springer, Luxembourg, 2006, pp. 288—-302.

[57] A. Lanz, B. Weber, M. Reichert, Workflow Time Patterns for Process-aware Information Systems, in: 11th International Workshop on Business-Process and Information Systems Modeling (BPMDS'10), Springer, Hammamet, 2010, pp. 94—-107.

[58] F. Gottschalk, T. A. C. Wagemakers, M. H. Jansen-Vullers, W. M. P. V. D. Aalst, M. L. Rosa, Configurable Process Models : Experiences from a Municipality Case Study, in: 21st International Conference on Advanced Information Systems Engineering (CAiSE'09), June, Springer, 2009, pp. 486–500.

[59] J. Mendling, H. A. Reijers, W. M. P. van der Aalst, Seven Process Modeling Guidelines (7PMG), Information and Software Technology 52 (2010) 127–136.

[60] F. Gottschalk, Configurable Process Models, Dissertation, Technical University of Eindhoven, 2009.

[61] Protos User Manual, Technical Report, Pallas Athena, BV, Apeldoorn, The Netherlands, 2004.

[62] A. Hallerbach, Management von Prozessvarianten, Dissertation, University of Ulm, 2009.

[63] C. Wohlin, R. Runeson, M. Halst, M. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering: An Introduction, Kluwer, 2000.

[64] K. Figl, R. Laue, Cognitive complexity in business process modeling, in: 23rd International Conference on Advanced Information Systems Engineering (CAiSE'11), Springer, London, 2011, pp. 452–466.

[65] R: A Language and Environment for Statistical Computing, Technical Report, R Foundation for Statistical Computing, Vienna, Austria, 2012.

[66] S. Siegel, N. Castellan, Nonparametric Statistics for the Behavorial Sciences, McGraw Hill, 2nd editio edition, 1998.

[67] H. J. Seltman, Experimental Design and Analysis, Technical Report, Carnegie Mellon University, Pittsburgh, 2012.

[68] S. Kuiper, J. Sklar, Practicing Statistics: Guided Investigations for the Second Course, Pearson, 2012.

[69] J. McDonald, Handbook of Biological Statistics, Sparky House Publishing, 2 edition, 2009.

[70] H. Levene, Robust Tests for the Equality of Variance, in: Contributions to Probability and Statistics, Stanford University Press, 1960, pp. 278–292.

[71] S. Lawner Weinberg, S. Knapp, Data Analysis for the Behavioral Sciences Using SPSS, Cambridge University Press, 2002.

[72] Jennifer Larson-Hall, A Guide to Doing Statistics in Second Language Research Using SPSS, Lawrence Erlbaum Associates, 2009.

[73] E. Mooi, M. Sarstedt, A Concise Guide to Market Research: The Process, Data, and Methods Using IBM SPSS Statistics, Springer, 2011.

[74] H. Motulsky, Choosing a statistical test, in: Intuitive Biostatistics, Oxford University Press, 1995.

[75] M. Weidlich, J. Mendling, M. Weske, Propagating changes between aligned process models, The Journal of Systems and Software 85 (2012) 1885–1898.

[76] M. J. Adams, A. H. M. ter Hofstede, D. Edmond, W. M. P. van der Aalst, Worklets : A Service-Oriented Implementation of Dynamic Flexibility in Workflows, in: Cooperative Information Systems (CoopIS '06), Springer, Montpellier, 2006, pp. 291–308.

[77] J. Mendling, H. A. Reijers, J. Cardoso, What Makes Process Models Understandable?, in: 5th International Conference on Business Process Management (BPM'07), Springer, Brisbane, 2007, pp. 48–63.

[78] D. van Greunen, A. van der Merwe, P. Kotze, Factors Influencing BPM Tools - The influence on user experience and user interfaces, International Journal of Computing and ICT Research 4 (2010) 47–57.

[79] J. Becker, P. Delfmann, R. Knackstedt, Konstruktion von Referenzmodellierungssprachen- Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle, Wirtschaftsinformatik 46 (2004) 251–264.

[80] S. Matook, M. Indulska, Improving the quality of process reference models : A quality function deployment-based approach, Decision Support Systems 47 (2009) 60–71.

[81] C. Li, M. Reichert, A. Wombacher, Mining business process variants: Challenges, scenarios, algorithms, Data & Knowledge Engineering 70 (2011) 404—-434.

[82] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, W. M. P. van der Aalst, Process flexibility: A survey of contemporary approaches, in: Advances in Enterprise Engineering I (CAiSE'08 Workshops), Springer, Montpellier, 2008, pp. 16–30.

[83] T. Burkhart, P. Loos, Flexible business processes-evaluation of current approaches, in: Multikonferenz Wirtschaftsinformatik 2010, Göttingen, pp. 1217–1228.

[84] F. Daoudi, S. Nurcan, A benchmarking framework for methods to design flexible business processes, Software Process: Improvement and Practice 12 (2007) 51–63.

[85] M. Kapuruge, J. Han, A. Colman, Support for Business Process Flexibility in Service Compositions: An Evaluative Survey, 21st Australian Software Engineering Conference (ASWEC'10) (2010) 97–106.

[86] C. Ayora, V. Torres, V. Pelechano, Dealing with Variability in Business Process Models : An Evaluation Framework - ProS-TR-2011-11, Technical Report, Universidad Politecnica De Valencia, 2011.

[87] S. Nurcan, A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts, in: 41st Annual Hawaii International Conference on System Sciences (HICSS'08), IEEE, 2008, pp. 378–378.

[88] C. Ayora, V. Torres, V. Pelechano, BP Variability Case Studies Development using different Modeling Approaches - ProS-TR-2011-03, Technical Report February, Universidad Politécnica de Valencia, 2011.

[89] J. Recker, M. Rosemann, W. M. P. van der Aalst, On the user perception of configurable reference process models initial insights, in: 16th Australasian Conference on Information Systems (ACIS'05), Sydney.

[90] V. Torres, S. Zugal, B. Weber, M. Reichert, C. Ayora, V. Pelechano, A qualitative comparison of approaches supporting business process variability, in: 3rd International Workshop on Reuse in Business Process Management (rBPM 2012), Springer, 2012.

[91] I. Rychkova, G. I. L. Regev, A. Wegmann, Using declarative specifications in business process design, Journal of Computer Science 5 (2008) 45 –68.

[92] R. Lu, S. Sadiq, G. Governatori, On managing business processes variants, Data & Knowledge Engineering 68 (2009) 642–664.

[93] B. Mutschler, B. Weber, M. Reichert, Workflow Management versus Case Handling - Results from a Controlled Experiment, in: 23rd Annual ACM Symposium on Applied Computing (SAC'08), ACM, Ceará, Brazil, 2008.

[94] J. Cardoso, Business Process Quality Metrics: Log-Based Complexity of Workflow Patterns, in: On the Move to Meaningful Internet Systems (OTM'07), Springer, 2007, pp. 427–434.

[95] R. Laue, V. Gruhn, Complexity Metrics for Business Process Models, in: 9th International Conference on Business Information Systems (BIS'06), GI e.V., Klagenfurt, 2006, pp. 1–12.

[96] H. Reijers, I. Vanderfeesten, W. M. P. van der Aalst, Cohesion and coupling metrics for workflow process design, Computers in Industry (2008) 420–437.

[97] I. Vanderfeesten, H. A. Reijers, W. M. P. van der Aalst, Evaluating workflow process designs using cohesion and coupling metrics, Computers in Industry 59 (2008) 420–437.

[98] G. Canfora, M. Piattini, F. Ruiz, C. A. Visaggio, A family of experiments to validate metrics for software process models, Journal of Systems and Software 77 (2005) 113–129.

[99] E. Rolón, J. Cardoso, F. García, F. Ruiz, M. Piattini, Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models, in: 10th International Workshop on Enterprise, Business-Process and Information Systems Modeling (BPMDS'09), Springer, Amsterdam, 2009.

[100] J. Mendling, H. Verbeek, B. van Dongen, W. M. P. van der Aalst, G. Neumann, Detection and prediction of errors in EPCs of the SAP reference model, Data & Knowledge Engineering 64 (2008) 312–329.

[101] T. A. Curran, A. Ladd, G. Keller, SAP R/3 Business Blueprint: Understanding the Business Process Reference Model, Prentice-Hall, 1997.

[102] R. Laue, J. Mendling, The impact of structuredness on error probability of process models, in: 2nd International United Information Systems Conference (UNISCON'08), Springer, Klagenfurt, 2008, pp. 585–590.

[103] R. Laue, V. Gruhn, Good and Bad Excuses for Unstructured Business Process Models, in: 12th European Conference on Pattern Languages of Programs (EuroPLoP '07), UVK - Universitaetsverlag Konstanz, Irsee, Germany, 2007, pp. 279–290.

[104] A. Polyvyanyy, S. Smirnov, M. Weske, The triconnected abstraction of process models, in: 7th International Conference on Business Process Management (BPM'09), 26, HPI, Uni Potsdam, Springer, Ulm, 2009, pp. 229–244.

[105] S. Yongchareon, C. Liu, X. Zhao, M. Kowalkiewicz, BPMN Process Views Construction, in: 15th International Conference on Systems for Advanced Applications (DASFAA'10), Springer, 2010, pp. 550–564.

[106] M. La Rosa, P. Wohed, J. Mendling, A. H. M. ter Hofstede, H. A. Reijers, W. M. P. van der Aalst, Managing Process Model Complexity via Abstract Syntax Modifications, IEEE Transactions on Industrial Informatics 7 (2011) 614–629.

[107] M. La Rosa, A. H. M. ter Hofstede, P. Wohed, H. A. Reijers, J. Mendling, W. M. P. van der Aalst, Managing Process Model Complexity via Concrete Syntax Modifications, IEEE Transactions on Industrial Informatics 7 (2011) 255–265.

[108] Z. Yan, R. Dijkman, P. Grefen, Business process model repositories - Framework and survey, Information and Software Technology 54 (2012) 380–395.

[109] B. Weber, M. Reichert, J. Mendling, H. A. Reijers, Refactoring large process model repositories, Computers in Industry 62 (2011) 467–486.

[110] R. Dijkman, M. La Rosa, H. A. Reijers, Managing Large Collections of Business Process Models - Current Techniques and Challenges, Computers in Industry 63 (2012) 91–97.

Appendix A. Handout for User Study on Comparing vBPMN and c-YAWL

Workflow Adaptation & Configuration

Survey and User Study

PURPOSE OF THE EVALUATION

Due to environmental changes, companies often need to quickly adapt their processes or add new variant behavior for example due to the integration of new subsidiaries. The proper integration of variant control-flow behavior into an existing reference workflow can be a tedious task. For process configuration, techniques exist which allow the determination of new allowed paths per variant in an all-embracing "master workflow". This approach shows you the whole complexity and all execution alternatives "at once" when modeling. As an alternative, we present an approach for decomposing variant modeling complexity into a "slim happy path" reference workflow, outsourcing variant behavior into modular and reusable adaptation patterns.

In this evaluation, our goal is to find out the strengths and weaknesses of each way of adapting and configuring workflows to different variant behavior. It is structured as follows:

- Brief questionnaire on process modeling background (5 minutes)
- Introduction to the BPMN and YAWL modeling languages and to the vBPMN and C-YAWL configuration mechanisms and tools. (25 minutes)
- Short hands-on training in both tools. (15 minutes)
- Evaluation of simple process model configuration (10 comprehension questions, 5 modeling tasks, 30 minutes).
- Evaluation of complex process model configuration (10 comprehension questions, 5 modeling tasks, 40 minutes)
- Post-evaluation questionnaire (5 minutes)

PRE-QUESTIONNAIRE

We would like to ask you to provide us with some first information on your professional background and your relation to workflow or rule modeling Please complete this short questionnaire at the end of the session and return it to the SAP contact (markus.doehring@sap.com). Thank you.

☐ Multiple selection (or none)
○ Exactly 1 selection

1. **What is your age?**

   ____

2. **Are you male or female?**

   ____

3. **Please describe your occupation/profession and role?**

   ____

1805

4. **What is type and field of your highest completed education or university degree?**

____

5. **Which process modeling (or flow diagram) languages are you familiar with?**

☐ BPMN ☐ EPC ☐ UML AD ☐ YAWL ☐ other: ________ ☐ none

6. **How many years of experience with process modeling do you have (0 if none)?**

____

7. **How many process models have you already read or looked at in total?**

____

8. **How many process models have you modified?**

____

9. **How many process models have you created from scratch?**

____

10. **What is the estimated size (number of nodes) of process models you primarily deal(t) with?**

____

The following questions only apply to participants with a sufficient experience and background in process modeling.

11. **How often do you think these processes are changed?**

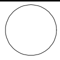
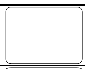
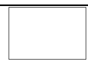
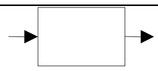
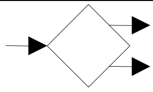
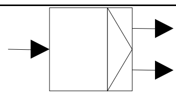
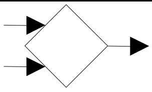
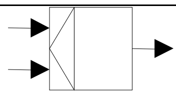
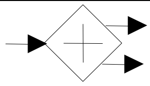
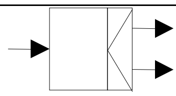
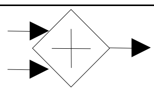
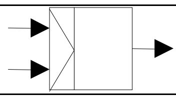
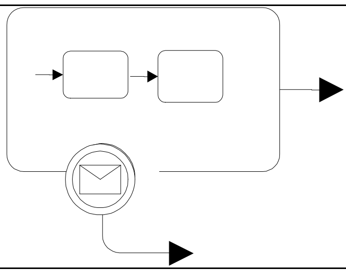
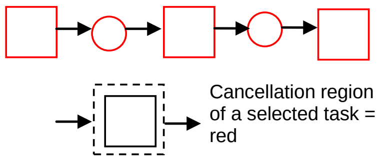
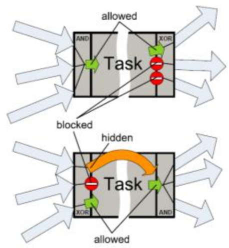
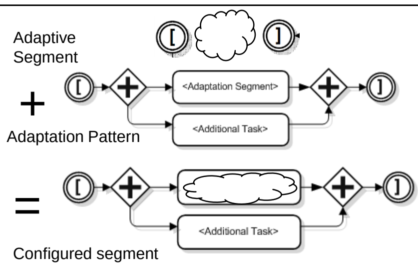
Hourly → ○ ○ ○ ○ ○ (yearly) ○ ○ ○ ○ ○ ← never ☐ don't know

12. **If processes are changed or tailored, what type of adjustment is typically conducted?**

☐ insertion/deletion of activities ☐ integration of time constraints ☐ cancellation mechanism for activites ☐ making activities optional ☐ other: ___ ☐ don't know

Thank you for your feedback!

PROCESS MODELING AND CONFIGURATION CHEAT SHEET

PROCESS MODELING	BPMN	YAWL
State: a passive component indicating a condition in which the process is.	no correspondence	
Task: an active component representing a unit of work.		
Sequence flow: determines “what may happen next”.		
Exclusive branching: determines that after the node is completed, only one of the multiple outgoing paths may be chosen for activation		
Exclusive synchronization: determines, that the node is activated as soon as one of the incoming paths is activated.		
Parallel branching: determines, that after the node is completed, all of the multiple outgoing paths are activated.		
Parallel synchronization: determines, that the node is activated as soon as all incoming paths are activated.		
Cancellation: as soon as the task or event is activated, all elements in its “cancellation region” are aborted.		 Cancellation region of a selected task = red
CONFIGURATION	vBPMN	C-YAWL
Conditional blocking: multiple incoming or outgoing paths can be disallowed for a particular configuration.	no correspondence	
Adaptive segment: Indicates that the segment enclosed in square brackets can be modified with adaptation patterns.		No correspondence
Pattern-based adaptation: determines “what may happen next”.		No correspondence

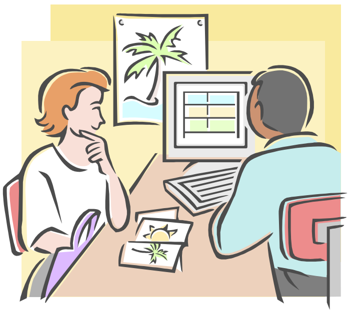
1807

I. INTRODUCTORY TASK

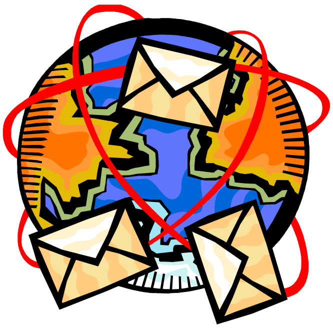
On your two screens, you see a very simple workflow where first A is executed and then B or C. In vBPMN as well as in C-YAWL, please generate two variants of the workflow as follows:

- For variant 1, task A should be skipped
- For variant 2, a task Z can be executed as an alternative to B or C.

II. TRAVEL BOOKING WORKFLOW



Variant 1: Travel Agency



Variant 2: Online Booking

Imagine a travel booking workflow executed within a company offering different kinds of travel services. The workflow more or less consists of four phases, namely order initiation, the choice of different booking services (e.g. hotel or train ticket bookings, receiving the payment for booked services and finally issuing the documents.
Currently, the company offers its services in two different forms: one is the traditional booking in a travel agency; one is online booking via the internet. In the following, you will answer a variety of comprehension questions and process a number of modeling tasks related to differing behavior between the two process variants displayed in the different tools on your computer screen.

A) COMPREHENSION QUESTIONS

TOOL 1

I. CAN "BOOK HOTEL" BE CANCELLED FOR BOTH VARIANTS?

ANSWER:
Yes O No O

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1O 2O 3O 4O 5O ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1O 2O 3O 4O 5O ← absolutely

TOOL 1

II. FOR WHICH VARIANTS CAN “CREDIT-CARD PAYMENT” POTENTIALLY BE EXECUTED?

ANSWER (Multiple):
Online Booking ☐ Travel Agency ☐

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

III. CAN “CREDIT-CARD PAYMENT” POTENTIALLY BE EXECUTED FOR BOTH VARIANTS?

ANSWER:
Yes ○ No ○

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

IV. FOR WHICH VARIANTS CAN “CASH-PAYMENT” POTENTIALLY BE EXECUTED?

ANSWER (Multiple):
Online Booking ☐ Travel Agency ☐

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

1809

B) MODELING TASKS

TOOL 1

I. ONLY FOR THE AGENCY, SKIP THE TASK "BOOK HOTEL" AND ADD A TASK "BOOK TRAIN" TO THE BOOKING TASKS INSTEAD. THE TASK SHOULD HAVE THE SAME CANCELLATION BEHAVIOR AS THE OTHER BOOKING TASKS.

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 1

II. CREATE A NEW VARIANT "ONLINE BOOKING 2" WHICH IS EQUAL TO NORMAL ONLINE BOOKING, BUT FOR WHICH THE DOCUMENTS ARE ADDITIONALLY PROVIDED VIA EMAIL

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

III. **ONLY FOR THE ONLINE BOOKING, SKIP THE TASK “BOOK HOTEL” AND ADD A TASK “BOOK FLIGHT” TO THE BOOKING TASKS INSTEAD. THE TASK SHOULD HAVE THE SAME CANCELLATION BEHAVIOR AS THE OTHER BOOKING TASKS.**

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1O 2O 3O 4O 5O ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1O 2O 3O 4O 5O ← absolutely

TOOL 2

IV. **CREATE A NEW VARIANT “AGENCY2” WHICH IS EQUAL TO NORMAL AGENCY BOOKIN, BUT FOR WHICH THE “REDUCTION CARD” TASK IS SKIPPED**

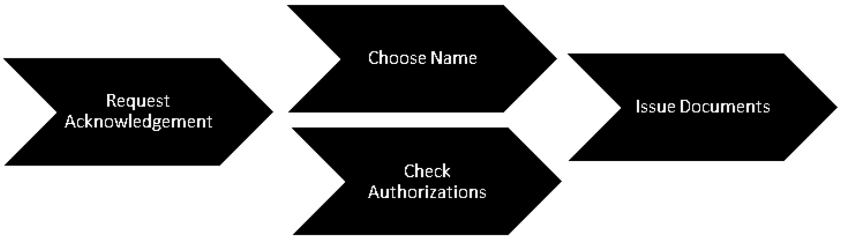
Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1O 2O 3O 4O 5O ← absolutely

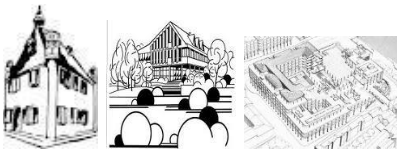
How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1O 2O 3O 4O 5O ← absolutely

1811

II. NAME REGISTRATION OF UNBORN CHILD



3 Different Municipalities



Imagine a name registration process for a child at a public municipality. First, one or both parents request acknowledgement of the child. Then, a variety of steps to determine the name of the child is performed together with checks if the requester is authorized to determine the name. The actual extensiveness and sequence of these steps heavily depends on peculiarities of the particular respective municipality in which the process is executed.

In the following, you will answer a variety of comprehension questions and process a number of modeling tasks related to differing behavior between the three process variants displayed in the different tools on your computer screen.

A) COMPREHENSION QUESTIONS

TOOL 1

I. FOR MUNICIPALITY 1, ARE "LAST NAME MOTHER" AND "MIN 1 PRESENT" MUTUALLY EXCLUSIVE, I.E. THEY CANNOT BOTH BE EXECUTED WITHIN A PROCESS?

Start Time: _______ Finishing Time: _______

ANSWER:
Yes O No O

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1O 2O 3O 4O 5O ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1O 2O 3O 4O 5O ← absolutely

TOOL 1

II. CAN THE TASKS "DETERMINE NATIONALITY" AND "DETERMINE IF AUTHORIZATION" BE EXECUTED IN ARBITRARY ORDER WITHIN MUNICIPALITY 2 OR 3?

Start Time: _______ Finishing Time: _______

ANSWER:
Yes ○ No ○

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 1

III. FOR WHICH MUNICIPALITIES CAN THE TASK "INFORM AUTHORITIES" BE EXECUTED?

Start Time: _______ Finishing Time: _______

ANSWER (Multiple):
Municipality 1 ☐ Municipality 2 ☐ Municipality 3 ☐

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

IV. FOR MUNICIPALITY 2, ARE "FIRST CHILD OF THE RELATION" AND "LASTNAME MOTHER" MUTUALLY EXCLUSIVE, I.E. THEY CANNOT BOTH BE EXECUTED WITHIN A PROCESS?

Start Time: _______ Finishing Time: _______

ANSWER:
Yes ○ No ○

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

V. DOES THE TASK "BOTH LIVE IN MUNICIPALITY" ALWAYS OCCUR AFTER "FIRST CHILD OF RELATION IN MUNICIPALITIES 1 AND 2?

Start Time: _______ Finishing Time: _______

ANSWER:
Yes ○ No ○

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

VI. FOR WHICH MUNICIPALITIES CAN THE TASK “DETERMINE NATIONALITY” BE EXECUTED?

Start Time: _______ Finishing Time: _______

ANSWER (Multiple):
Municipality 1 ☐ Municipality 2 ☐ Municipality 3 ☐

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

B) MODELING TASKS

TOOL 1

I. **PROPERLY REMOVE THE TASK “LAST NAME MOTHER” FROM THE MODEL, MAINTAINING ALL OF ITS SOURROUNDING PATHS, SUCH THAT THE REMAING BEHAVIOUR OF THE VARIANTS REMAINS UNCHANGED.**

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 1

II. **INTRODUCE A TASK “CONFIRM PAST LIVING MUNICIPALITIES” DIRECTLY BEFORE “BOTH LIVE IN THE MUNICIPALITY”, KEEPING THE OVERALL BEHAVIOR FOR THE REST OF THE WORFLOW (I.E. THE NEW TASK SHOULD BE EXECUTED ONLY IF “BOTH LIVE IN THE MUNICIPALITY” ALSO IS EXECUTED).**

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

III. ONLY FOR MUNICIPALITY 1, TASK “DETERMINE NATIONALITY” SHOULD BE EXECUTED DIRECTLY AFTER “CONFIRM IDENTITY”, THEN PROCEED AS BEFORE

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

IV. PROPERLY REMOVE THE TASK "UNMARRIED" FROM THE MODEL, MAINTAINING ALL OF ITS SOURROUNDING PATHS, SUCH THAT THE REMAINIG BEHAVIOUR OF THE VARIANTS REMAINS UNCHANGED.

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

V. INTRODUCE A TASK "BOTH PARENTS ARE FULL AGE" DIRECTLY AFTER "BOTH PARENTS PRESENT", KEEPING THE OVERALL BEHAVIOR FOR THE REST OF THE WORFLOW (I.E. THE NEW TASK SHOULD BE EXECUTED ONLY IF "BOTH PARENTS PRESENT" ALSO IS EXECUTED).

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

TOOL 2

VI. ONLY FOR MUNICIPALITY 3, “INFORM AUTHORITY” SHOULD BE EXECUTED DIRECTLY AFTER “CONFIRM IDENTITY”, THEN IT SHOULD BE PROCEEDED AS BEFORE

Start Time: _______ Finishing Time: _______

How confident did you feel when processing this task, i.e. was it clear to you how to achieve the task?

Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

How easy was the processing of this task for you, i.e. could you easily and quickly realize what you wanted to achieve?

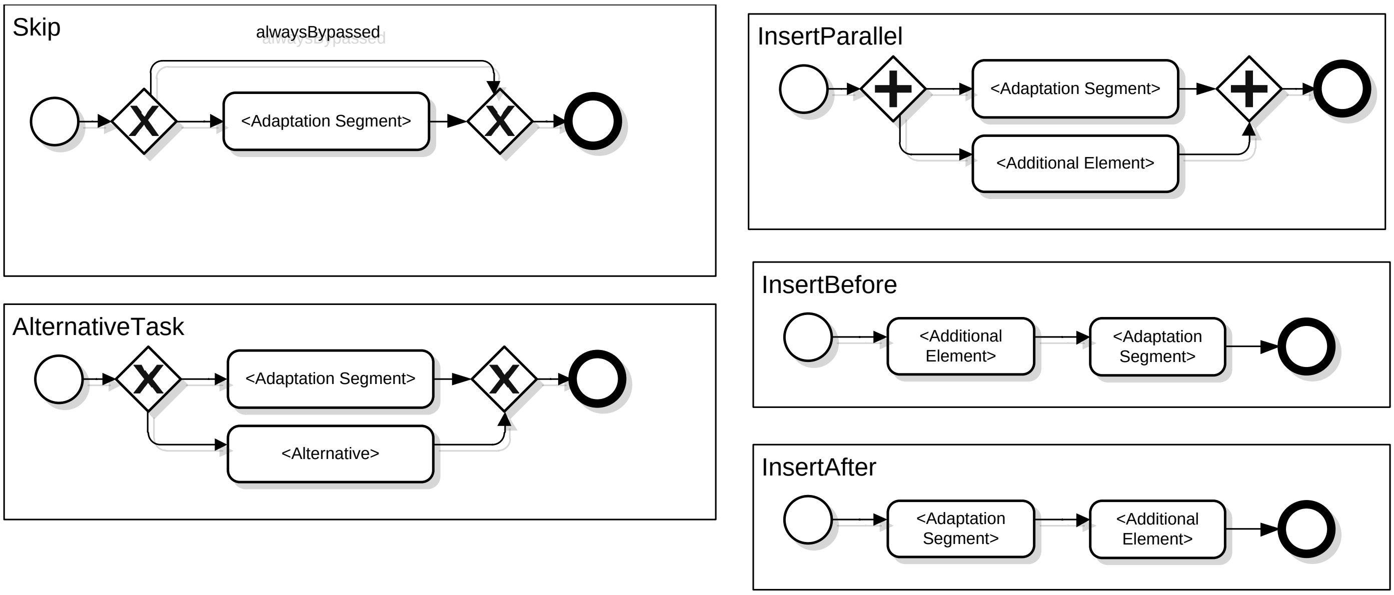
Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely

EX-POST QUESTIONNAIRE

	vBPMN	C-YAWL
Finding logical parts in the model was easy and convenient	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
Understanding the overall model was easy and convenient	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
The tool provided me with all means required for process configuration/adaptation.	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
Configuring/adapting the model was easy and convenient.	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
I would use the approach for process variant management.	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
I found the tool features where I would have expected them.	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely	Not at all → 1○ 2○ 3○ 4○ 5○ ← absolutely
2 Things I would like to have improved for the approach.	• . •	• . •
2 thing I would immediately “buy” from the approach	• . •	• . •

Appendix B. Models used as Starting Points for User Tasks

Appendix B.1. Simple Processes used for Introductory Task



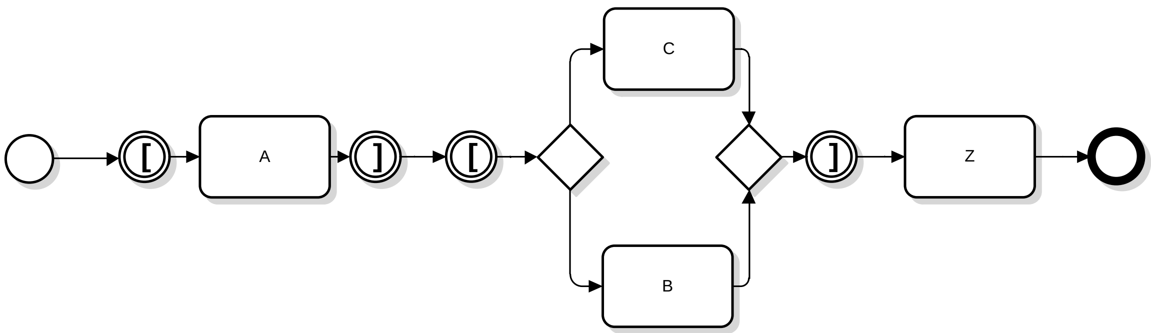


Figure B.14: Simple Process used for Introduction to vBPMN

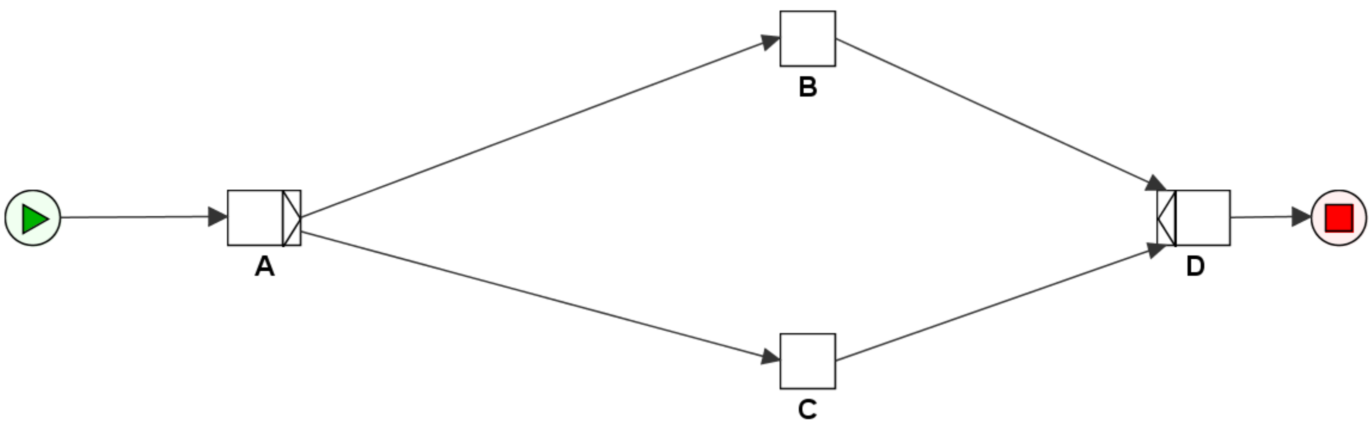


Figure B.15: Simple Process used for Introduction to C-YAWL

Appendix B.2. Variants for Travel Process

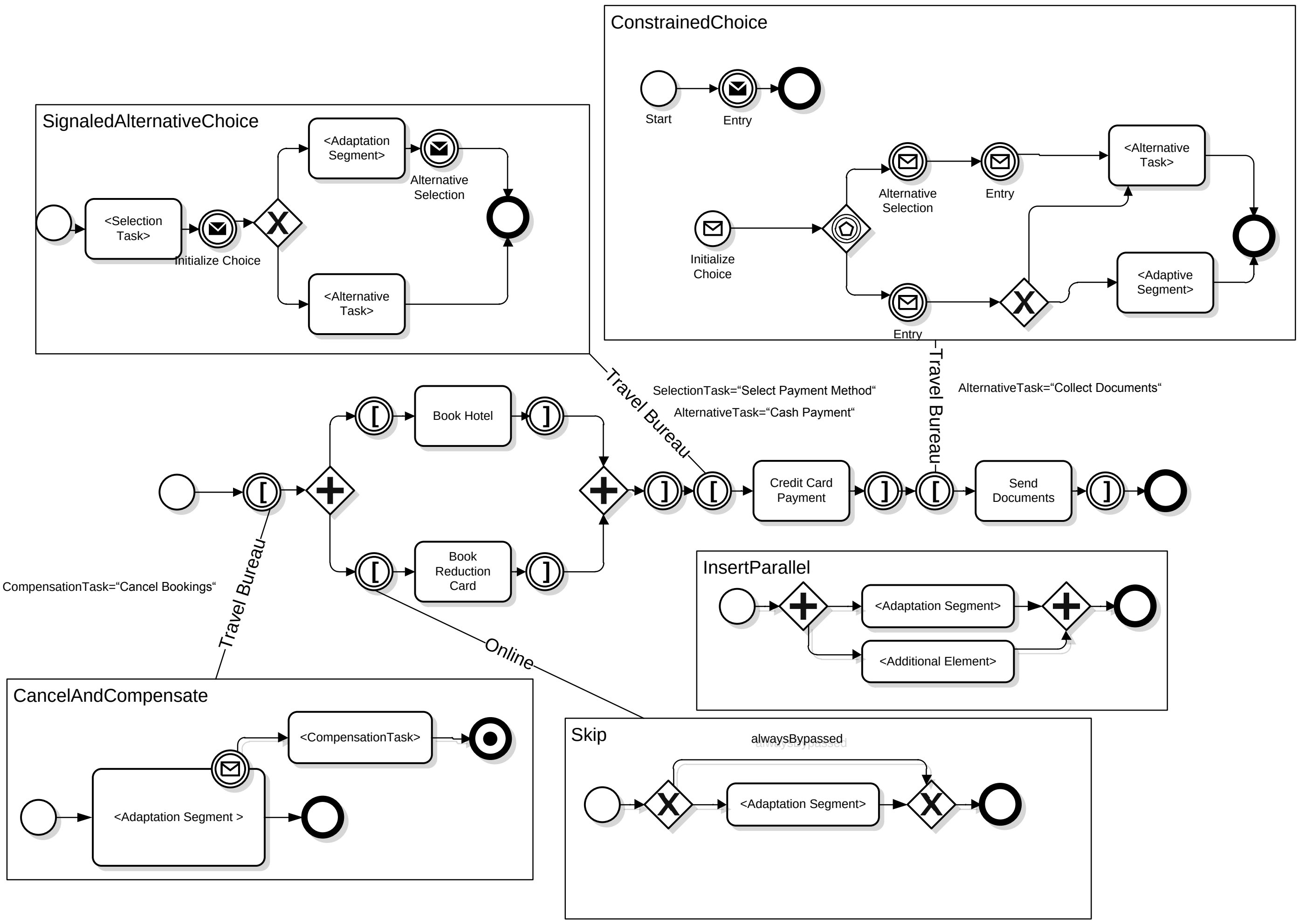


Figure B.16: Travel Process Variants for Online Booking and Travel Agency Realized in vBPMN

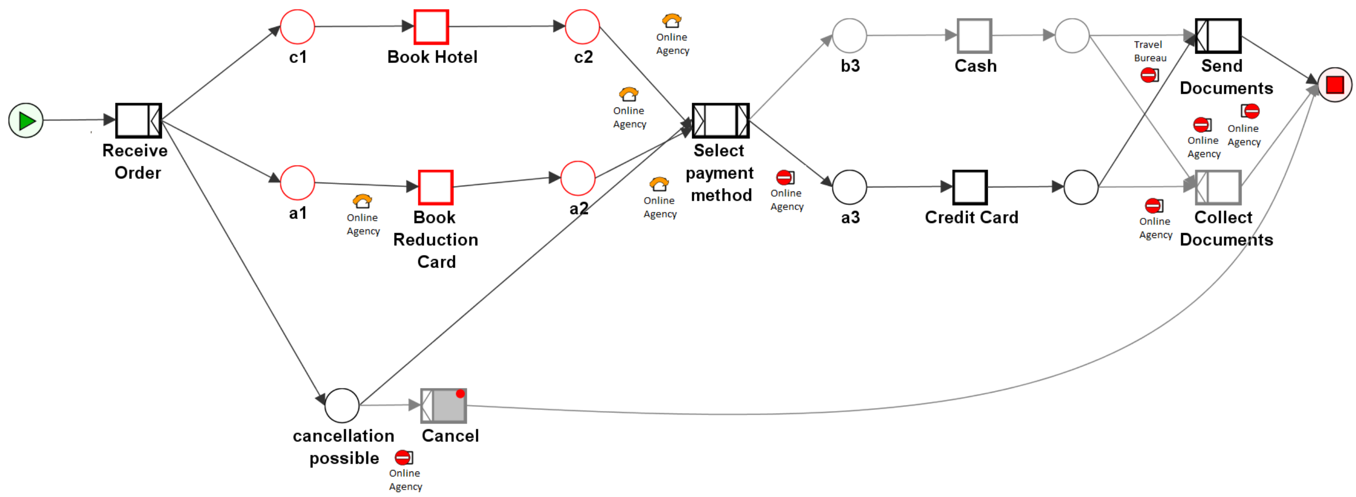


Figure B.17: Travel Process Variants for Online Booking and Travel Agency Realized in C-YAWL

Appendix C. Plots for Target Variables

Figure C.18 shows how the distribution of second values for the target variable “execution time” can be approximated by a normal distribution after a log transformation. The normal distribution assumption is a precondition for the ANOVA statistics we apply on this target variable. Figures C.19, C.20, C.21 contain the boxplots for the three target variables *success*, *Concept Convenience* and *easiness* against the four independent variables *professional level*, *task type*, *model complexity* and *execution tool*, averaged per subject (i.e. participant). Figure C.22 contains the boxplots for the four independent variables against the plain execution time measurements in seconds.

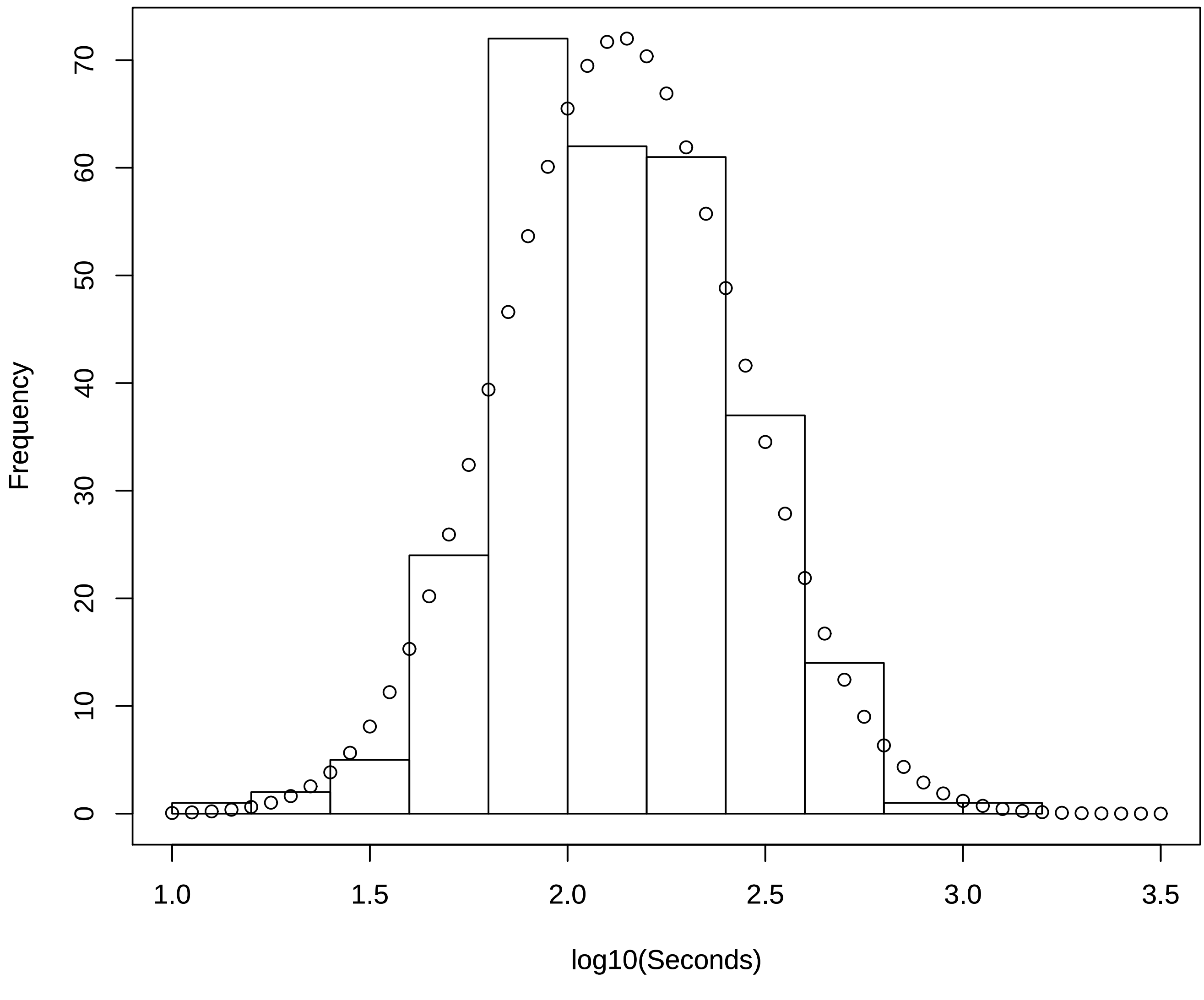
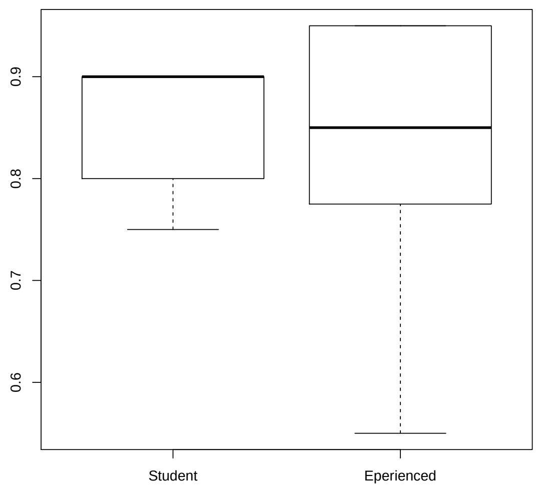
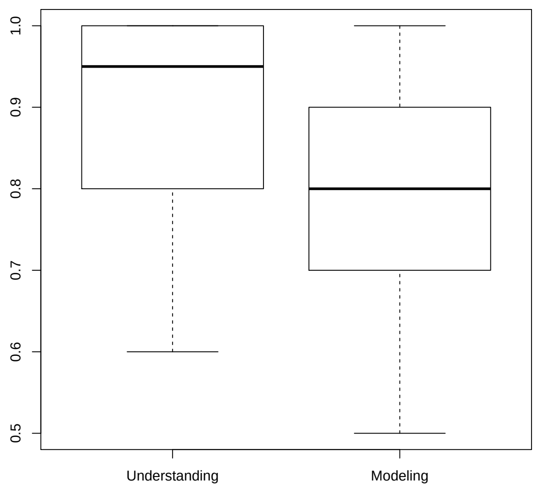


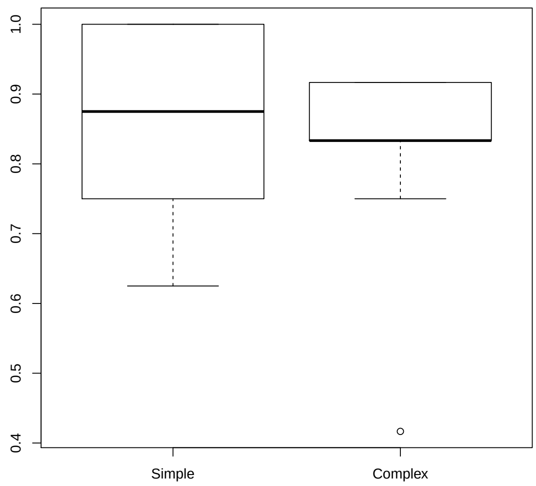
Figure C.18: Approximated Normal Distribution of Log-Transformed Task Execution Times (Precondition for ANOVA)



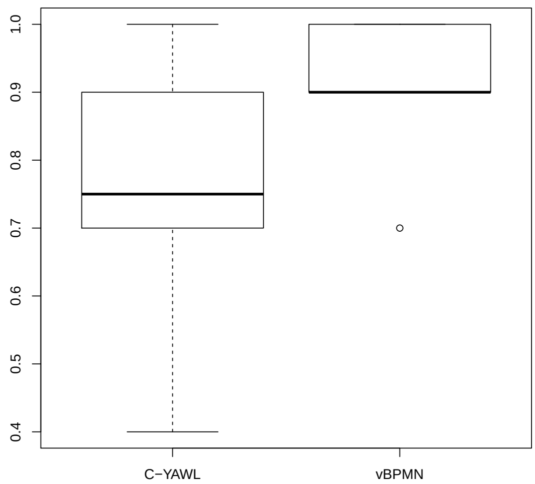
(a) Boxplot for Professional Level against Task Success Rate



(b) Boxplot for Task Type against Task Success Rate

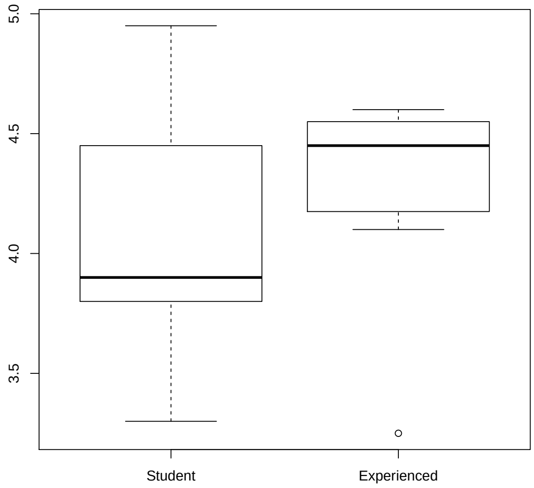


(c) Boxplot for Model Complexity against Task Success Rate

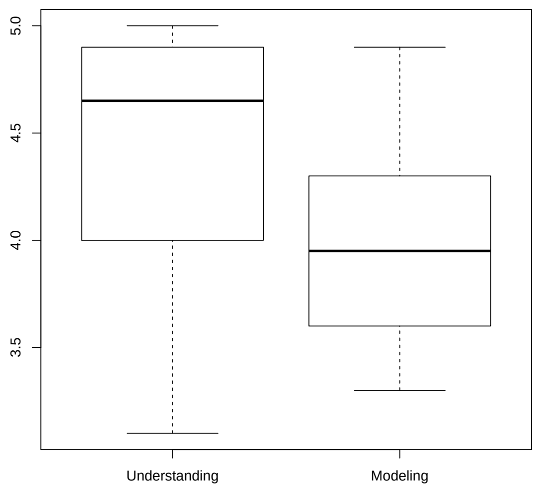


(d) Boxplot for Execution Tool against Task Success Rate

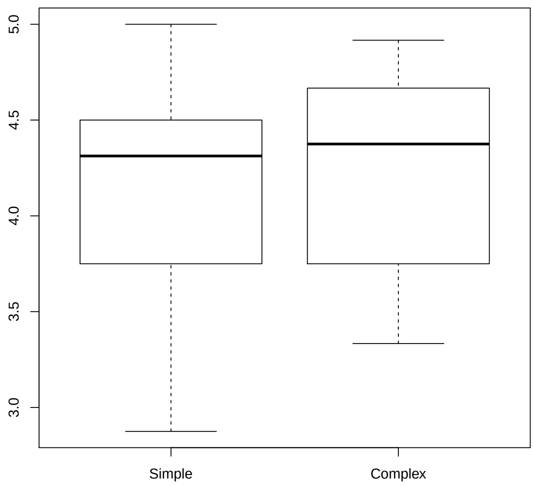
Figure C.19: Boxplots for Response Variable “Success”



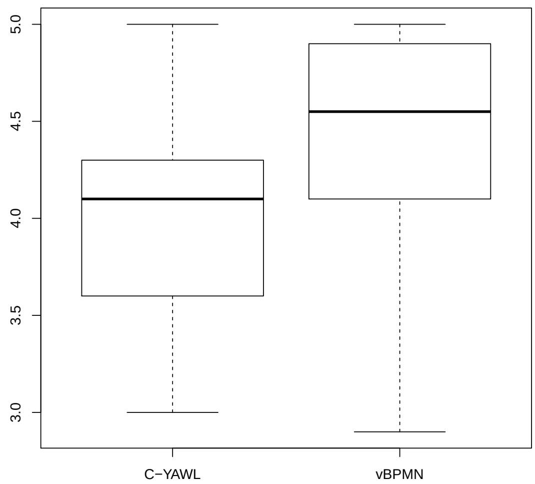
(a) Boxplot for Professional Level against Concept Convenience



(b) Boxplot for Task Type against Concept Convenience

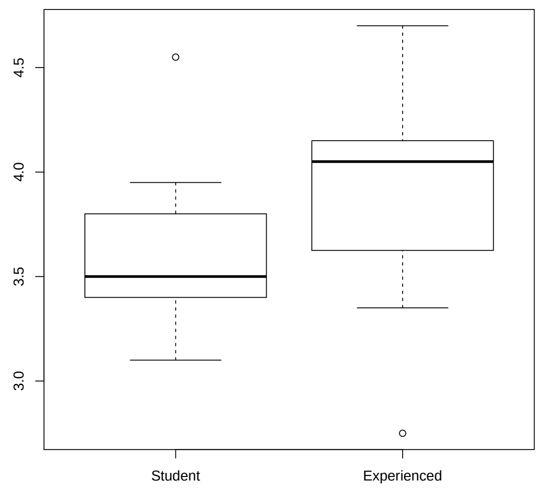


(c) Boxplot for Model Complexity against Concept Convenience

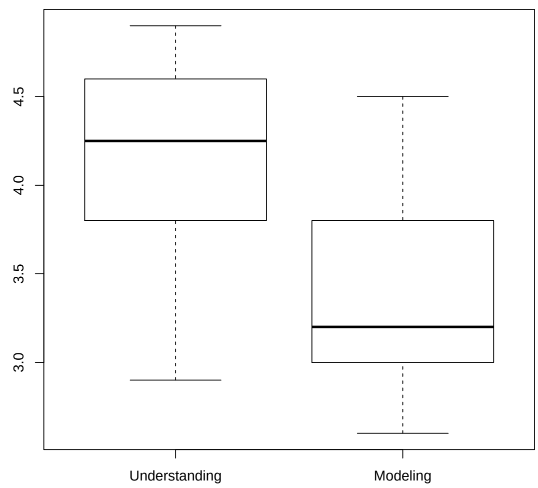


(d) Boxplot for Execution Tool against Concept Convenience

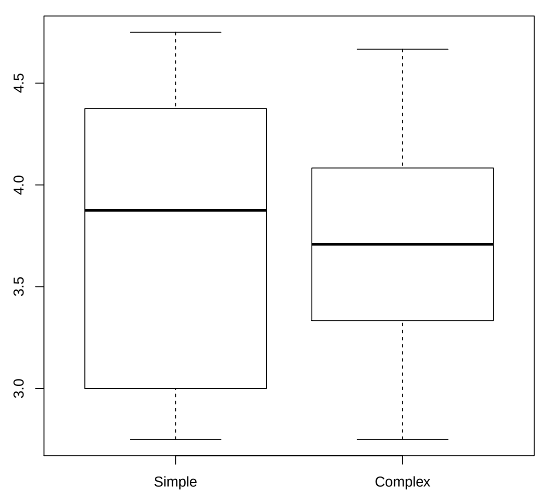
Figure C.20: Boxplots for Response Variable “Concept Convenience”



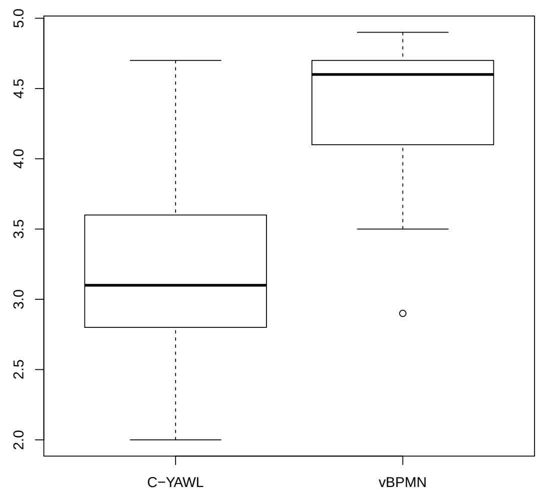
(a) Boxplot for Professional Level against Easiness



(b) Boxplot for Task Type against Easiness

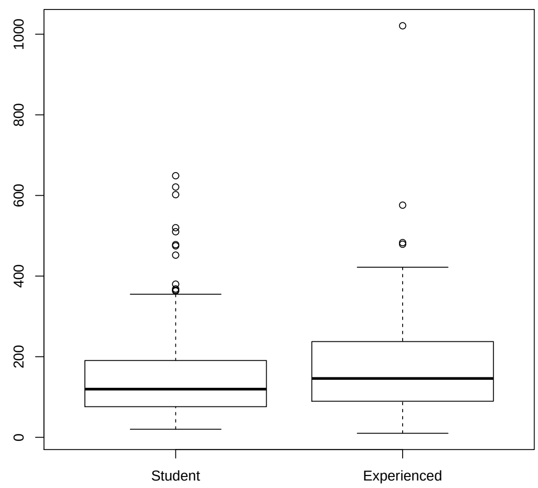


(c) Boxplot for Model Complexity against Easiness

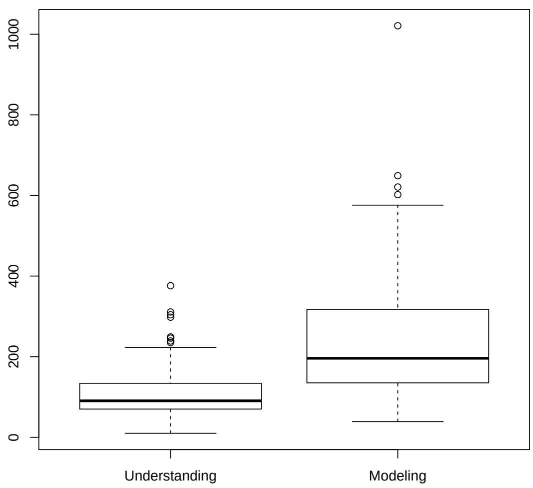


(d) Boxplot for Execution Tool against Easiness

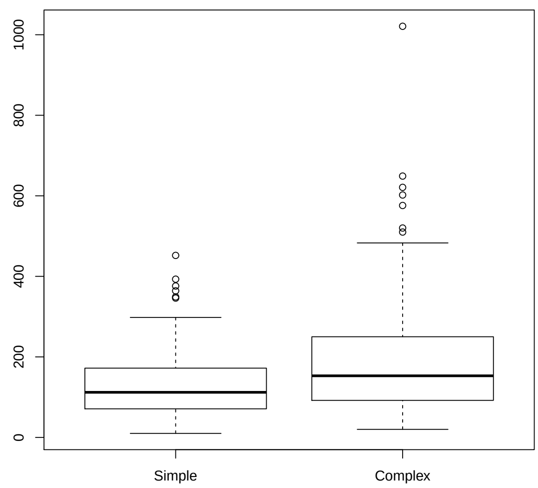
Figure C.21: Boxplots for Response Variable “Easiness”



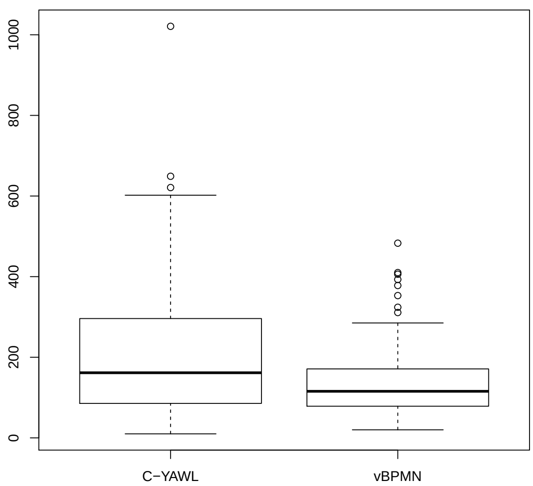
(a) Boxplot for Professional Level Tool against Processing Time



(b) Boxplot for Task Type against Processing Time



(c) Boxplot for Model Complexity Tool against Processing Time



(d) Boxplot for Execution Tool against Processing Time

Figure C.22: Boxplots for Response Variable “Processing Time”