Software & Systems Modeling (2019) 18:1155–1178
https://doi.org/10.1007/s10270-017-0648-z

SPECIAL SECTION PAPER

Selecting a process variant modeling approach: guidelines and application

Banu Aysolmaz[1] · **Dennis M. M. Schunselaar**[2] · **Hajo A. Reijers**[2] · **Ali Yaldiz**[3]

Received: 6 January 2017 / Revised: 13 October 2017 / Accepted: 25 November 2017 / Published online: 9 December 2017

Abstract

Various modeling approaches have been introduced to manage process diversity in a business context. For practitioners, it is difficult to select an approach suitable for the needs and limitations of their organization due to the limited number of examples and guidelines. In this paper, we report on an action research study to perform a comparative process variant modeling application in a process management consultancy company. This company experienced difficulties in maintaining and reusing process definitions of their customers. We describe how the requirements were determined and led to the selection of two specific approaches, the Decomposition Driven Method and the Provop approach. We comparatively evaluated the suitability of these approaches to develop variant models for six software project management processes of five customers. This study contributes to the field by presenting an industrial case for process variant modeling, reporting in-depth, real-life applications of two approaches, applying the approaches for hierarchical processes, and presenting guidelines for choosing an approach under comparable conditions.

Keywords Business process modeling · Process variant modeling · Decomposition driven method · Provop

1 Introduction

Business process models (or process models for short) are among the most important information assets for organizations. They are employed to fulfill a variety of purposes such as documenting processes, communicating process knowledge, and developing Business Process Management Systems (BPMSs) [1]. To manage their process models, which may amount to hundreds of thousands of them, organizations establish process repositories [2]. While the existence of large collections of process models in repositories brings opportunities to an organization, such as reuse of process knowledge, a number of issues arise [3]. One of the these relates to the phenomenon of variability in process models [4]. Due to the diversity in business contexts, variants of the same process may emerge in multiple cases in the same organization [5]. This diversity may be caused by various factors such as differences in delivered products, customer types, and divergent business requirements in countries [6]. When such factors are present, organizations need to develop their process models to effectively represent process variants [7]. However, in the design of a process model, it is a challenging task to either maintain variants of the same process separately while managing the relations between them, or integrate the process variants into a single model while preventing complexity and redundancy [5]. Such difficulties become even more prominent in hierarchical process repositories as interplay between a large number of processes takes place [8].

To overcome issues that relate to the modeling of process variants, various approaches have been proposed to incorporate variant management into the phases of the

Communicated by Dr ’ Ilia Bider and Rainer Schmidt.

✉ Banu Aysolmaz
b.aysolmaz@maastrichtuniversity.nl

Dennis M. M. Schunselaar
d.m.m.schunselaar@vu.nl

Hajo A. Reijers
h.a.reijers@vu.nl

Ali Yaldiz
ali.yaldiz@4s.com.tr

[1] School of Business and Economics, Maastricht University, PO Box 616, 6200 MD Maastricht, The Netherlands

[2] Faculty of Sciences, Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

[3] 4S Information Technologies, Ankara, Turkey

business process management (BPM) life-cycle [9]. The literature provides a categorization and comparison of available approaches [5,9,10]. However, in real-life settings, it is difficult for an organization to make the proper choice between variant modeling approaches. Hardly any studies have been performed on the evaluation and comparison of approaches in practice or even the definition of guidelines to support such a selection [9]. The evaluation of variant modeling approach use is even more limited in hierarchical processes [7]. To compare and evaluate the application of approaches in a real-life setting, we performed an action research study in a company where we observed difficulties in managing process variants. The company we worked with for the action research is 4S Information Technologies (4S for short), a company that provides consultancy services to its customers to analyze and improve their processes and develop BPMSs using HP PPM tools [11]. For each customer, 4S defines a new variant of a process, e.g. software project management, demand management, software change request management, and risk and issue management, in their hierarchical process repositories. 4S maintains separate process definitions for each variant, yet the interrelations between the variants are not tracked. As a result, the maintenance of process models becomes an effort-intensive and error-prone task, since the same changes need to be applied in various similar process models in the repository. Moreover, 4S cannot systematically reuse its process knowledge for creating a new variant for a new customer. For these reasons, 4S was motivated to implement a process variant management approach to efficiently apply its knowledge in process analysis, design and improvement activities.

The aim of this study is to implement and compare process variant modeling approaches in a real-life setting for the analysis and documentation phase of the BPM life-cycle in hierarchical processes. We chose 4S as the suitable candidate to conduct an action research study to fulfill this aim. A team of six employees from the company participated in the study. The team was led by one of the authors and supported by the rest. The team identified the requirements for selecting process variant modeling approaches based on an analysis of the literature. As a result of the selection procedure performed, two different process variant modeling approaches were chosen to evaluate in detail: the Decomposition Driven Method (DDM) [12] and the Provop approach [13]. These approaches were applied to six variants of software project management process and a related subprocess of five 4S customers. The team assessed the business value of applying these variant modeling approaches, and evaluated how the two approaches performed in practice to meet specified user needs and requirements. On the basis of the findings, we present a set of selection guidelines that companies may follow when they face a similar situation to find the most suitable process variant modeling approach.

This paper extends the results of [14] in several ways. We report on the design and performance of the participatory action research study implemented in two iterations of the cyclical action research process. We define the requirements for process variant modeling approaches based on established criteria from the literature. We provide an in-depth analysis of approach applications, and apply and evaluate the approaches for hierarchical processes. We enrich our guidelines with the new insights from these extensions. In this way, this study does not only guide companies in selecting a process variant modeling approach, but also for defining their needs and requirements to make the most suitable selection.

The rest of the paper is structured as follows. Section 2 presents the design of this study including the definition of the research method, description of the case, and the plan. In Sect. 3, we describe the process variant modeling approach selection procedures. In the subsequent sections, we respectively explain the application of our approaches for our industrial case, the Decomposition Driven Method in Sect. 4, and the Provop approach in Sect. 5. Section 6 presents our findings together with the guidelines for approach selection. In Sect. 7, we elaborate on the related work. Section 8 concludes the study.

2 The design of the process variant modeling study

This section consists of the design of our comparative process variant modeling research study. In Sect. 2.1, we describe the research method followed. In Sect. 2.2, we define the industrial case in which we conducted the action research study. Lastly, in Sect. 2.3, we introduce the research plan we followed for the study.

2.1 Research method

This study aims to comparatively evaluate process variant modeling approaches to support practitioners in selecting suitable approaches. The action research method involves researchers and practitioners working on a problem together, developing solutions, and achieving reflective learning [15]. In this way, researchers may obtain first-hand understanding of the phenomenon, while providing value to the participant organization [16,17]. It is extensively used in information systems research [18,19], and found to be a promising method specifically for the BPM field [20]. Action research is suggested for cases where a demonstration or evaluation of an artifact is necessary [20]. For these characteristics, we chose to apply action research in this study. Action research fits particularly well to our research aim, because it enables us to acquire deep knowledge for advancing scientific knowledge while solving practical problems [21,22]. To conduct the action research method for comparatively applying and evaluating process variant modeling approaches, we identified a company in need of process variant modeling as our case. In applying intensive qualitative research methods that examine a phenomenon, such as case study and action research, it is common to perform the study on a single case with carefully-identified units of analysis (in this study, variant modeling approaches) [23–25]. This enables the researchers to get a deeper understanding of the subject and produce richer results. We followed the approach of participatory action research (PAR), which involves practitioners participating in research as both subjects and co-researchers [26]. We followed the cyclical process of action research [17,23]. As a result of the first stage in the cycle, *diagnosis of the problem*, user needs are specified (Sect. 2.2) and requirements for selecting the process variant approaches are determined (Sect. 3.1). The rest of Sect. 3 together with Sects. 4 and 5 report on the stages of *action planning* and *action taking*. Lastly, the application of *evaluation* and *specifying learning* stages are described in Sect. 6. We performed two iterations of the action research cycle, as explained in Sect. 2.3. We applied PAR in an *experimental* format, where the researchers and the company collaborated in all phases to take action and evaluate the consequences [23]. In the next section, we introduce the company selected as the case.

2.2 Description of the case

We chose 4S for this action research as we identified that the needs of the company suit well with the aim of our research. 4S is a consultancy company that provides process analysis, improvement, and automation services to its customers using the HP PPM product [11]. HP PPM provides a flexible process definition and execution environment with a special emphasis on project and demand management process areas. Other process areas 4S works on are change, risk, and issue management. 4S has customers from various countries and industries, which are active in different domains. 4S maintains a process model repository for each customer that contain process models in a hierarchical structure. Due to the fact that 4S provides consultancy services in certain process areas, it has repositories containing multiple variants of the same process. Usually, 4S analysts need to rely on their own expertise to examine and maintain these repositories. Moreover, for new customers, they cannot systematically exploit process knowledge obtained from similar companies that the analysts worked with previously. We identified four user needs of 4S which together motivate the use of process variant modeling as a suitable solution:

User Need 1 Establishing a knowledge-base When 4S analysts start to work with a new customer, they need to combine their knowledge on previous customers for understanding the new as-is processes, suggesting efficiency improvements for the processes, and automating them accordingly. By relying on their expertise, there is an inherent risk that certain aspects are overlooked in the new process definitions. As a result, there is a need for an integrated model which can be used as a jump start for a new customer in the project initiation phase. Process variant models are known to serve this purpose by providing a baseline (reference model) [5,27].

User Need 2 Maintenance When 4S needs to make efficiency improvements or updates in processes, they need to go over each process of each customer to find out which ones are affected. Such updates require a large effort. Furthermore, these updates are a major cause of "smells" in model repositories containing variants of the same process [4]. The changes 4S performs on the processes are either generic or specific. Generic changes affect many processes and, as a result, many variants, e.g., a change in legislation [28]. A specific change only requires certain variant(s) to be updated in harmony with the overall view. By using process variant models, the complications in implementing both the generic and specific changes can be alleviated [7].

User Need 3 Reuse Ideally, 4S consultants can take (reuse) an existing variant of a process at a customer and tailor this toward the needs of a new customer. Unfortunately, there are two obstacles for doing so. First and foremost, there is currently no way to systematically find an existing variant most suitable for a new customer. Second, if an existing variant is tailored, it is not possible to link both variants to each other and to store links to existing processes. One might argue that a query language for model repositories might provide a solution [29,30]. Unfortunately, querying a repository requires specifying part of the expected behavior as a query. This means that the analysts need to have the exact process information they are looking for and formulate complex queries containing fragments of processes. Process variant modeling is an established means for systematic reuse of process knowledge [8,31]. This is often achieved by providing an integrated view of possible variants, removing the need to model the process. Instead, those parts of the process which are applicable for a new customer are selected.

User Need 4 Variability in a hierarchical structure Process variant management is a way to deal with the complexity of large process model collections stored in process repositories [8]. Processes can be kept in a hierarchical structure in such repositories, as is the case in 4S. However, when repositories are considered, analysts need to implement mechanisms to manage variability in hierarchies in addition to individual processes [7,30,32]. Such mechanisms are usually not inherently considered by process variant modeling approaches [9]. Therefore, we need to evaluate the applicability of process variant modeling approaches for a hierarchical repository structure.

In the following section, we explain our research plan for performing the action research in the selected company.

2.3 Research plan

A team of six employees was established in 4S to perform the study. One of the authors of this paper, who is affiliated with 4S, worked as a leader of the team in the company that identified the requirements of 4S, implemented the selected approaches, and evaluated the results. The rest of the authors also supported the team throughout the study. Thus, the team was composed of one of the researchers who led the study in 4S, five other 4S analysts, and the rest of the researchers that provided knowledge on variant modeling approaches and how to apply them. In this way, we established an action research setting where we aimed "to solve current practical problems while expanding scientific knowledge" [17].

The team decided to use Software Project Management (SPM) processes of five customers for this study. Four of the companies were from Turkey, and the other was a Turkish branch of an international company. For all of them, their SPM processes were defined based on PMIs PMBOK guide [33]. Although 4S used the PMBOK guide as the baseline, the best practices provided just the essential steps of a project management process. The SPM process was selected due to its business value for 4S, since it was the most frequent process that 4S works with its customers and it had the highest variation. 4S used the HP PPM tool to define their processes as workflow definitions. The analysts were experienced in using BPMN process models to analyze and communicate processes. The team used available process models and workflow definitions in the company to analyze and discuss the process variants through the application of the two selected approaches. A summary of the selected companies and process metrics for the initially available SPM processes can be found in Table 1. We planned to execute the following steps to fulfill the aim of this research in conformance with the action research cycle [17]:

1. *Identify the process area to apply the process variant modeling approaches* The team selected the SPM process, which was the most frequent process for which 4S provided consultancy for their customers.
2. *Identify the context for application* Five customers were identified that were representatives of different domains. One customer implemented two variants of the SPM process.
3. *Select process variant modeling approaches to compare* In this step, the team defined the requirements of 4S for process variant modeling approaches in conformance with the user needs defined in Sect. 2.2 and organizational characteristics. The team decided to select two approaches to analyze and compare them in detail with a manageable amount of effort.
4. *Apply the selected approaches for the identified process area and the context* The team conducted relevant steps for applying the selected approaches. The team used the available process models and workflow definitions to understand the variant processes, and developed new process models when needed. The two approaches were applied in parallel.
5. *Evaluate the process of approach application and identify findings* The team collected data on the effort spent on each approach and compared the outputs. The team evaluated the process of applying the selected approaches, and performed a comparative analysis on how the approaches meet the user needs and requirements in 4S.
6. *List the selection guidelines for process variant modeling approaches* In the light of the findings provided by the team, a list of guidelines were identified to support practitioners in selecting process variant modeling approaches suitable for their needs.

Table 1 Information on six SPM processes of five customers in 4S

Variant	Domain	# Tasks	# Gateways
Company 1	Insurance	10	4
Company 2	Insurance	12	6
Company 3	Banking	9	0
Company 4-1	Banking	14	6
Company 4-2	Banking	9	2
Company 5	Telecom	11	2
Average		10.8	3.3
PMBOK	N/A	5	0

The team performed the action research cycle in two iterations. In the first iteration, which was reported in [14], all the steps were performed for the top-level SPM process. In the second iteration, which is reported in this paper, user needs were revised, the steps 1 and 4 were repeated for a subprocess, and steps 5 and 6 were performed to integrate the new findings with the previous ones. The detailed analysis in the second iteration resulted in refinements in some of the outputs of the first iteration, which are visible in Table 3 and Figs. 3, 5, 6 and 7.

In the next section, we describe the selection procedure for process variant modeling approaches as the third step of our action research plan.

3 Process variant modeling approach selection

This section contains the selection procedure of process variant modeling approaches to be applied as part of the action research study. In Sect. 3.1, we present the requirements identified toward selecting the relevant approaches for the given case. In Sect. 3.2, we explain the procedure applied for finding candidate approaches from the literature together with a comparison of these approaches based on the evaluation criteria.

3.1 Process variant modeling approach requirements

Next to the aforementioned user needs, also the company and the problem domain pose requirements on the process variant modeling approaches. Within this section, we build upon the work of La Rosa et al. [5]. The team identified these requirements based on their process variant modeling needs and organizational characteristics. In the following requirement list, we map each requirement on the evaluation criteria postulated by La Rosa et al. indicated in parenthesis at the end of the requirement text, if applicable.

To better position the requirements, we have subdivided them into three sets. In the first set, we have the requirements specific for the problem domain, i.e., another company within the same domain would have the same requirements. The second set consists of the requirements specific for 4S, i.e., these are determined by the focus of 4S onto particular solution directions. The third and final set consists of requirements specific for the 4S employees, i.e., these requirements are based on the experience of the employees of 4S.

3.1.1 Domain requirements

The customers of 4S cover a variety of domains, e.g., insurance, banking, and telecom. At the same time, some process variant modeling approaches are found to focus on a specific domain [9]. To ensure that we can explore the application of the selected approach in a generic manner in 4S, our first requirement is:

Requirement 1 *The variability modeling approach needs to be domain-independent.*

3.1.2 4S specific requirements

The variability 4S is trying to capture is during the design phase of the BPM life-cycle. In the work of La Rosa et al. [5], this is called the *conceptual process type*. Next to variability during the design phase, techniques can offer variability during the execution phase. This is currently not in scope of 4S. As a result, our second requirement is:

Requirement 2 *Variability needs to be supported during the design phase of the BPM life-cycle (conceptual process type).*

In 4S, the variability within process models occurs most frequently in the control-flow perspective. The control-flow perspective represents the activities and the relation between them, e.g., sequence, parallel, or a choice. This is the most-studied perspective within variant modeling approaches [5]. In summary, our third requirement is:

Requirement 3 *The variability modeling approach needs to support variability for the control-flow perspective (control-flow).*

To be able to apply an approach in a practical setting, the approach needs to reach a certain maturity level by showing its applicability in practice. This is essential for 4S to ensure that the approaches are applicable to real-life processes. As a result, our fourth requirement is:

Requirement 4 *The variability modeling approach needs to have been validated in real-life (validation).*

In 4S, we need to go beyond variability for a single process model. We need to manage variability within repositories. As mentioned, variability within repositories can be supported by means of hierarchies. This leads to our fifth requirement:

Requirement 5 *The variability modeling approach needs to support a hierarchical process structure.*

3.1.3 Employee specific requirements

Although process variant modeling approaches build on existing process modeling concepts, the introduction of additional constructs and representational elements increase the cognitive load of using them [34]. For experienced modelers, the use of familiar process modeling notations may increase the understanding of process models [35]. Process variant modeling approaches usually build on a specific process modeling notation, or require customization of a notation. In 4S, the analysts are experienced in BPMN for the purposes of analyzing and communicating process knowledge. To alleviate the additional cognitive load for the analysts due to the use of new methods, we have as our sixth requirement:

Requirement 6 *The variability modeling approach needs to build on BPMN as the process modeling notation (process modeling language).*[1]

[1] Since 4S is interested in the approach and not in the implementation, it is not relevant which process modeling notation is used in the implementation of the approach.

The analysts at 4S are new to variability modeling. Furthermore, applying variant modeling approaches is not a trivial task [34]. As a result, for a more successful adoption, our seventh and final requirement is:

Requirement 7 *The variability modeling approach needs to provide decision support on the creation of modeling artifacts and their application (decision support).*

The team decided to select two approaches conforming to these requirements while having distinctive variant modeling mechanisms in alignment with our action research design. Applying different approaches helped us to achieve our research aim to perform a comparative analysis. At the same time, 4S was able to experience diverse approaches and evaluate pros and cons for their future use. In the next section, we describe how the two approaches were selected based on these requirements.

3.2 Selection of the approaches

To identify the potential process variant modeling approaches that meet the requirements as identified in Sect. 3.1, a list of approaches were identified from the most recent surveys and systematic literature reviews [5,9,10]. Next to this, two recent theses were examined on the topic of variability to ensure that the list is up-to-date [36,37].

In total, there are 240 publications listed in [5,9,10,36,37]. After removal of publications in non English, and publications that could not be obtained, we have a total of 238 publications. Some publications were listed multiple times. After removal of the duplicates, we had a total of 160 publications.

For each of these 160 publications, we have read the title, abstract, and keywords. Based on this, we determined whether this work might be relevant for our setting, e.g., some work was related to run-time flexibility, similarity in process models, or correctness of process models. This resulted in 69 publications. Each of these publications has been read to determine the relevance; which resulted in 43 publications being relevant.

We decided to first differentiate the publications based on the used modeling formalism (Req. 6) since this is easiest to verify for a given publications. In total 20 publications contained among others BPMN, e.g., ADOM [27] has been applied to i.a. UML Activity Diagrams, and BPMN. Using, among others, the work of La Rosa [5], our second differentiator was the validation in real life (Req. 4). If [5] indicated that there has been a validation, then we have followed this. If [5] did not indicate there has been a validation, then we have searched for papers possibly containing the validation since a validation paper could have been published after the work of [5]. This resulted in 10 publications related to four approaches, e.g., Provop [13] was listed with 5 different publications. The four approaches are as follows:

- Application-based domain modeling (ADOM) [27],
- Decomposition driven method (DDM) [12],
- Process family engineering in service oriented applications (PESOA) [38], and
- Process variants by options (Provop) [13].

Note that, among others, the ideas for C-EPCs can theoretically be transferred to other formalisms, e.g., C-BPMN (see [37] for a discussion). From 4S, there was a requirement (Req. 4) that the approach has a real-life evaluation. Furthermore, applying an evaluated approach on a different formalism does not necessarily mean that the evaluation will also hold for the other formalism. As a result, 4S did not want to consider approaches that can theoretically be applied upon other process modeling formalisms; even though, from an academic perspective, unevaluated approaches might be more interesting. We present the list of evaluated approaches and our comparison criteria in Table 2.

Within Table 2, we use + if an approach supports a requirement, − if it is not supported, and ± if there is only partial support, e.g., the concepts required for a requirement are listed, but it is not described how to use these concepts.

The evaluation in Table 2 follows the insights as presented by La Rosa et al. [5]. On the requirements not covered by La Rosa et al., we present the reasoning behind the scores. For ADOM [27], we have a "+" for domain independence since the approach does not make any assumptions on the application domain. As the team chose approaches that use BPMN as the process modeling language (Req. 6), hierarchy is natively supported through the decomposition of activities to subprocesses. However, the hierarchy support provided by

Table 2 The list of candidate approaches for process variability modeling and their evaluation

Approach	Design phase	BPMN	Control-flow perspective	Domain independent	Decision support	Validation	Hierarchy support
ADOM [27]	+	+	+	+	−	±	±
DDM [12]	+	+	+	+	±	+	+
PESOA [39]	+	+	±	±	±	+	+
Provop [13]	+	+	+	+	±	±	±

BPMN may not be sufficient for handling variabilities, and additional mechanisms may be needed to manage variability within a hierarchical structure [40]. ADOM gets a "±" for hierarchy support since a subprocess is defined as a *dependent element* as part of the method, but no specific mechanism is provided to manage such elements. DDM is developed for design-time analysis of process variations for the control-flow perspective [12]. The method provides decision support in building variant models via the concept of business driver, though specific support is not defined for configuring the variants. Thus, DDM gets a "±" for decision support. The approach is applied in real-life applications from different domains and the results are validated with domain experts, giving it a "+" for validation. DDM is developed by considering the hierarchical structure of process models and incorporates a decomposition mechanism to the management of variants, so it is scored with a "+" for hierarchy support. For PESOA, we have a "±" for domain independence since the main goal of PESOA is the customization of process-aware software systems [38] and not the process models themselves. PESOA provides partial decision support by means of the definition of constraints over feature values. However, guidance is not provided for selecting an appropriate feature set to configure variants. PESOA defines techniques to encapsulate varying subprocesses and configuring them using feature models, giving it a "+" for hierarchy support. Provop has a "+" for domain independence since it has been applied in automotive and healthcare industries [41]. It provides decision support not for modeling the variants but configuring them by designing options that include reusable operations ("±" for decision support). Provop has a "±" for hierarchy support since it mentions the use of subgraphs to represent one level of decomposition as part of the options.

The team selected two approaches to be able to comparatively apply different process variant modeling approaches that meet their requirements but offer distinctive variant modeling mechanisms. The selected approaches were DDM and Provop. PESOA was excluded due to limitations in managing variations in the control-flow perspective and limitations in decision support. ADOM was eliminated due to a lack of decision support. We evaluated that the selected approaches, DDM and Provop, were good representatives of diverse variability management mechanisms due to the following approach differences:

- *The two approaches have different inherent types* [5]. Provop provides variability support by both (1) extension of one simpler base process to derive a specific variant and (2) by restriction of an extensive base process by removing unrelated elements. DDM follows a restriction approach by providing an extensive variation map that represents all possible variants of subprocesses.
- *The two approaches use different techniques for building the configurable process model* [9]. Provop follows a multi-artifact approach through the definition of an option list together with the so-called base process. DDM is a single-artifact approach capturing all variability in an integrated variation map.
- *The two approaches have different approaches for hierarchy support*. DDM is natively built on a decomposition mechanism to represent variability. Thus, theoretically, it supports variability in process repositories. Provop directly implements the BPMN language, thus by definition should be able to support a hierarchical structure. However, it is a topic open for exploration for Provop.

In the following two sections (Sects. 4, 5), we describe the application of the DDM and Provop approaches in a stepwise manner. Then, we compare the two applications and present the findings in Sect. 6.

4 Applying the decomposition driven method

The approach starts with the definition of a main top-level process [12]. Then, each activity in the main process is defined in detail in a subprocess. Later, the subprocesses are further decomposed into subprocesses until there is no meaningful decomposition possible. At every level, the so-called variation map is created which contains activities and relations necessary to configure every variant. In the following sections, we describe the execution of each step as prescribed by the approach [12].

4.1 Step 1: Model the main process

The team started by developing a main SPM process that acts as a process map applicable to all variants. The process can be seen in Fig. 2. All activities of the best practice model in Fig. 1 were used. Two more activities, *Plan Project* and *Create Asset*, were added, since they were found to be common and frequent activities in process variants. The established process model captures the milestones of the process and provides a basis for further decomposition to subprocesses.

4.2 Step 2: Identify variation drivers

A key tenet of DDM is the consideration of variation drivers, which include business and syntactic drivers, to understand the emergence of variations and using them to flexibly develop the models [42]. Business drivers are the drivers causing variations in an organization's processes due to its business environment. To understand its business environment and identify relevant business drivers, an organization

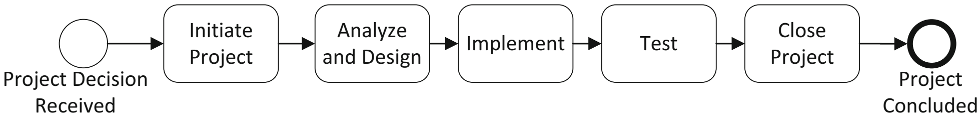


Fig. 1 Best practice process model

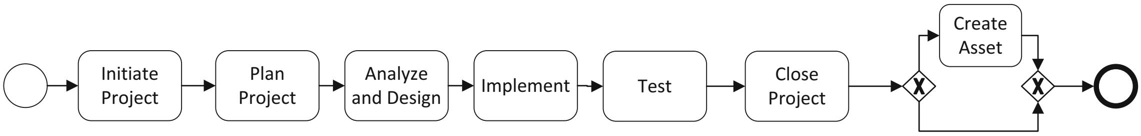


Fig. 2 Software project management (SPM) process

may consider the product and services produced (*what*), physical and virtual markets it operates in (*where*), internal and external customers (*who*), and time intervals such as seasons that affect operation (*when*) [12]. An example business driver in the "*what*" category is different products customized for customers, and an example in the "*when*" category is high season when the number of process executions increase. Other reasons of variation identified based on the subjective judgment of domain experts, such as variations in processes due to the different ways employees perform, are identified as syntactic drivers.

In our case, the team focused on how the activities in SPM process in Fig. 2 are performed and possible causes of variation. The team observed that the main cause of variation is the variety of 4S customers, or *companies*, which is identified as the main business driver. Another driver was identified as the *location* of the services. This driver was used to differentiate the processes of Company 4, leading to different process variants for national and international services. No syntactic drivers were identified by the team for SPM process. Accordingly, when we refer to "drivers" in the rest of the paper, we consider only the business drivers identified.

4.3 Step 3: Assess the relative strength of variation drivers

In this step, variation drivers are analyzed to specify their priority as well as their effect on defining variants [43]. The business driver with the highest priority was found to be the *companies* in our case. This driver was found as the distinguishing driver for process variants of five companies. Additionally, the team identified the driver for the *location* of services. This driver was used to define two variations for Company 4 processes: international (variant 4-1) versus national (variant 4-2) services. Although two variants were identified for Company 5 at the beginning, they were decided to be merged as a distinguishing driver was not found.

4.4 Step 4: Identify the variants of each subprocess

In this step of applying DDM, a variation matrix is developed that shows the variants of each subprocess of the main process by using the identified variation drivers, and discussing how the processes are performed [12]. The team populated a variation matrix for each subprocess of SPM process as suggested by the approach. The variation matrix can be seen in Table 3. To generate this matrix, the team first identified the set of activities performed for the subprocess of each activity, and categorized those sets into a table format (Table 4). This table helped the team to reveal possible variations of each subprocess. For example, six different variants were identified for the "Plan Project" subprocess in this way, while fewer variants were identified for the other subprocesses. The categories used (i.e. basic, fast, simple, moderate, detailed, and complex) served as labels that reflect the characteristics of these subprocesses based on their cost and speed. For example, the variants in the "basic" category reflect the leanest way of performing the related subprocess, the "fast" variants represent a more costly but still time-efficient way, the "simple" variants describe less-detailed and fast options with respect to the rest. From "moderate" to "detailed" and "complex" categories, the time and cost of executing the variants increase. It should be noted that these categories are used only within the context of 4S and may not be relevant in other cases.

DDM suggests a subjective identification and categorization of variants by domain experts. However, since no systematic approach is suggested by DDM, it is highly possible that variant definitions are updated in later stages as more details are revealed about subprocesses [12]. The team also needed to change some of the variant definitions after the first iteration of the research cycle [14], resulting in updates in Table 3 and Fig. 3. The approach followed by the team at the second iteration via Table 4, variant identification by defining activities and assigning them to categories, provided a basis for team members to systematically observe possible variants, discuss their similarities based on the activities, and come to an agreement on the variants.

Table 3 Variation matrix for immediate subprocesses of SPM process (refined from [14])

	Initiate project	Plan project	Analyze and design	Implement	Test	Close project	Create asset
Company 1	Complex initiation	Moderate planning	Basic analysis and design	Basic implementation	Basic test	Detailed closure	
Company 2	Moderate initiation	Complex planning	Detailed analysis and design	Detailed implementation	Detailed test	Complex closure	Asset creation
Company 3	Simple initiation	Basic planning	Basic analysis and design	Basic implementation	Basic test	Basic closure	Asset creation
Company 4-1	Detailed initiation	Simple planning	Detailed analysis and design	Detailed implementation	Basic test	Fast closure	
Company 4-2	Simple initiation	Fast planning	Basic analysis and design	Basic implementation	Basic test	Simple closure	Asset creation
Company 5	Basic initiation	Detailed planning	Detailed analysis and design	Detailed implementation	Detailed test	Moderate closure	

In the variation matrix in Table 3, the subprocess variants applicable for each business driver are depicted. For example *"Simple Initiation"* variant of *Initiate Project* subprocess is used by Company 3 and Company 4-2. This variant can then be related to a variant category in Table 4, for this example "simple" as no 3. As can be seen in Table 4, this subprocess includes two activities: *Initiate Project* and *Assign PM*. *"Complex Initiation"* variant used by Company 1 (having the category "complex" in no 6 of Table 4) includes a wider extent of activities, namely *Initiate quality control* and *Approve scope*.

4.5 Step 5: Perform similarity assessment of variants for each subprocess

In this step, the team assessed the similarity of variants by analyzing each subprocess of the variation matrix in Table 3. The team discussed the similarity of the activities in the subprocesses for each driver. To evaluate the similarity, the experts focused on how those activities are performed. For this, they used the activities defined in Table 4 for each variant and investigated the information on data used and produced while performing activities, and the workflow steps. As a result, the team decided which variants are similar and marked them gray in Table 4. For example, while *"Moderate Initiation"* variant of *Initiate Project* subprocess contained an activity named *Approve Initiation*, *"Detailed Initiation"* variant did not contain this activity but another one called *"Announce Initiation"*. The team indicated that these two activities are actually highly similar, thus merging the *"Moderate Initiation"* and *"Detailed Initiation"* variants. Additionally, the *"Moderate Planning"* variant was marked to be similar with the *"Simple Planning"* variant for the *Plan Project* activity, and the *"Basic Closure"* variant was marked to be similar with the *"Fast Closure"* variant for the *Close Project* activity. For *Analyze and Design*, *Implement*, and *Test* activities, basic and detailed variants were assessed to be similar. While DDM suggests subjective assessment of similarity of variants, the team observed that the use of Table 4, which was not defined as part of DDM, proved to be helpful to visualize variants and evaluate their similarities.

4.6 Step 6: Construct the variation map

In the previous steps, the team obtained the variation matrix and the similarity assessments of the variants. Using these inputs, the team mapped the variants in the variation map as seen in Fig. 3. By using the similarity assessments and the decision framework of DDM, the team merged some variants in the variation matrix [12]. As a result, four variants for the *Initiate Project* subprocess, five variants of the *Plan Project* subprocess, and four variants of the *Close Project* subprocess were identified. No variants were identified for the subprocesses of *Analyze and Design*, *Implement*, *Test*, and *Create Asset* activities at this level of decomposition. This does not mean that all variants are performed in the same way for these activities. Although the variants are similar on this level of decomposition, there may be variations in deeper levels of the hierarchy.

4.7 Step 7: Configure a specific process variant

The generated variation map acts as a reference model to observe both the process map and help the experts to arrive at possible variations by means of the flow defined by gateways. This model does not include knowledge of a specific variant. Thus, if one wants to configure a process variant, she needs to understand that specific variant and go through the variation map to select relevant activities. This selection was highlighted for the variant of Company 4-1 as shown with darker colored activities in Fig. 3. The process of Company 4-1 starts with the *"Moderate Initiation"* variant of the *Initiate Project* subprocess (this is shown as *"Detailed Initiation"* in Table 3, but these variants were assessed to

Table 4 Variants of each subprocess in Table 3 and related activities (in the cells)

No	Variant category	Initiate Project	Plan Project	Analyze and Design	Implement	Test	Close Project	Create Asset
0	-	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	-Create Asset
1	Basic	-Initiate project	-Plan resources	-Analyze and design	-Implement	-Perform user acceptance test	-Close project	(N/A)
2	Fast	(N/A)	-Plan resources -Develop schedule	(N/A)	(N/A)	(N/A)	-Monitor and control -Close	(N/A)
3	Simple	-Initiate project -Assign PM	-Plan resources -Approve plan	(N/A)	(N/A)	(N/A)	-Approve closure -Close project	(N/A)
4	Moder-ate	-Initiate project -Approve initiation -Approve scope	-Plan resources -Control plan quality -Approve plan	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)
5	Detailed	-Initiate project -Approve scope -Announce initiation	-Plan resources -Prepare Work Breakdown Structure (WBS) -Approve plan	-Analyze and design -Control design quality	-Implement -Control implementation quality	-Perform user acceptance test -Control test quality	-Prepare closure report -Approve closure Close project	(N/A)
6	Complex	-Assign PM -Initiate project -Initiate quality control -Approve scope	-Plan resources -Prepare WBS -Control plan quality -Approve plan	(N/A)	(N/A)	(N/A)	-Monitor and control -Control closure quality -Close project	(N/A)

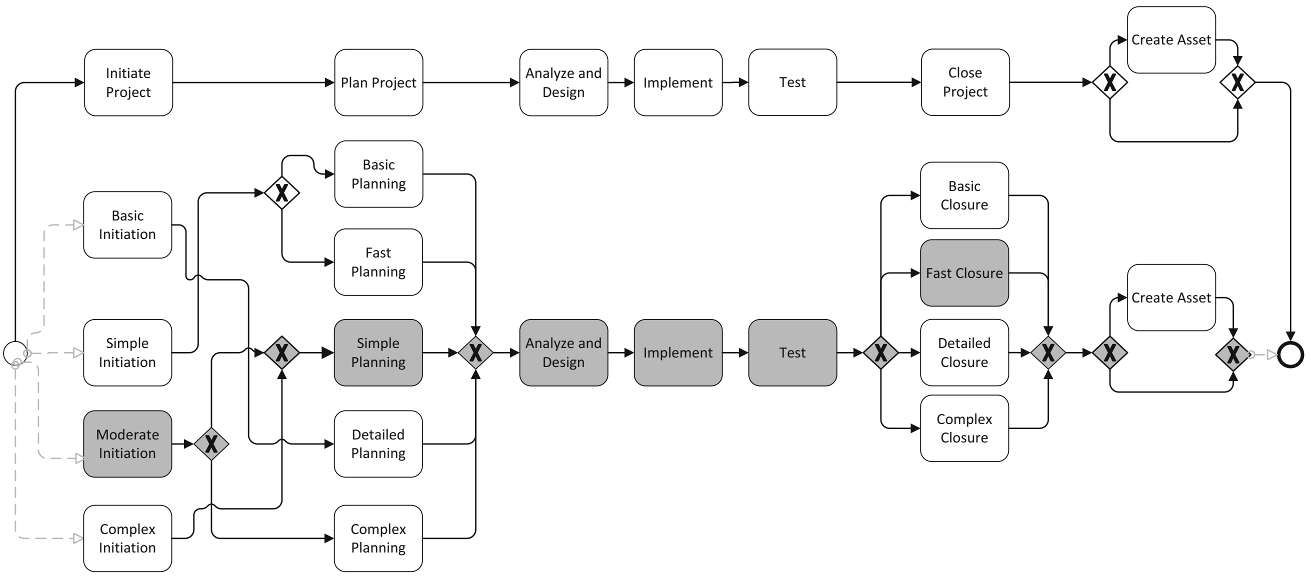


Fig. 3 Variation map for SPM process (refined from [14])

be similar in Table 4). As specified by the variation matrix (Table 3), *"Moderate Initiation"* is either followed by the *"Simple Planning"* (for Company 4-1), or *"Complex Planning"* (for Company 2) variant. This is represented by an exclusive gateway after the activity symbol representing the *"Moderate Initiation"* variant in the variation map. The team manually verified that all variants could be generated as syntactically correct and sound [44].

The variation map helped the team to find out patterns about the process variants. For example, the team observed that *"Simple Initiation"* may be followed by either *"Basic Planning"* and *"Fast Planning"*. As 4S develops variant models for a higher number of customers in the future, such patterns may support the analysts to discover more relations between variants and perform deeper analysis on the processes.

4.8 Step 8: Apply DDM for a subprocess

DDM by definition builds on the construction of a hierarchical structure to define the family of process variants [12]. In this way, the approach provides a mechanism to manage process variants in a process repository by applying the same steps for the subprocesses identified in the variation map in Fig. 3. It is reported that DDM was applied within a hierarchical structure, but an example of a subprocess variant model was not presented [12]. To apply the approach in a hierarchical structure in conformance with the needs of 4S, the team decided to focus on the variations in the subprocesses of the *Plan Project* activity. This activity was selected because from a variation management perspective, it was the most complex one due to the high number of variants and number of subprocesses with respect to the others. Hence, this subprocess was evaluated to provide the most business value to 4S. Here, to exemplify the application and the problem encountered, we focus on the two variants among five variants of the *Plan Project* subprocess, namely *"Detailed Planning"*, and *"Complex Planning"* variants.

DDM explains a subprocess variant modeling mechanism for activities that does not have variants [12]. Examples of such activities in our case are the *Analyze and Design*, *Implement*, and *Test* activities in the variation map (Fig. 3). There are no variants of the subprocesses of these activities, according to the analysis in Table 4. However, DDM does not explicitly specify a solution for a case where an activity has variation in its subprocesses. In our case, namely, *Initiate Project*, *Plan Project*, and *Close Project* activities are examples of this case because they have more than one variant subprocess.

The two variants that we use as examples, *"Detailed Planning"* and *"Complex Planning"*, have some common activities, for example *Prepare WBS*, that may also have variations. These variations may be the same or different for *"Detailed Planning"* and *"Complex Planning"* subprocesses. The point not specifically addressed by DDM is how to analyze variant subprocesses of the same activity in SPM process (i.e. *Plan Project*) that share common activities, while also considering variations in those common activities (i.e. *Prepare WBS*). To handle this case based on our selected subprocess, the team identified three alternative solutions:

1. *Move the common activity in the variant subprocesses to the main variation map* In our case, one can move the *Prepare WBS* activity to SPM variation map (Fig. 3), and remove this activity from the definition of related variants of the *Plan Project* subprocess. In this way, the variants of the *Prepare WBS* activity can be managed from this single point of reference. However, this activity is not on the relevant granularity level for SPM process. It does not represent a milestone in this process as is the case for the other activities. If the same situation arises for other activities in the repository and on other decomposition levels, the variation maps may become overloaded with activities of different granularity levels. Additionally, the order of activities cannot be modeled correctly within the relevant variant subprocess of the *Plan Project* activity.
2. *Develop separate variation maps for the common activity in variant subprocesses* Another solution is to handle the variations of the *WBS preparation* subprocess in two separate variation

   maps. In this case, two variation maps would be created for the *"WBS Preparation"* subprocess, one to be referred to from the *"Detailed Planning"* and the other from the *"Complex Planning"* subprocess. In this way, variations of the *"WBS Preparation"* subprocess specific to the *"Detailed Planning"* and *"Complex Planning"* subprocesses would be shown explicitly. However, then, multiple models need to be maintained for the variants of the same subprocess, which would increase the maintenance effort. Moreover, the problem would be propagated to common activities shared by the variant subprocesses of the *Prepare WBS* activity.
3. *Develop a common variation map for the subprocess* The variations of the *WBS Preparation* subprocess may be managed in a single variation map which is referred to from all relevant variant subprocesses (i.e. *"Detailed Planning"* and *"Complex Planning"*). In this solution, both the *"Detailed Planning"* and *"Complex Planning"* variants of the *Plan Project* activity would have their own subprocess definitions containing *Prepare WBS* activity, and refer to the same variation map of *WBS Preparation* subprocess. In such a solution, one cannot observe from the variation map of, for example, the *"Detailed Planning"* subprocess, which variations of "WBS Preparation" are related to that. However, all process variants are managed in a single model, and similar approach can be propagated to even further decomposition of subprocesses.

For 4S, the team decided to apply the third solution. Thus, the team developed a single variation map and a variation matrix for the variants of the *WBS preparation* subprocess. The business drivers were identified as *company*, and the variants as *"Company 2"* and *"Company 5"* as imposed by the SPM variation map (Fig. 3). Moreover, the team identified one more driver applicable to this decomposition level: *project type* which can be "*new project*", or "*configuration project (conf. project)*". This driver is relevant for both company variants. The emergence of a new driver made the team realize that different drivers may be in place on different decomposition levels of a process repository. The team identified the variations of the *WBS preparation* subprocess in

Table 5 Variation matrix for WBS preparation subprocesses

	High-level project planning (PP)	Project categorization	Process appropriateness evaluation (PAE) polling decision	PAE polling	Software project management plan (SPMP) preparation	WBS control	SPMP review
Company 2 New project	Simple high-level PP	Project categorization	PAE polling decision	Detailed polling	Simple SPMP preparation	WBS control	SPMP review
Company 2 Conf. project	Simple high-level PP	Project categorization	(None)	(None)	Simple SPMP preparation	WBS control	SPMP review
Company 5 New project	Detailed high-level PP	Project categorization	PAE Polling decision	Complex polling	Detailed SPMP preparation	WBS control	SPMP review
Company 5 Conf. project	Detailed high-level PP	Project categorization	(None)	(None)	Detailed SPMP preparation	WBS control	SPMP review

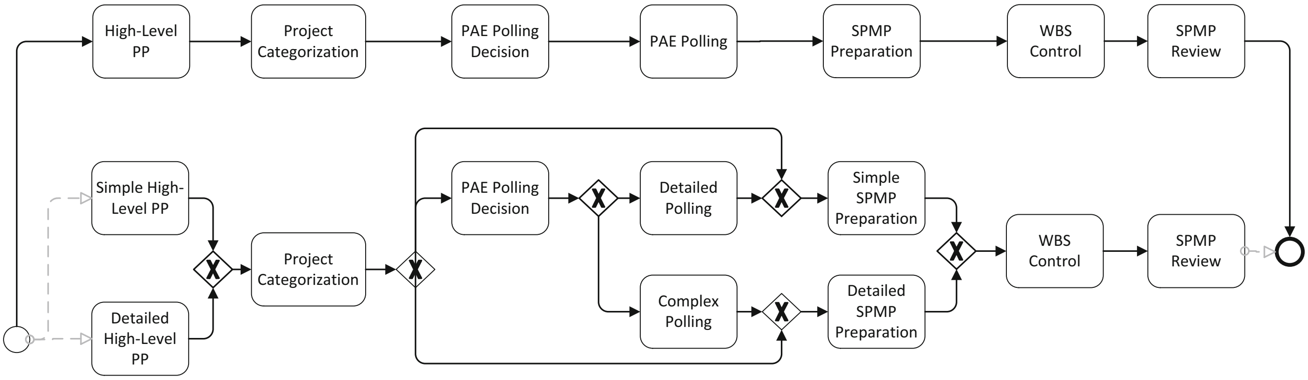


Fig. 4 Variation map for WBS preparation subprocess

the variation matrix in Table 5. It was noted that, for some variants, there is no applicable subprocess such as for the second and fourth variants of the *PAE polling* activity. Based on the variation matrix, the team developed the variation map in Fig. 4 for this subprocess.

In the next section, we describe how the team applied the Provop approach to SPM process, and to one immediate subprocess in the process hierarchy.

5 Applying the Provop approach

Provop focuses on creating a single model called a base process, which includes adjustment points and their related sets of options [41]. The options include a set of atomic operations such as insert, delete, move and modify which are used to configure the base process to reach a certain variant. The options provide a reusability mechanism to define common operations for multiple variants. This mechanism decreases the complexity and increases the controllability to configure a variant. Moreover, Provop can support automated variant configuration by defining context-aware configuration options.

5.1 Step 1: Design a base process

Provop offers different policies to identify the base process from which the process variants are configured. The policies show alternative approaches to design the base process. After the base process is created, it is then enriched with adjustment points. Considering the variations of a process, the policies that can be followed to identify the base process are as follows. One can either (1) use the standard reference process used within the particular industry, (2) use the most frequent process variant, (3) design a version that has minimal average distance to all variants, (4) create a superset or (5) intersection of all process variants. The first two policies are different from the others in the sense that a base process designed according to one of these policies are valid processes and represent a specific use case, while in other cases the base process may not be semantically valid [41]. The selection of the policy depends on the context and purpose of variant management, and it is not mandatory to stick to one policy in the application of Provop [45]. In our case, the team applied a combination of these policies. First, the standard PMBOK reference model is taken as the starting point (policy 1). Next, the team extended this model by consider-

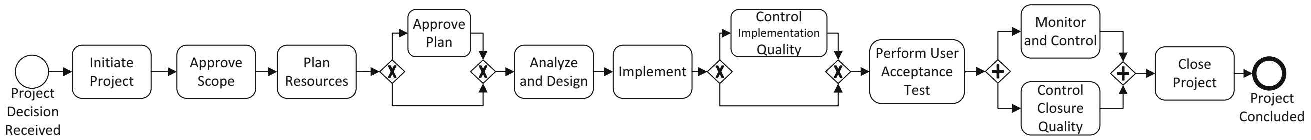


Fig. 5 Final SPM base process (refined from [14])

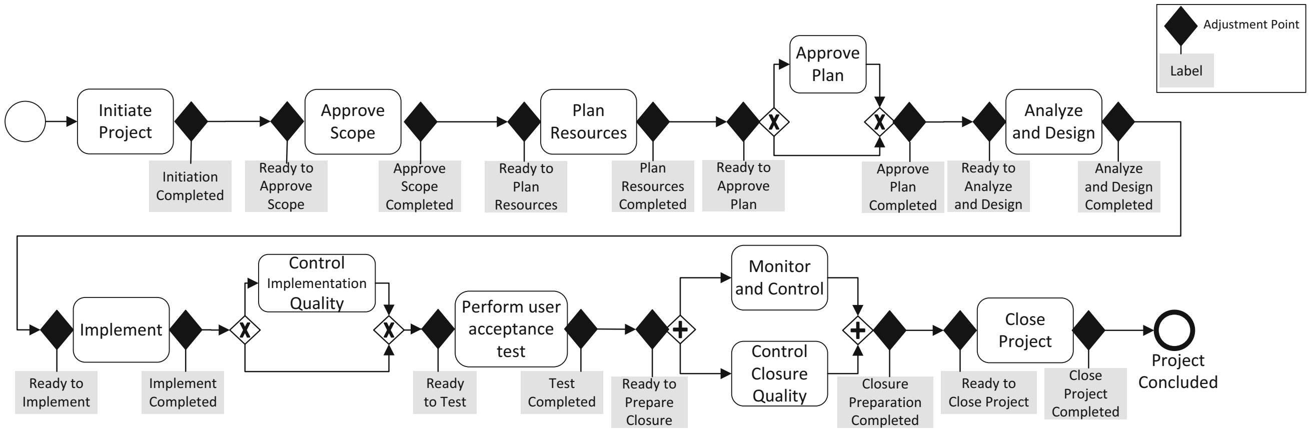
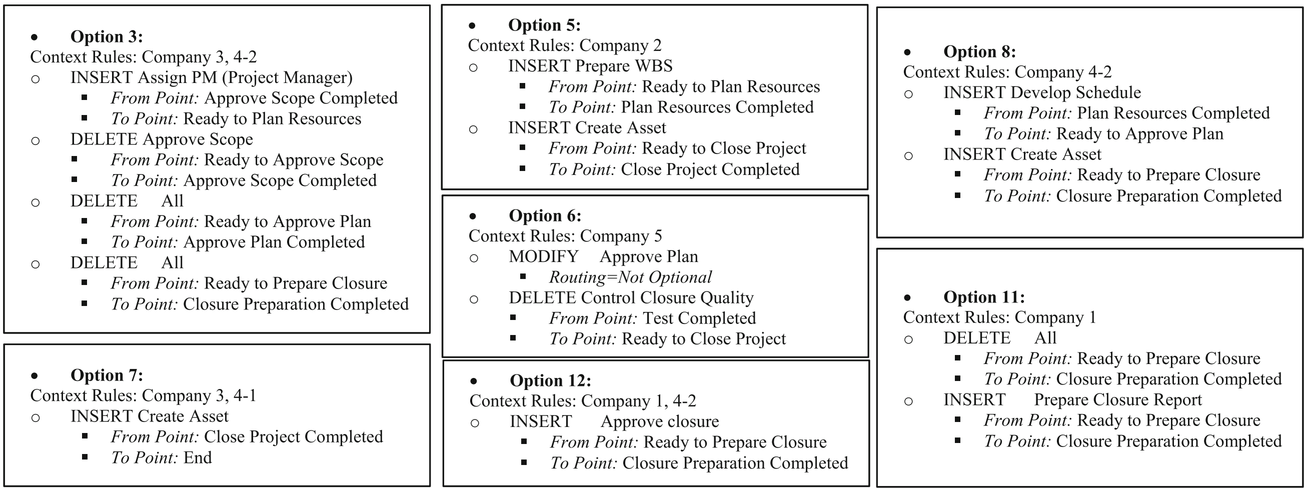


Fig. 6 SPM base process with adjustment points (refined from [14])



- **Option 3:**
  Context Rules: Company 3, 4-2
  - INSERT Assign PM (Project Manager)
    - *From Point:* Approve Scope Completed
    - *To Point:* Ready to Plan Resources
  - DELETE Approve Scope
    - *From Point:* Ready to Approve Scope
    - *To Point:* Approve Scope Completed
  - DELETE All
    - *From Point:* Ready to Approve Plan
    - *To Point:* Approve Plan Completed
  - DELETE All
    - *From Point:* Ready to Prepare Closure
    - *To Point:* Closure Preparation Completed

- **Option 7:**
  Context Rules: Company 3, 4-1
  - INSERT Create Asset
    - *From Point:* Close Project Completed
    - *To Point:* End

- **Option 5:**
  Context Rules: Company 2
  - INSERT Prepare WBS
    - *From Point:* Ready to Plan Resources
    - *To Point:* Plan Resources Completed
  - INSERT Create Asset
    - *From Point:* Ready to Close Project
    - *To Point:* Close Project Completed

- **Option 6:**
  Context Rules: Company 5
  - MODIFY Approve Plan
    - *Routing=Not Optional*
  - DELETE Control Closure Quality
    - *From Point:* Test Completed
    - *To Point:* Ready to Close Project

- **Option 12:**
  Context Rules: Company 1, 4-2
  - INSERT Approve closure
    - *From Point:* Ready to Prepare Closure
    - *To Point:* Closure Preparation Completed

- **Option 8:**
  Context Rules: Company 4-2
  - INSERT Develop Schedule
    - *From Point:* Plan Resources Completed
    - *To Point:* Ready to Approve Plan
  - INSERT Create Asset
    - *From Point:* Ready to Prepare Closure
    - *To Point:* Closure Preparation Completed

- **Option 11:**
  Context Rules: Company 1
  - DELETE All
    - *From Point:* Ready to Prepare Closure
    - *To Point:* Closure Preparation Completed
  - INSERT Prepare Closure Report
    - *From Point:* Ready to Prepare Closure
    - *To Point:* Closure Preparation Completed

Fig. 7 Example options from the option list of base SPM process (refined from [14])

ation of policy 2, that is the variant of Company 1 which is the most frequent process worked on in 4S. The team utilized policy 3 to identify process elements so that it will require the least number of operations in total to reach process variants, while the team also included activities at the intersection of all variants as suggested by policy 4. As a result, the initial version of the base process evolved from the best practice model in Fig. 1 to the version prepared at the end of the first iteration of the research cycle [14]. The team found this base process to be simple but still to provide an overview of the key activities observed in most of the variants. However, due to the issues encountered in designing the options as described in Sect. 5.3 below, the team updated the initial base process to the final one in Fig. 5 in the second iteration. Based on the different base process, the application of Provop was updated as seen in Figs. 6 and 7.

5.2 Step 2: Define adjustment points

The next step of Provop is to determine the explicit positions of the adjustment points that specify where the options can be applied on the base process. In this step, the team analyzed the base process and identified the adjustment points necessary to be able to generate all process variants. Initially, the team placed an adjustment point at the start and end of all activities and single-entry-single-exit (SESE) fragments ([46]) on the main flow. Later, the team removed the unnecessary adjustment points based on the option design as described in Sect. 5.3 below. The final model with adjustment points can be seen in Fig. 6.

5.3 Step 3: Design and model the options

In this step, options are designed so that the process variants can be derived from the base process. For this purpose, Provop offers four different operations: “insert”, “delete”, “move”, and “modify”. The first three operations are on changing the control flow of the process, while the “modify” operation is for changing the properties of process elements. With the “insert” and “delete” operations, it is possible to add or remove process fragments between the adjustment points [41]. The team initially developed the base process in [14]. However, further investigation of the design revealed some issues, which made the team decide to update the base process to the one in Fig. 5 and design the options exemplified in Fig. 7. The issues encountered and our solutions are described below.

- *Deleting process fragments* In some cases, the team needed to “delete” a process fragment rather than an individual element. For example, it was necessary to delete the whole block structure while removing the optional *Control Implementation Quality* activity in Fig. 6. In this case, as the process fragment to be deleted already existed in the base process, it was not necessary to model the fragment in the option list. The team used the keyword “*All*” to imply that the fragment between the start and end adjustment points is to be removed completely in a delete operation. This approach was used in options 3, 6, and 11 in Fig. 7. Although this simplified the option design, the experts found it harder to read the option list as they had to find the exact point in the base process to find out the elements to be deleted.
- *Removing routing elements* The team encountered another limitation in updating the control flow while changing an optional activity to a fixed one in the sequence flow. As an example, in one of the variants, the *Approve Plan* activity was not optional, although it is an optional activity in the base process (Fig. 6). A possible solution was to apply “delete” operation on the related SESE fragment, then insert the activity back. Due to readability concerns, the team did not prefer to follow this approach. Instead, the team designed the option by using the “modify” operation to change the activity property as “not optional”. This approach was used in option 6 in Fig. 7.
- *Possible conflicts during configuration* The team found option design a challenging and error-prone task. The team had to evaluate possible conflicts that could emerge as a result of applying operations. For example, the last operation of Option 3 could be defined “*From Point: Test Completed, To Point: Ready to Close Project*”, instead of the current design. Applying the operation in this way would result in the deletion of the two adjustment points: ”Ready to Prepare Closure”, and “Closure Preparation Completed”. While Provop does not clarify whether deletions of adjustment points are allowed in the options, such a case requires careful revision of other options to ensure that they do not use the deleted adjustment points.
- *Storing process fragments* The “insert” operation can be used to add more than one process element at a time. To use the “insert” operation in this way, the team had to store the fragments of process models as part of the option design, rather than as part of the process model repository. The definition of process fragments increases maintenance efforts as these fragments need to be stored and managed in addition to the base process itself, and changes need to be propagated to all variants [9]. Moreover, Provop suggests the use of process fragments in operations applicable to multiple variants [45]. However, the team observed that it is possible to design operations on fragments which are applicable to only a single variant, or a fragment may turn out to be applicable to only one variant in time. For example, considering the “insert” operations in option 11 and 12 in Fig. 7, a process fragment including the activities *Prepare Closure Report* and *Approve Closure* could be defined. This fragment would only be used by the variant *Company 1*. To avoid these two challenges, maintaining additional process information in the form of process fragments and defining fragments that are not reused by different variants, the team decided to “insert” a single process element at a time in an operation. In this way, the labels of activities were sufficient to design the options rather than “models of process fragments”.

  The two challenges below emerged due to the decision of the team not to use process fragments.
- *Inserting optional activities* Provop provides a mechanism to add parallel activities while configuring variants. With an “insert” operation, a process element is added in parallel to the process fragment between the start and end adjustment points [47]. However, the approach does not specify a technique to insert an optional activity which is not part of a process fragment. With the decision to avoid the insertion of process fragments, the team had no mechanism to insert optional (or exclusive) activities in the control flow. To solve this, the team had to place the optional activities of the variants in the base process. This resulted in the addition of optional *Approve Plan* and *Control Implementation Quality* activities in Fig. 5.

- *Inserting consecutive activities* In relation with the problem described above, since the team decided not to use the process fragment insertion feature of Provop, another technique to add multiple individual activities in a sequence was not available. For this reason, the team designed the base process so that no variant required the addition of multiple consecutive activities within the same start and end adjustment points.

As mentioned above, the team found that the design of the options is a challenging task, but it can also support the analysts to discover detailed control-flow properties of the variants. Granularity of options refers to the number of operations combined to define the options. It is an important factor in option design to enhance reusability and maintainability of options while keeping the number of options minimal [13]. As Provop suggests, two policies can be followed to group the operations into options: (1) to design options consisting of one operation and reusing them for multiple variants; (2) to create one option for each process variant that contains all operations for that variant. In practice, it is necessary to find a balance between these two policies by keeping the number of options low while increasing reusability. In 4S, the team first identified the minimum number of operations necessary to create the variants and the context rules for them. The context rules identify, for each option, the variant(s) that option is to be applied. Then, based on the context rules, the team started merging the operations. The team focused on increasing the reusability of options among the process variants, thus preferring low granularity as favored by policy one. The team preferred this approach because in this way, they were able to use the option list to observe the similarities in different process variants. However, it was possible to combine operations in various other ways to define the options. An alternative design for our options could be as follows. The last "delete" operation of Option 3 and the "delete" operation of Option 11 are the same. The team could take out these operations from these options and create another option applicable for *Company 1, 3,* and *4-2*. Although this would create an option reused by three variants, it would increase the number of options in total and decrease the number of reused operations in Option 3. Provop provides hardly any support to discover and choose between such alternative option designs.

The effort given to design the options enabled the team to discover interesting characteristics of control-flow variations. For example, three options were defined that include the insertion of the *Create Asset* activity (In Option 5, 7, and 8). They could not be merged because they were based on different start and end adjustment points. Thus, even when the activities looked similar in process variants, there were significant differences in how they were performed.

5.4 Step 4: Configure variants

For variant configuration, Provop suggests the use of three substeps. First, relevant options need to be selected to configure a process variant. This can be done by asking users to manually choose specific variants, which is hard if there are a plethora of options and specialized knowledge is required. To overcome the problem, Provop suggests the definition of context rules by identifying, for each option, the context in which the options are applicable. In our case, the six SPM process variations of five 4S customers identified at the beginning of the study (Table 1) was used to define the context. For each option, the team identified the set of variants that are to be configured via this option. This can be seen in Fig. 7 as context rules. Another point to be considered while applying the options is the possible constraints with the options. For example, there may be implication relations between options, an option implying the usage of another one [13]. It was observed that the modelers need to pay special attention for constraints especially for options effective on the same adjustment point pairs.

In conformance with the constraints, the team applied the relevant set of options (exemplified in Fig. 7) to the base process on Fig. 6 to achieve the variant processes of Company 1 and 2 as can be seen in Fig. 8.

5.5 Step 5: Apply Provop for a subprocess

After the definition of the base process with adjustment points for the SPM process, the team needed to analyze the variants for subprocesses of the activities in the base process. The team was not able to find any guidelines for applying Provop in a hierarchical process structure in the literature. The team decided to review the base process for SPM process, and apply the steps of the approach in the same way for the subprocesses of this process. The team developed a base process *WBS preparation* subprocess, which we call low-level base process, that can be seen in Fig. 9. Example options from the option list for this base process are shown in Fig. 10. The team encountered the following issues during the application of Provop for the subprocesses and handled them as described below.

- *Invisible activities* For some variants, some activities were invisible in the base process because they are added through an "insert" operation. *Prepare WBS* activity is an example of such a case. When the team developed the base process for the related *WBS preparation* subprocess, there was no possibility to refer to this model from the SPM base process. This caused this subprocess to be "lost" in the variant process model hierarchy structure. To solve this problem in the case of *WBS preparation* base

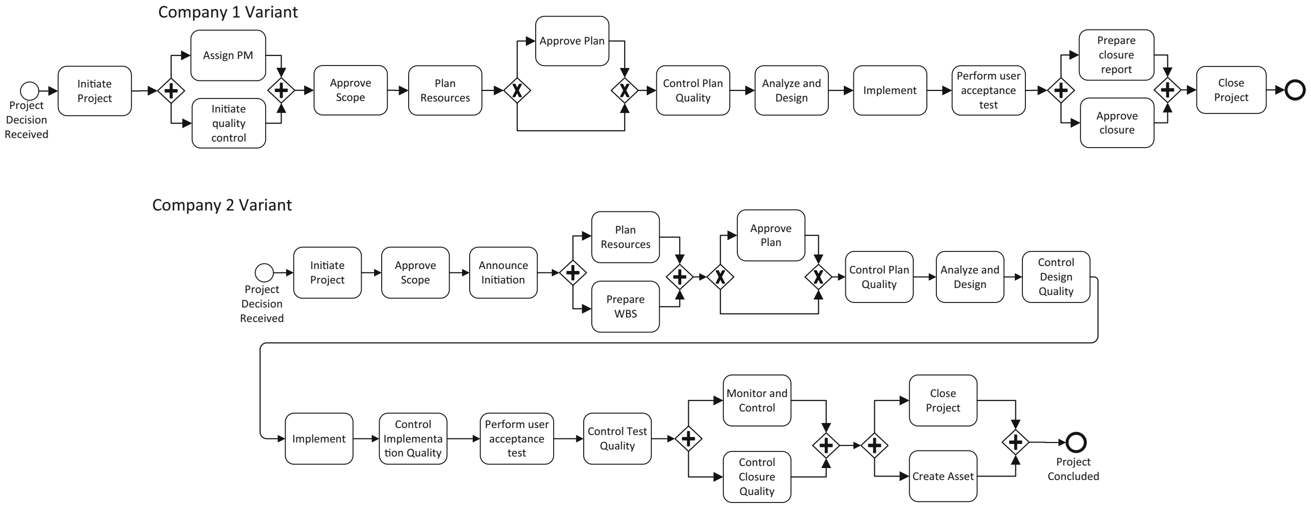


Fig. 8 Process models for company 1 (top) and 2 (bottom) variants after configuration

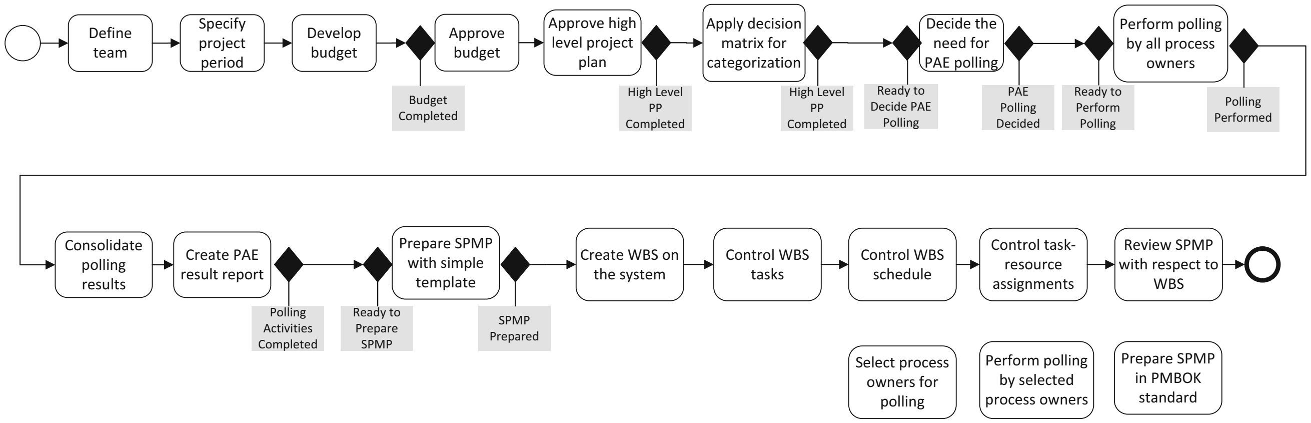
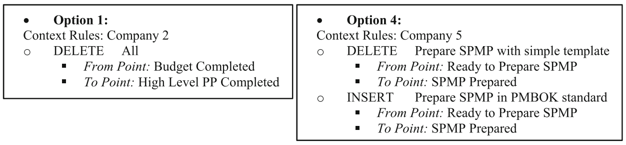

Define team
Specify project period
Develop budget
Budget Completed
Approve budget
Approve high level project plan
High Level PP Completed
Apply decision matrix for categorization
High Level PP Completed
Ready to Decide PAE Polling
Decide the need for PAE polling
PAE Polling Decided
Ready to Perform Polling
Perform polling by all process owners
Polling Performed
Consolidate polling results
Create PAE result report
Polling Activities Completed
Ready to Prepare SPMP
Prepare SPMP with simple template
SPMP Prepared
Create WBS on the system
Control WBS tasks
Control WBS schedule
Control task-resource assignments
Review SPMP with respect to WBS
Select process owners for polling
Perform polling by selected process owners
Prepare SPMP in PMBOK standard

Fig. 9 WBS preparation base process with adjustment points


- **Option 1:**
  Context Rules: Company 2
  - ○ DELETE All
    - ▪ *From Point:* Budget Completed
    - ▪ *To Point:* High Level PP Completed

- **Option 4:**
  Context Rules: Company 5
  - ○ DELETE Prepare SPMP with simple template
    - ▪ *From Point:* Ready to Prepare SPMP
    - ▪ *To Point:* SPMP Prepared
  - ○ INSERT Prepare SPMP in PMBOK standard
    - ▪ *From Point:* Ready to Prepare SPMP
    - ▪ *To Point:* SPMP Prepared

Fig. 10 Example options from the option list of WBS preparation base process

process, the team placed "standalone" activity symbols for the activities that are inserted to the base process in Fig. 9. SPM base process was also updated accordingly (which is not shown in Fig. 6).

– *Mismatching adjustment points* Mismatches can occur between the adjustment points and events of base processes in different levels, as the same activity can be inserted between different adjustment points in different variants. For example *Prepare WBS* activity can be inserted either parallel to *Plan Resources* activity, or after it. In both cases, different start and end adjustment points will be applicable. To avoid conflicts with the low-level base process, the team did not reuse adjustment points and did not name start and end events in the low-level base process.

The application of the two approaches in Sects. 4 and 5 on the selected processes not only provides a high business value for 4S due to the processes' organizational importance but also academic value as these processes have a higher number of activities and gateways than current examples in the literature [12,13,41,42]. In the following section, we present our findings from applying the two approaches and provide a set of guidelines for process variant modeling approach selection.

6 Findings

The application of the two selected process variant modeling approaches, namely DDM and Provop, in our action study provided various insights about both the benefits of using a

Table 6 A comparison of the approach characteristics and outputs produced (refined from [14])

	DDM	Provop
Inherent type	Variability by restriction	Variability by restriction and extension
Modeling technique	Single artifact	Multi-artifact
Produced outputs	Variation map	Base process Option list with context rules
Metrics for main SPM process	24 activities, 10 XOR gateways, and 45 edges	11 activities, 6 XOR gateways, and 20 edges 11 activities inserted by 14 options 17 adjustment points
Metrics for subprocess	17 activities, 6 XOR gateways, 29 edges	16 activities and 17 edges 3 activities inserted by 4 options 10 adjustment points

Table 7 Return on investment (ROI) for DDM and Provop calculated based on yearly and one-time variant management costs

	Current (actual)	DDM (estimated)	Provop (estimated)
Yearly effort (hours)			
Variant creation	2952	1515	1063
Maintenance	2117	1870	1552
One-time effort (hours)			
Transformation	0	4886	4067
Yearly cost (TL)			
Variant creation and transformation	253.440	169.260	130.752
One-time cost (TL)			
Transformation	0	244.320	203.360
Return on investment (years)		2.9	1.7

variant modeling approach instead of following standard process modeling approaches and the challenges related to these approaches. Table 6 provides a comparison of the characteristics of the approaches and the resulting outputs. In total, the team spent 28 h in seven sessions on applying DDM, whereas 23 h in six sessions were needed for applying Provop.

In the rest of this section, we provide our findings from the application of the two selected process variant modeling approaches. This includes the evaluation of the business value of applying a variant management approach in Sect. 6.1, and evaluation of the two approaches against the identified user needs and requirements in Sect. 6.2. Lastly, we present the selection guidelines identified based on these evaluations in Sect. 6.3.

6.1 Evaluation of the value of applying a variant modeling approach

The team evaluated the business value to be achieved by switching from the current way of process modeling to a specific process model variant management approach. For this purpose, the team calculated the expected Return on Investment (ROI) for DDM and Provop, as presented in Table 7. We describe the details of this table below.

Variant creation 4S works on average with nine customers per year, two of them being added as a new customer, and seven being existing customers. For these existing customers, only 30% of the effort is spent on defining new processes. 4S analysts works on around eight different process areas per customer for which variant modeling is performed. Around 90 h of effort are spent by 4S analysts for creating a new variant. As a result, $(2 + 7 \cdot 0.3) \cdot 8 \cdot 90 = 2952$ h are spend in total for variant creation.

For both DDM and Provop, a reduction is expected in the variant creation time. For DDM, a reduction of 49% is expected. For Provop, a reduction of 64% is expected.

Maintenance As mentioned, 4S works with nine customers per year. Two of these are new and do not require any maintenance. For the seven existing customers, 30% of the effort is spent on variant creation. The remaining 70% are on maintenance. As mentioned, 4S works on eight different process areas. 4S analysts spend on average 54 h of effort per process area. As a result, $(7 \cdot 0.7) \cdot 8 \cdot 54 = 2116.8$ h are spent in total on maintenance per year.

Similar to the variant creation, also in the maintenance of variants a reduction in effort is expected by the team. For DDM, this reduction is expected to be 12%. For Provop, a reduction of 27% is expected.

Transformation For transforming from the current way of working to either DDM or Provop, two kinds of costs are involved: (1) transforming the current set of variants, and (2) training the analysts of 4S to become proficient with either DDM or Provop.

In the past five years, 4S has worked with 17 customers. For each customer, a variant has been created for each of the aforementioned eight process areas. For DDM, it is expected that, on average, the transformation of one variant takes around $32,4$ h. For Provop, this is $25,2$ h. Currently 20 analysts are employed by 4S. 4S expects that a total of 24 and 32 training hours are needed for DDM and Provop respectively. In total, the transformation for DDM is $(17 \cdot 8 \cdot 32,4) + (20 \cdot 24) = 4886,4$ h and for Provop, this is $(17 \cdot 8 \cdot 25,2) + (20 \cdot 32) = 4067,2$ h.

Costs and return on investment The costs are calculated based on an hourly wage of 50 Turkish Lira (TL), which is about € 12,50. Considering the yearly savings to be obtained in variant creation and maintenance activities, it is calculated that by using DDM, 4S will start to gain benefits in 2.9 years for the investment of transformation costs, whereas this number is 1.7 years for Provop. Although this calculation is specific to 4S and all values are prone to change in specific cases, it shows that considerable business value can be obtained by organizations in short-term by applying any process variant management approach.

6.2 Evaluation of the approaches with respect to user needs and requirements

In this section, we evaluate the application of the two approaches with respect to the user needs identified in Sect. 2.2 and the requirements specified in Sect. 3.1. We drop three of the requirements, namely Req. 1 domain independence, Req. 2 conceptual process type, and Req. 4 validation, since the definition of the approaches already supports these requirements and there were no further arguments by the team. We keep the four remaining requirements, control-flow (Req. 3), hierarchy (Req. 5), process modeling language (Req. 6), and decision support (Req. 7) that deserve further discussion about the extent the approaches satisfy them. Table 8 provides a summary of the evaluation. In this table, a benefit and/or advantage provided by an approach is indicated by a "+" sign, whereas a challenge is expressed by a "−" sign before the item. We further elaborate on each user need/requirement below.

User Need 1 Establishing a knowledge-base Both Provop and DDM were observed to provide a knowledge-base by combining the process knowledge from multiple customers in an integrated model. This was found to be an improvement with respect to the current way of working where 4S analysts developed and maintained process models of each customer separately. The team found it relatively easier to read Provop's base process as a reference. The base process provided essential information about the process without the need to use the option list. Moreover, the option list proved to be an effective tool to discover detailed similarities between the variants by means of operations grouped for multiple variants. Other interesting characteristics of control-flow variations were also realized by means of the option list. For example, although three variants included the *create asset* activity, each of them used this activity in the control-flow in a different way. Thus, three different operations had to be defined for inserting the same activity in different ways. The team found it non-intuitive to interpret the variation map of DDM, so it was hard to use it as a knowledge-base. However, main variation map included high-level activities, and provided a more summarized view that was detailed through lower levels.

In conformance with this evaluation, the team estimated a drop in the variant creation activities for both approaches, but more for Provop as seen in Table 7. Such an effort saving is foreseen since the analysts can use the created models as a knowledge-base while developing new process models.

User Need 2 Maintenance Since both approaches provided an integrated environment that facilitates the impact analysis for changes, the team evaluated that maintenance effort to be lowered with respect to the current status. Maintaining DDM variation maps was evaluated to be harder with respect to Provop's base process as there may be multiple subprocesses defined for variants of the same activity. A generic change may require the analysts to check each subprocess and apply the change in a consistent way. The team tried to alleviate this issue by applying their solution to integrate subprocess variant models that are referenced from multiple processes. Provop outputs were found easier to maintain than the outputs of DDM as changes can be applied on a single model. Provop was found to facilitate the maintenance of specific changes in addition to generic ones due to this fact. However, special attention needs to be given to maintain consistency with the option list. Accordingly, the team estimated the maintenance effort to be lowered for both approaches and more for Provop (Table 7).

User Need 3 Reuse By providing an integrated knowledge-base from which similar variants can be looked up and a new variant can be configured, the team evaluated that both approaches can facilitate systematic reuse of process knowledge in comparison with the current approach. The team found DDM's variation map more flexible than the Provop's base process for defining a new process, as they can see all alternatives together with the constraints on the map. Business drivers helped the team to identify potentially similar variants in the existing models. However, Provop was found to be more practical for configuration-based reuse as described in the *decision support* item below. The support for

Table 8 Comparison of the approaches based on the user needs and requirements

User need/requirement	Approach	
	DDM	Provop
User Need 1 Establishing a knowledge-base	+ Both approaches prove helpful by combining multiple variants in an integrated model	
	+ Main variation map provides an overall view	+ Base process is easy to read and use
		+ Detailed similarities are stored by means of the option list
User Need 2 Maintenance	+ Impact analysis of changes are facilitated by means of integrated models	
	− A generic change may require the update of multiple models	+ Both generic and specific changes may be eased
		− Maintenance of the option list requires extra care.
User Need 3 Reuse	+ Process knowledge can be used systematically due to integrated knowledge-base	
	+ Variation map enables the observation of all alternatives together	+ Particularly practical for configuration-based reuse
	+ Business and syntactic drivers are useful to identify similar variants	
User Need 4 Variability in a hierarchical structure and Requirement 5 Hierarchy	− Challenges arise due to the use of variant modeling approaches	
	− Subprocesses that are used by multiple process variants require a special solution.	− Activities invisible in the base process require a special solution.
		− Extra care is required to keep adjustment points consistent among process levels
Requirement 6 Process modeling language	+ Both approaches build on BPMN	
	+ Only native BPMN constructs are used	− Additional constructs need to be used
		+ Complexity of the resulting process models is low
		− BPMN modeling rules need to be tailored
Requirement 3 Control-flow	+ Both approaches provide techniques to manage control-flow variability	
	+ Representation of variants as distinct subprocesses provide flexibility	+ The approach enables the analysis of control-flow differences in detail
		− The approach does not provide rules to deal with some control-flow-related issues
Requirement 7 Decision support	+ Both approaches provide some sort of decision-support mechanisms	
	+ Drivers provide good support to understand business context	− Option list identification is an ambiguous and challenging task
	− Specialized knowledge of specific variants is needed for configuration	+ Option list, once defined, makes it easy to configure a variant

variant modeling of hierarchical processes was found to be another aspect that affected reuse, which is discussed below. The expected improvements on the reuse were reflected in the effort estimation for variant creation in Table 7.

User Need 4 Variability in a hierarchical structure and Requirement 5 Hierarchy This user need emerged with the implementation of a process variant management approach, since BPMN natively supports the organization of process models in a hierarchical structure. The team evaluated that the application of both DDM and Provop requires additional consideration and effort for managing hierarchical process models. Although DDM by definition supports variant modeling in hierarchies, in practice the team encountered a problem to develop variant models for subprocesses that are used by multiple process variants. Among the solutions identified, the team applied the one which increased the reusability on low-level process variants. For Provop, the team could apply the native subprocess definition mechanism of BPMN. In this case, the team realized that they need to represent activities that are invisible in the base process because they are inserted by options. To establish a consistent hierarchical structure in the repository, the team placed such "standalone" activities in the base process. Moreover, another issue that arose with the application of Provop in a hierarchical structure is that, special attention must be paid to the consistency of adjustment points and events between different process levels.

Requirement 6 Process modeling language 4S preferred to use a variability management approach based on BPMN since it was the notation used in the company. This requirement was met by both approaches. However, for applying Provop, it was necessary to add specific symbols representing adjustment points on top of standard BPMN models. Moreover, the team had to tailor the use of BPMN for specific purposes (such as standalone activity symbols for invisible activities). In summary, Provop required some changes over standard BPMN notation that the analysts need to learn and deal with. On the other hand, standard BPMN constructs were sufficient to use DDM.

Based on the metrics in Table 6, Provop seems to produce process models with less complexity in terms of BPMN constructs. This may enhance the understandability of the approach in comparison with DDM [48]. However, there is an extra artifact, the list of options, required to read and customize the Provop base process. The use of an extra artifact can cause cognitive difficulty due to the split attention effect [34]. Nevertheless, the team indicated that it was easier for them to read the Provop's base process with respect to DDM's variation map, and "picturize" how the adjustments may be conducted even without seeing the option list.

Requirement 3 Control-flow The team found DDM relatively more flexible in defining control-flow variations, because the variants were developed as distinct subprocess models. The team experienced some limitations in modeling control-flow variability in Provop, such as inserting optional and consecutive activities, deleting process fragments, and dealing with possible configuration conflicts. The team had to extend and limit the application of the approach to resolve those limitations. An important limitation introduced by the team's decision was avoiding the use of the fragment insertion feature of Provop. While this decision may have increased the complexity of the base process, it alleviated the maintenance effort related to process fragments. In the end, as Provop required the team to handle control-flow variations in detail, it also enabled them to understand detailed differences between variants from this perspective. The transformation costs in Table 7 were estimated based on the characteristics discussed under this and the previous item (Req. 3 and 6). The overall transformation costs of Provop is estimated to be lower, whereas the training costs are foreseen to be higher.

Requirement 7 Decision support We evaluate the decision support provided by the approaches in two dimensions: the development of variant models, and configuring the models for process variants. For the first dimension, the team appreciated the support of DDM to identify business and syntactic drivers. In this way, they could better understand their business context. This benefit became even more prominent with the emergence of new drivers at low levels. Provop provides policies to define the base process. However, the team had to update the initial base process prepared according to such policies. Moreover, the team found the option design a challenging task which can also end up in different design alternatives. The approach provided hardly any guidance on how to design them.

For the configuration dimension, the team found it easier to use the Provop base process for the configuration of process variants. Similar change operations grouped in the option list decreased the complexity to generate a variant, and made it easier to configure a variant without much knowledge of the customer. The variation map of DDM does not provide any information on variants, one needs to have specialized knowledge for this task. The team suggested to add the driver names causing variations in related edges of the variation map to alleviate this problem. Overall, this requirement is strongly related with the user needs of *establishing a knowledge-base* and *reuse*, as decision support features of the approaches facilitated the conduct of the activities related to these needs. Both approaches were evaluated to outperform the current modeling approach.

6.3 Selection guidelines for process variant modeling approaches

Based on the evaluations, the team defined the guidelines in Table 9 for variant modeling approach selection. For each guideline, we refer to its source, the evaluated user needs and requirements that lead to its definition.

The highlight of DDM was its capability to embed all variant information in a single type of artifact, process model, while following standard BPMN notation and rules. The variation drivers based on business context provided a clear mechanism to identify and reuse variants. Based on DDM's notable properties, guidelines 1, 5 and 8 were identified. DDM may struggle if low-level variations are to be analyzed in detail and if variation starts at the top-level of the process hierarchy. Provop comes forward with its support to discover control-flow variations in fine detail, and accordingly provide a robust configuration and maintenance mechanism. Consequently, guidelines 2, 3, 4, and 6 were identified. These features come with the difficulties in consistently defining and maintaining artifacts other than the process models. Although such challenges may be even more prominent in a hierarchical process structure, Provop may be applied similar

Table 9 Process variant modeling approach selection guidelines together with related user need (N) and requirement (R) for each item (refined from [14])

No.	Source	Needs and constraints of the organization	Suggested approach	
			DDM	Provop
1	N1 N3 R3	You need an overall view of process variants with all information embedded in the models	X	
2	N1	You want lean models that are easy to read but can be detailed with additional artifacts		X
3	N1 R3	You need to analyze variations in control-flow perspective in detail		X
4	N2	You need to maintain your variants frequently due to specific and generic changes		X
5	N3 R7	The business context is the key for identifying and reusing variant information	X	
6	N3 R7	You need a practical mechanism to configure new process variants		X
7	N4 R5	You want to follow standard BPMN approach to model variants in a hierarchical structure		X
8	R3 R6	You don't want to use extra constructs in your modeling tool or define special rules for process modeling	X	

to standard BPMN way in those cases. This brings us to the identification of the guideline 7.

To identify these guidelines, in conformance with the participatory action research method, we worked on the problem of selecting a suitable process variant modeling approach in a specific case for two approaches. This work allowed us to obtain first-hand and deep knowledge on the problem. Eventually, the meticulous identification of user needs and requirements and in-depth analysis of the approach applications may show the way for other organizations to implement process variant modeling approaches in their own context. Although these user needs and requirements are not directly generalizable to other organizations, they point out the concerns to focus on for approach selection. Similarly, the guidelines identified here for the two approaches potentially represent key points in variant modeling, which organizations can use to assess their situation and evaluate other approaches as well.

7 Related work

The goal of this paper is to establish guidelines for practitioners to select relevant process variant modeling approaches for their use cases. For this purpose, we reported the design of our comparative process variant modeling study, and presented our findings from the application of the selected approaches in practice. To the best of our knowledge, the current literature supports practitioners by presenting a comparative analysis of the available approaches rather than defining a real-life use case and providing guidelines. In this respect, there are three studies that provide a systematic analysis of the approaches on process variability in the literature. La Rosa et al. [5] define a set of evaluation criteria for various dimensions of conceptual process variability approaches, and present a comparative overview of approaches identified with a systematic literature review (SLR). Ayora et al. [9] present a framework for the systematic evaluation of process variability support in process-aware information systems. The authors categorize the approaches identified with an SLR based on various aspects of variability such as the modeling language, the constructs used for representing variability, the techniques for configuring the variant models, and the tool support. These two studies provide an evaluation framework to understand the characteristics of process variability approaches and compare them with each other. Valenca et al. analyze the characteristics of business process variability and classify the approaches based on their characteristics, tool support, and validation status [10]. Although these studies are valuable to learn about the available process variant modeling approaches, they do not provide any particular guidance for practitioners to define their specific use case for process variant modeling for making a selection among the available approaches.

There are a few other studies that are complementary to our work to support practitioners in selecting relevant process variant modeling approaches. Torres et al. approach the field of process variant modeling from a cognitive perspective [34]. The authors compare the understandability of structural and behavioral process variant modeling approaches (restriction and extension as defined by La Rosa et al. [5]). The study may help practitioners understand cognitive complexities that can be introduced by the variant modeling approach they apply. Mechrez et al. bring out the deficiencies in representing variability for process perspectives and elements together with the neglected guidance for variability modeling [49]. Rosa et al. provide a summary of different variability modeling mechanisms that support reuse and present the details of approaches following such mechanisms [31]. Although a comparison is not provided, the study is beneficial to grasp the details of different variability mechanisms together. Lastly, Döhring et al. compare two selected approaches that implement different variability mechanisms with an experiment [7]. It emphasizes the importance of modularity support in variant management. The focus of this study is not to handle a real-life process variant modeling selection need in practice. Thus, the approach selection and implementation procedures are not defined for such a purpose in mind. However, the study displays a valuable example of support to practitioners to select a process variant modeling approach.

8 Conclusion

In this study, we report on a participatory action research study to investigate the use of process variant modeling approaches in practice and provide guidelines for practitioners to choose a suitable approach for their needs. We identified a process management consultancy company that is in need of process variant modeling as an industrial case. The involved company had various process definitions of the same process since they provided BPM services in similar areas to their customers. The company experienced problems in maintaining a high number of similar process models, which they had to frequently update. Moreover, they were not able to reuse their knowledge when they work on the processes of existing customers, or create processes for new customers.

We established a team of company employees and researchers to perform the comparative variant modeling study. The team defined requirements for process variant modeling approaches and applied a selection procedure by using related literature based on the identified requirements. Two approaches were selected that conformed to the requirements and displayed different variant modeling characteristics to make a comparative analysis. The chosen approaches were DDM and Provop. The team implemented these approaches for the selected software project management process. The approaches were further applied to a subprocess to observe how the approaches can be used to manage variability in hierarchical processes. It was observed that both of these approaches may bring value to an organization in a reasonable time interval and may support the company to meet their variant management needs. In general, the findings indicated that DDM provides a good support to understand the business context and manage the variants with a high-level perspective. On the other hand, Provop provides tools to deeply analyze the variant relations mostly based on the control-flow perspective.

Both approaches we analyzed have their merits and limitations. Thus, there is no single answer to the question of which approach is superior. However, our experiences may provide various benefits to practitioners. Even when professionals decide to use another variant modeling approach or no approach at all, learning about variant analysis through these approaches may prove useful. For example, when organizations explore business drivers causing variations, they can use this information to evaluate root-causes and deal with this variation on a strategical level. By means of the list of guidelines we present in this study, we aim to help organizations to select a proper approach suitable for their needs and provide insights on how to manage variability. In this work, we focused on approaches that capture variability in the control-flow perspective. This decision is both in line with the literature, since this perspective is most commonly examined, and with the needs of the selected company. However, in some cases, it may be essential to manage variability in other process perspectives. In future work, we aim to address this need by performing a similar study for approaches on other perspectives. Additionally, we are planning to further investigate the application of process variant modeling approaches in hierarchical process repositories.

Acknowledgements This study has been conducted in 4S Information Systems in collaboration with the process management team of the company. This study received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 660646. We would like to thank to the anonymous reviewers for their constructive suggestions.

References

1. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data Knowl. Eng. **58**(3), 358 (2006)
2. La Rosa, M., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R.M., Mendling, J., Dumas, M., García-Bañuelos, L.: APROMORE: an advanced process model repository. Expert Syst. Appl. **38**(6), 7029 (2011)

3. Raduescu, C., Tan, H.M., Jayaganesh, M., Bandara, W., zur Muehlen, M., Lippe, S.: A framework of issues in large process modeling projects. In: ECIS (2006), pp. 1594–1605 (2006)
4. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Comput. Ind. **62**(5), 467 (2011)
5. Rosa, M.L., Van Der Aalst, W.M.P., Dumas, M., Milani, F.P.: Business process variability modeling: a survey. ACM Comput. Surv. **50**, 2 (2016)
6. Armas-Cervantes, A., Baldan, P., Dumas, M., García-Bañuelos, L.: Diagnosing behavioral differences between business process models: an approach based on event structures. Inf. Syst. **56**, 304–325 (2016)
7. Döhring, M., Reijers, H.A., Smirnov, S.: Configuration vs. adaptation for business process variant maintenance: an empirical study. Inf. Syst. **39**, 108 (2014)
8. Dijkman, R.M., Rosa, M.L., Reijers, H.A.: Editorial: Managing large collections of business process models-current techniques and challenges. Comput. Ind. **63**(2), 91 (2012)
9. Ayora, C., Torres, V., Weber, B., Reichert, M., Pelechano, V.: VIVACE: a framework for the systematic evaluation of variability support in process-aware information systems. Inf. Softw. Technol. **57**, 248 (2015)
10. Valenca, G., Alves, C., Alves, V., Niu, N.: A systematic mapping study on business process variability. Int. J. Comput. Sci. Inf. Technol. **5**(1), 1 (2013)
11. H.P. Enterprise.: Project and Portfolio Management-PPM (2016). http://www8.hp.com/us/en/software-solutions/ppm-it-project-portfolio-management/
12. Milani, F., Dumas, M., Ahmed, N., Matulevičius, R.: Modelling families of business process variants: a decomposition driven method. Inf. Syst. **56**, 55 (2016)
13. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. J. Softw. Maint. Evol. Res. Pract. **22**(6–7), 519 (2010)
14. Aysolmaz, B., Yaldiz, A., Reijers, H.A.: A Process Variant Modeling Method Comparison: Experience Report, pp. 285–300. Springer, Berlin (2016)
15. Avison, D.E., Lau, F., Myers, M.D., Nielsen, P.A.: Action research. Commun. ACM **42**(1), 94 (1999)
16. Sjoberg, D.I.K., Dyba, T., Jorgensen, M.: The future of empirical methods in software engineering research. In: Future of Software Engineering, FOSE ’07, pp. 358–378. IEEE Computer Society, Washington, DC (2007)
17. Recker, J.: Scientific Research in Information Systems: A Beginner’s Guide. Springer, Berlin (2013)
18. Baskerville, R., Myers, M.D.: Special issue on action research in information systems: making is research relevant to practice–foreword. Mis Q. **28**(3), 329 (2004)
19. Davison, R., Martinsons, M.G., Kock, N.: Principles of canonical action research. Inf. Syst. J. **14**(1), 65 (2004)
20. Recker, J., Mendling, J.: The state of the art of business process management research as published in the BPM conference. Bus. Inf. Syst. Eng. **58**, 55–72 (2015)
21. Whyte, W.F.: Advancing scientific knowledge through participatory action research. Sociol. Forum **4**(3), 367 (1989)
22. Wohlin, C., Runeson, P., Bost, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell (2000)
23. Susman, G.I., Evered, R.D.: An assessment of the scientific merits of action research. Adm. Sci. Q. **23**(4), 582 (1978)
24. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131 (2009)
25. Yin, R.K.: Case Study Research: Design and Methods, 3rd edn. SAGE Publications Inc, Thousand Oaks (2003)
26. Argyris, C., Schon, D.A.: Participatory action research and action science compared. Am. Behav. Sci. **32**(5), 612 (1989)
27. Berger, I.R., Soffer, P., Sturm, A.: Organisational reference models: supporting an adequate design of local business processes. Int. J. Bus. Process Integr. Manag. **4**(2), 134 (2009)
28. Hallerbach, A., Bauer, T., Reichert, M.: Managing process variants in the process life cycle. In: ICEIS 2008-Proceedings of 10th International Conference on Enterprise Information Systems, vol. 2 ISAS, pp. 154–161. Centre for Telematics and Information Technology, University of Twente, Enschede (2007)
29. Kunze, M., Weidlich, M., Weske, M.: Querying process models by behavior inclusion. Softw. Syst. Model. **14**(3), 1105 (2015)
30. Lu, R., Sadiq, S., Governatori, G.: On managing business processes variants. Data Knowl. Eng. **68**(7), 642 (2009)
31. Rosa, M.L., Dumas, M., ter Hofstede, A.H.M.: Modelling business process variability for design-time configuration. Handb. Res. Bus. Process Model. **2008**, 204 (2009)
32. Derguech, W., Vulcu, G., Bhiri, S.: An Indexing Structure for Maintaining Configurable Process Models, pp. 157–168. Springer, Berlin (2010)
33. Project Management Institute Inc.: A Guide to the Project Management Body of Knowledge (PMBOK® guide) (2000). ISBN: 978-1-935589-67-9
34. Torres, V., Zugal, S., Weber, B., Reichert, M., Ayora, C., Pelechano, V.: A qualitative comparison of approaches supporting business process variability. In: Business Process Management Workshops-BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Paper, pp. 560–572 (2013)
35. Recker, J., Dreiling, A.: The effects of content presentation format and user characteristics on novice developers’ understanding of process models. Commun. Assoc. Inf. Syst. **28**(1), 65 (2011)
36. Milani, F.: On Sub-Processes, Process Variation and Their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Ph.D., University of Tartu (2015)
37. Schunselaar, D.M.M.: Configurable Process Trees: Elicitation, Analysis, and Enactment. Ph.D., Technische Universiteit Eindhoven (2016)
38. Puhlmann, F., Schnieders, A., Weiland, J., Weske, M.: Variability mechanisms for process models. In: Technical Report, PESOA-Report No. Tr 17/2005. Process Family Engineering in Service-Oriented Applications (2005)
39. Schnieders, A.: Variability mechanism centric process family architectures. In: Proceedings of International Symposium and Workshop on Engineering of Computer Based Systems, pp. 287–298 (2006)
40. Döhring, M., Zimmermann, B.: vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules, pp. 332–341. Springer, Berlin (2011)
41. Hallerbach, A., Bauer, T., Reichert, M.: International handbooks on information systems configuration and management of process variants. In: Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1 SE-11, pp. 237–255. Springer, Berlin (2010)
42. Milani, F., Dumas, M., Matulevičius, R.: Decomposition driven consolidation of process models. In: Salinesi, C., Norrie, M., Pastor, Ó. (eds.) Advanced Information Systems Engineering SE-13, Lecture Notes in Computer Science, vol. 7908, pp. 193–207. Springer, Berlin (2013)
43. Milani, F., Dumas, M., Matulevičius, R.: Identifying and Classifying Variations in Business Processes, pp. 136–150. Springer, Berlin (2012)
44. van der Aalst, W.M.P., van Hee, K.M., ter Hofstede, A.H.M., Sidorova, N., Verbeek, H.M.W., Voorhoeve, M., Wynn, M.T.: Soundness of workflow nets: classification, decidability, and analysis. Formal Aspects Comput. **23**(3), 333 (2011)

45. Hallerbach, A., Bauer, T., Reichert, M.: Guaranteeing soundness of configurable process variants in Provop. In: CEC, pp. 98–105. IEEE (2009)
46. Dumas, M., García-Bañuelos, L., Polyvyanyy, A.: Unraveling Unstructured Process Models, pp. 1–7. Springer, Berlin (2010)
47. Hallerbach, A., Bauer, T., Reichert, M.: Issues in Modeling Process Variants with Provop, pp. 56–67. Springer, Berlin (2009)
48. Reijers, H.A., Mendling, J.: A study into the factors that influence the understandability of business process models. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(3), 449 (2011)
49. Mechrez, I., Reinhartz-Berger, I.: Modeling Design-Time Variability in Business Processes: Existing Support and Deficiencies, pp. 378–392. Springer, Berlin (2014)

Banu Aysolmaz is an assistant professor at the Accounting and Information Management Department, Maastricht University. She worked as a post-doctoral researcher and a Marie Curie fellow with the Department of Computer Science at the Vrije Universiteit Amsterdam. Her research interests include business process modeling, software engineering, process model comprehension, and visualization. She obtained her Ph.D. in information systems from Middle East Technical University (METU), Ankara, Turkey. Her doctoral thesis received 2014 METU year of the thesis award. She worked as a consultant in the areas of business process management and software process improvement in many organizations in Turkey.

Dennis Schunselaar is a post-doctoral researcher in the Business Informatics group at the Vrije Universiteit Amsterdam. Currently, he is working as a project leader within the AMUSE project. This project is a collaborating with the Utrecht University and AFAS, a Dutch ERP software vendor, on model-driven software engineering based on enterprise models. His research interests include: business process modelling, configurable process models, hybrid process models, enterprise modelling, and business process analysis. He obtained his Ph.D. from the Eindhoven University of Technology.

Hajo A. Reijers is a full professor at Vrije Universiteit Amsterdam, where he heads the Information Management and Software Engineering group of the Computer Science department. He is also a part-time, full professor at the Department of Mathematics and Computer Science of Eindhoven University of Technology. His expertise is in enterprise systems, business process management, process mining, conceptual modeling, and workflow technology. Reijers has published over 150 scientific papers, chapters in edited books, and articles in professional journals.

Ali Yaldiz received his B.S. degree in computer engineering from Bilkent University, Ankara, Turkey in 2011 and the M.S. degree in software management from Middle East Technical University, Ankara, Turkey in 2015. His research interests are business process management, process improvement, and project and portfolio management. He has been working in industry in these areas since 2011.