Middle East Technical University
Informatics Institute

EVALUATION OF PROCESS VARIANT MODELING APPROACHES: A CASE STUDY

Advisor: Prof. Dr. Onur DEMİRÖRS
(METU)

Co-Advisor: Dr. Banu AYSOLMAZ
(METU)

Student Name: Ali YALDIZ
(SM)

January 2016

TECHNICAL REPORT
METU/II-TR-2016-

Orta Doğu Teknik Üniversitesi
Enformatik Enstitüsü

SÜREÇ DEĞİŞKENLİK MODELLEMESİNİN DEĞERLENDİRİLMESİ: BİR VAKA ÇALIŞMASI

Danışman: Prof. Dr. Onur DEMİRÖRS
(ODTÜ)

Yardımcı Danışman: Dr. Banu AYSOLMAZ
(METU)

Öğrenci Adı: Ali YALDIZ
(SM)

Ocak 2016

TEKNİK RAPOR
ODTÜ/II-TR-2016-

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Internal Use)	**2. REPORT DATE** 12.01.2016

3. TITLE AND SUBTITLE

EVALUATION OF PROCESS VARIANT MODELING APPROACHES: A CASE STUDY

4. AUTHOR (S) Ali YALDIZ	**5. REPORT NUMBER (Internal Use)** METU/II-TR-2016-

6. SPONSORING/ MONITORING AGENCY NAME(S) AND SIGNATURE(S)
Software Management Master's Programme, Department of Information Systems, Informatics Institute, METU
Advisor: Prof. Dr. Onur DEMİRÖRS Signature:

7. SUPPLEMENTARY NOTES

8. ABSTRACT (MAXIMUM 200 WORDS)

In this study the usage of two different process variant modeling approaches are investigated in an organization where processes for the same process area are modeled for different customers based on the same best practices. The study is conducted as a case study on project management process models of 5 different companies. The efficiency of two different process variant modeling approaches with respect to each other and traditional methods is investigated. The first approach, the decomposition driven approach includes the decomposition of a main process into sub-processes and consequently results in a variation map in order to increase comprehensibility and degrade complexity. On the other hand, the Provop approach produces a base model and its options by analyzing the process variants of the companies. In this approach, not only controllability but also flexibility of the design improve. The case study concludes that the decomposition driven approach is suitable for the process variants, modelling of which requires an essential consideration of compact and clear design whereas the Provop approach is appropriate for the design of the process variants in which the controllability and manageability is a necessity.

9. SUBJECT TERMS Process Variants, Decomposition Driven Method, Provop, Business Process Management	**10. NUMBER OF PAGES** 31

Table of Contents

List of Figures

List of Table

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In enterprises, business process modelling (shortly process modelling) is of great importance since it provides an improvement on the efficiency and the quality of the activities that an organization has to make in order to achieve a particular business goal. In process modelling, one of the problems that the designers encounter is to deal with business process variability. Based on the changing business requirements in various countries, diversities in the organizational structure, or the differences in the infrastructure of information technologies, it is inevitable that the process modelling requires the consideration of process variants (Döhring, Reijers, & Smirnov, 2014). However, in the design of the process model, it is a challenging task to integrate the process variants into the core model while preventing the complication and redundancy (Fahland et al., 2009), (Rolon et al., 2009). Also, another difficulty is to design the base model as a reference for the related process models and, at the same time, to manage the relations between the process variants (Döhring et al., 2014), (Briand, Bunse, & Daly, 2001).

1.2 PURPOSE OF THE STUDY

The aim of the study is to investigate the methods proposed for process variant modelling in the literature and to evaluate the proposed methods in terms of efficiency, applicability, and complexity in a real life setting. For that aim, the following process variant management methods are studied:

- Decomposition driven method in (Milani, Dumas, Ahmed, & Matulevicius, 2013)
- Provop Approach in (A. Hallerbach, Bauer, & Reichert, 2010a)Also, another goal for this study is to provide a result based on the methods in (Milani et al., 2013) and (A. Hallerbach et al., 2010a) by analysing the real business process models.

1.3 SIGNIFICANCE

When an organization needs to maintain multiple process definitions due to the factors such as location, changing customers, time; managing process definitions and BPM systems becomes more complex and consequently it becomes difficult to organize the process variants in an appropriate manner. To solve that problem, process variants modelling approaches are proposed in the literature. However, the companies can have difficulties while determining the suitable methods for their need and constraints. To our best knowledge, there is no study in the literature to overcome these issues. For that aim, this study evaluates two process variants modelling approaches in terms of complexity, comprehensibility, controllability and flexibility, and it provides a perspective for the companies that employ business process management approaches efficiently and accurately. Another significance of the study is that based on the process variants obtained from different companies, a common model can be generated and this model provides a vision for the other companies to realize their needs and figure out the potential alternative ways to reach the outcome effectively.

The main contributions of this study can be summarized as follows:

- For the first time, two process variants modelling approaches are analyzed and consequently an outline is provided in order to guide the companies that employ process variant modelling approaches for their needs.
- A common model is proposed based on the variants obtained from different companies. This common model provides a support for the future applications of the companies and with the consideration of this common model, the newcomers

can improve their processes efficiently by exchanging the cumulative experience gained from other professionals.

1.4 RESEARCH METHODOLOGY AND RESEARCH QUESTIONS

In this study, the Decomposition Driven method and Provop approach in (Milani et al., 2013) and (A. Hallerbach et al., 2010a) are compared based on a case study in which the methods are applied to the project management business processes of 5 different companies. The processes developed for the companies will be compared with the best practices obtained from a process management automation system based on Project Management Institute (PMI) standards. To evaluate the efficiency and applicability of the resultant process variant models with respect to conventional models, interviews with the employees of the companies will be performed.

Based on the methodology described above, the following questions can be answered for the model proposed in this study:

- How can we develop process variant models for a process area where different process models are developed for diverse companies based on the same best practices?
- How does the application of two variant modeling methods, the Decomposition Driven (Milani et al., 2013) and the Provop (A. Hallerbach et al., 2010a) methods, compare for flexibility in terms of reusing the knowledge to define processes for new customers and maintain all variants in case of a change in one process?
- What factors are to be considered for an organization to select a proper variant modeling method based on its setting, needs and constraints?

CHAPTER 2

LITERATURE REVIEW

The efficiency and quality of process models is of great importance for the companies and their corresponding customers (Döhring et al., 2014),(Lind & Goldkuhl, 2005),(Weidlich & Weske, 2010). Process management (BPM) tools are proposed in order to achieve the efficiency and quality goals. Also, the technologies, based on workflow analysis and case handling, help to improve the business process model. Moreover, the change requests on business requirements needs to be handled. Some business requirements supplement existing requirements by leading to different behaviour of processes as executed before. As a result, the need for an effective process variant modelling appears.

Business process models indicate the activities which an organization has to perform in order to reach a particular business goal. In the model, the process type is implemented based on the constraints such as control flow and resources. One of the important problems in the design of the business process model is to deal with business process variability. For each process type, there exist multiple process variants depending on the process context. However, BPM tools do not provide an efficient support for modelling and maintaining these process variants. Therefore, it requires architecture to solve the design issues based on process variability.

Conventional Business Variant Modelling Approaches

The conventional methods for the business variant modelling can be considered in two different groups:

The multi model approach models the process variants in separate process models (Dumas, 2012), (A. Hallerbach, Bauer, & Reichert, 2010b), (Awad, Sakr, Kunze, & Weske, 2011). In practice, it creates redundant model data since most of the similar parts are repeated for different variants. This leads to significant modelling and maintenance efforts. In other words, the effort need for modelling and maintaining a process is high since fundamental changes are handled in a new and separate process model for each individual process model. Also, the process variants are not strongly connected with each other since they occur in different model scheme. Another problem of this approach is that it is not possible to combine the existing variants to new ones (Wahler & Küster, 2008). Besides, since each variant requires a different modelling activity, it takes too much time to maintain the variants. Therefore, this approach is not very useful for the companies which require time-variant process modelling.

The other modelling approach is single-model approach which captures multiple variants in one single process model (Dumas, 2012), (A. Hallerbach et al., 2010b), (Reijers, Mans, & van der Toorn, 2009), (Rosemann & van der Aalst, 2007), (Gottschalk, Van Der Aalst, Jansen-Vullers, & La Rosa, 2008). This approach uses conditional branches and labels to indicate the flow in the model. Each variant can be represented by a particular execution path. Even though this type of approach creates a large model, it makes combining and merging operations possible for the variants which have similar parts. However, it has some drawbacks such as complicated representation and unawareness of the single variant based process model. In single model approach, it is difficult define variants neither explicitly nor transparently.

As a conclusion, both single-model and multi-model approaches could cause deficiencies while handling business process variants and providing feasible solution in many cases. However, since Multi-Model Approach causes high redundancy and high effort need to maintain, the Single-Model Approach is accepted as more feasible for the industry.

CHAPTER 3

CASE STUDY

As discussed earlier, most of the studies in the literature considers two different approaches. One of them is single-model approach, which makes possible reusing existing models by embedding new variants to the common process model via conditional branching. On the other hand, the other method is multi-model approach in which process variants are generated by copying a process model and setting its parameters with respect to corresponding variants.

In this study, we have selected two of the well accepted Single-Model Approach methods which are Decomposition Driven Method and Provop approach. In this part, we express the case study which created by applying these two approaches. In order to apply methods, we have the Software Project Management processes of 5 different company. The four of these companies are from Turkey and the remaining one is Turkey office of an international company while their sectors are banking, telecommunication and insurance.

The Organization and the Need for Variant Modelling

4S is a consultancy company that provides process analysis, improvement and automation services to its customers using HP PPM product. 4S has customers from various countries and industries focusing on different process areas. Usually, 4S analysts need to rely on their own expertise to discover other activities and improve the existing process. They cannot systematically exploit process knowledge obtained from previous similar companies for new customers. Based on the problem, the need for using a process variant modelling method for 4S can be summarized as follows:

- When they start to work with a new customer, 4S analysts need to combine their knowledge on previous customers as a baseline for understanding the new as-is process and suggesting improvements.

- Through the steps of process analysis, improvement and design, 4S analysts design various processes for customers. Even when they start developing a process based on a previously encountered process, the knowledge of such related pro-cesses and the connections hereto are soon lost. Analysts cannot benefit from one another's experiences as it is hard for them to go over each process to find out if it is relevant for a new case. The same problem persists through process enactment phase; as developers cannot easily find out similar automated processes and activities.

- When an improvement or update is needed, 4S needs to go over each customer's processes to find out which ones are affected and where updates are needed. This requires a lot of effort and can introduce errors due to manual review process.

During the case study, below company processes are considered (see Figure 1).

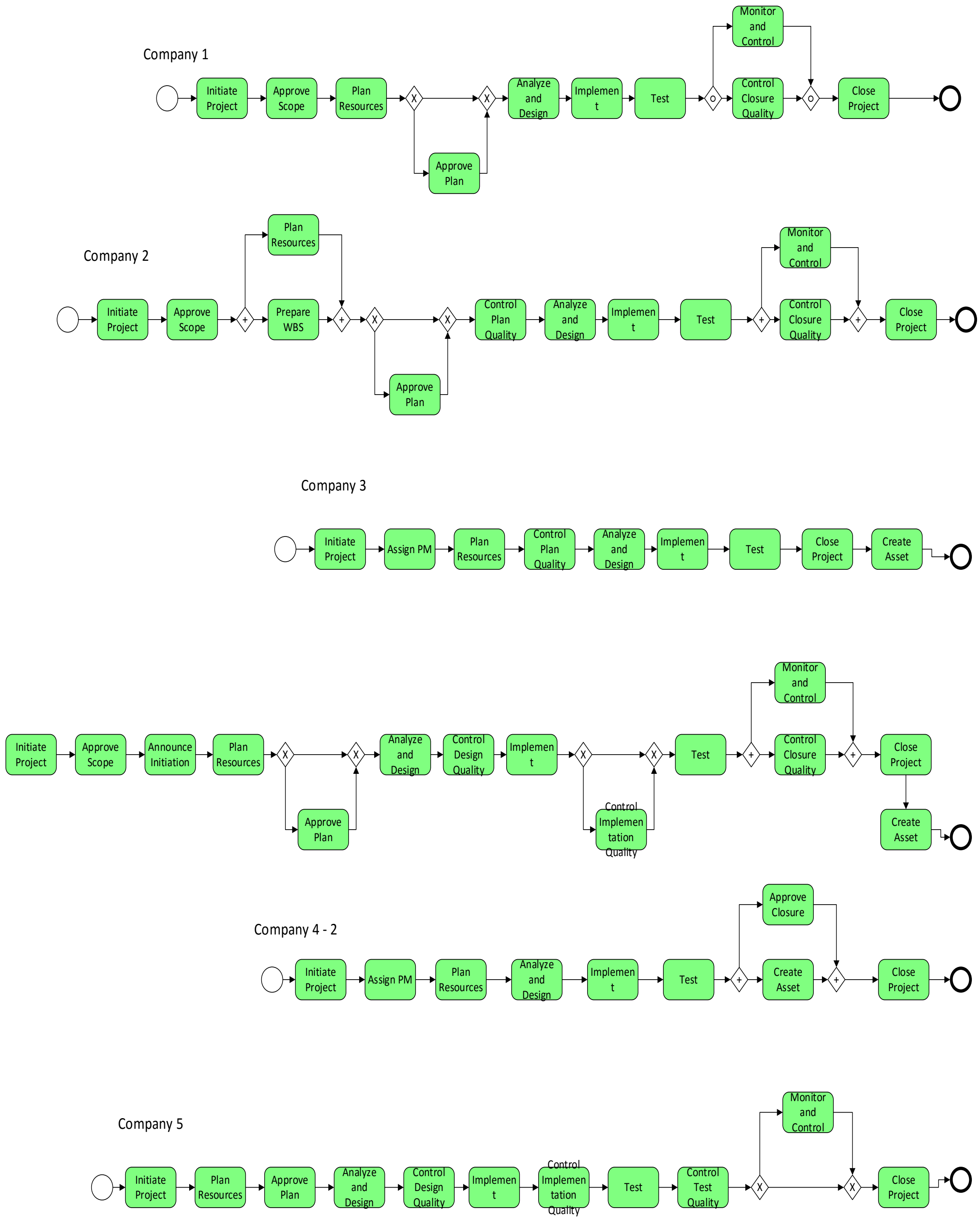

Figure 1. Six variants of the software project management process

3.1 DECOMPOSITION DRIVEN METHOD

The method in (Milani et al., 2013) is based on decomposition of process models. The method starts with the definition of a main top-level process. Then, each activity in the main process is defined in detail in a sub-process. Later, the sub-processes is further decomposed into sub-processes until there is no further possible decomposition.

While decomposing the processes, it is important to decide how to model multiple variants of a sub-process; either in an integrated way as a variation map, or as separate process models for each variant. The method proposes two different drivers to deal with this problem. The first type of drivers is business drivers which are the reasons to treat the processes as separate or together. Thus, two or more variants can be modelled separately or together based on the business drivers. Business drivers are determined based on the resources, type of organisation, products, services, and customers, etc. Also, timing is an important concern for identifying business drivers. For example, a tourism agency can define various business drivers according to tourist seasons. In summer seasons, airline companies optimize the number of flights by increasing flights to the seaside places. On the other hand, in winter seasons due to weather conditions they optimize the balance between the safe flight and profit. Therefore, the season is a business driver for airline companies. The second type of drivers is syntactic drivers that are about the differences in the way two or more variants produce their outcomes. In this type of drivers, modelling can be performed for these variants separately. If they are modelled in a consolidated way, complexity increases; but, comprehensibility decreases. However, in practice it is difficult to model them separately since syntactic drivers require the availability of the model of separate variants and the usage of same notation for same level of granularity and modelling conventions. Therefore, first the similarity between variants of a process is specified based on the opinions of domain experts, then the expected differences between the separate models of these variants are analysed.

In the proposed method, the decisions on whether to model two or more process variants together or separately can be specified after each decomposition. The proposed method also uses consolidated modelling approach for each sub-process as a default option until the fragmented approach become applicable based on the business and syntactic drivers.

In our case study, 7 different Software Project Management processes of 5 different companies from Turkey was analysed. The processes were defined as workflow definitions on HP PPM, but process models were not developed for analysis purposes. We converted the low level workflow models to process models in BPMN notation through discussion sessions with experts. We aggregated workflow tasks to higher level activities in BPMN. A sample conversion, can be seen on Figure 2.

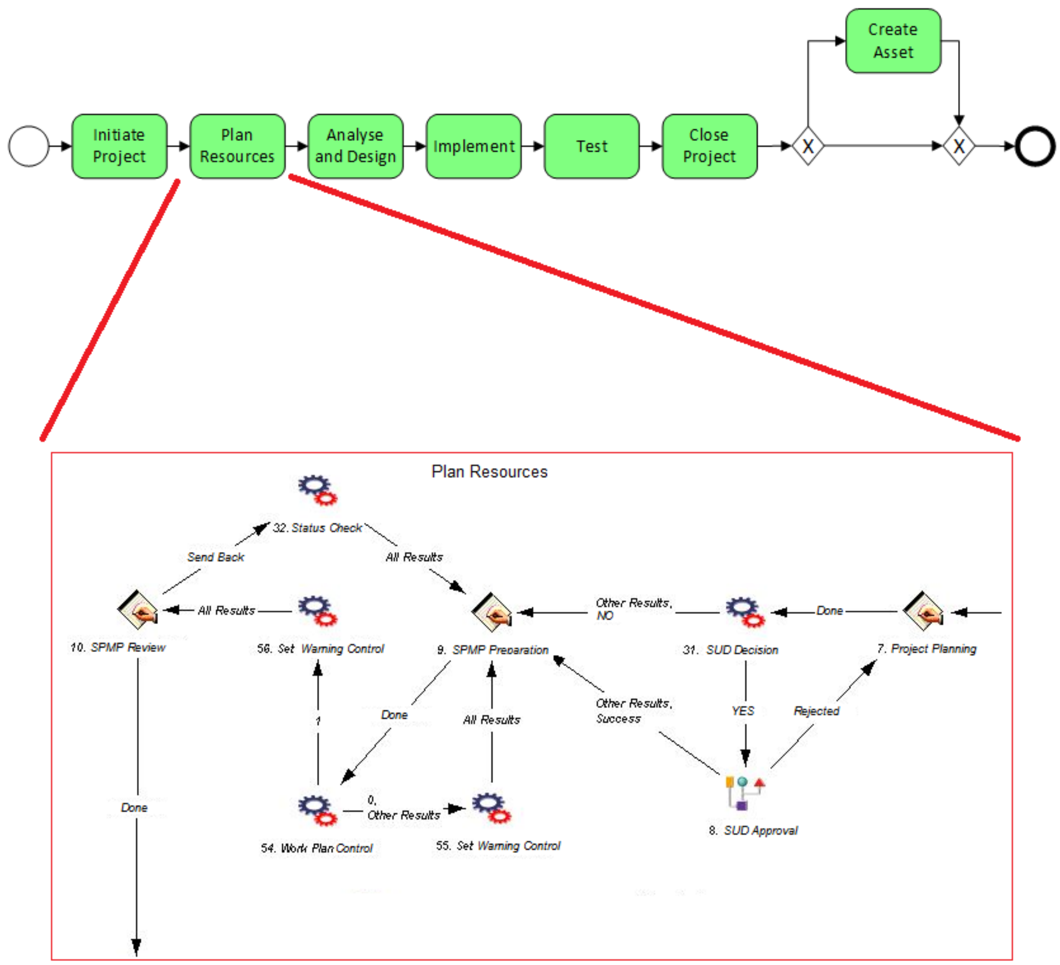

Figure 2. Conversion of Plan Resources activity

Then considering the decomposition driven method, the following steps are performed to evaluate the trade-off between modelling multiple variants of a business process together and modelling them separately.

Step 1 - Model the main process of Software Project Management

As first step of the case study, we modelled the main process of Software Project Management by using BPMN notation. During modelling of the main process, we used existing models of companies and the proposed best practice model. The companies' existing processes are evaluated according to best practice model and by grouping some existing activities, the tasks (sub-processes) of the main process are reached. As a result, we have added only "Plan Resources" step to the existing best practise model, and it is observed that the remaining best practise steps are in use for companies. While modelling the main process, we also investigated and summarized each company's existing processes in order to point out how they add value to the process. The metrics of existing Software Project Management processes are listed on

Table 1. These process are running on a process automation system and the process metrics are collected from this system. As a result of the step 1, the main model of the Software Project Management process is figured out (see Figure 3).

The main process is initiated when a project decision is received. The first task is "Initiate Project" meaning entering the project data and definition to process automation system. The next task, "Plan Resources", exists for planning activities of the project. After planning completed, "Analyse and Design" takes place. When design outputs received by development team, this time the "Implementation" takes place. The next task on the process is "Test". After all tests are completed, some sub tasks like deployment, monitor and control and quality control can be performed, but they are not indicated on this main process. The next task is "Close Project" which means all planned task and controls are performed and project will be archived and become inactive on process automation system. The last task of the model is "Create Asset", however as it is demonstrated on the model, this is an optional task and it does not be performed for all instances.

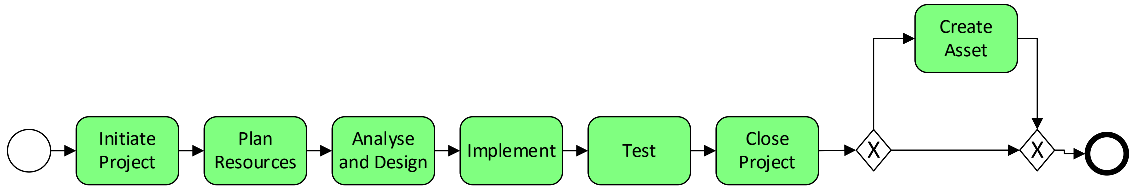

Figure 3. Model for main process of Software Project Management

Table 1. Metrics of existing Software Project Management processes

Process	Field	Number of workflow tasks	Number of workflow gateways	Number of BPMN activities	Number of BPMN gateways
Company1	Annuity Insurance	15	7	10	4
Company2	Insurance	40	9	12	6
Company3	Banking	21	14	9	0
Company4-1	Banking	48	7	14	6
Company4-2	Banking	8	2	9	2
Company5-1	Telecom	46	11	11	2
Company5-2	Telecom	44	8	11	2
Average		31.7	8.3	10.9	3
PMBOK Best Practice		13	0		

Step 2 - Identify variation drivers

In the second step, identified variation drivers of the process. As the process that we study on is somehow well known and widely applied by industry contributors, we focused on approaches of companies for the classification of business drivers. In order to determine the business drivers of the main process, some examples from (Milani et al., 2013) are investigated. As we have pretty linear main process, we focused on how the task on main process are performing and what can cause a variation. It is observed that the application of tasks from main process can vary according to companies. For example, the "Initiate Project" task is handled in different ways for each company, but almost all of them providing same outputs. Obtaining nearly same outputs from same tasks for any company leads us, there is no explicit syntactic drivers. In order to handle this approach more effectively in the scope of this study, we decided to go on with business drivers and selected company driven variations.

Step 3 – Asses the relative strength of variation drivers

In order to assess relative strength of variation drivers, the outputs of previous step, which are companies, were used. As it can be predicted, all companies have same strengths to be handled as variation drivers. Therefore, in our case, all companies are variation drivers, but company 4 and Company 5 has two Software Project Management processes each. Both process models of Company 5 are very similar but, the process models of Company 4 are mostly different. Therefore, we used an additional variation driver as "Company 4 – 2" with the same strength.

Step 4 - Identify the variants of each sub-process of the main process

In this step, we populated a variation matrix in Table 3 by using the outputs of previous steps. First of all, in order to define first column of the variation matrix, we used the variation drivers and their relative strength, whereas the first row of the variation matrix consists of the tasks of the main process. Then, the process automation system was investigated for the details of each sub-process under the main process. As a result of the investigation, each tasks on the main process was decomposed into sub-processes as variation definitions (see Table 2).

Table 2. Variation Definitions

Level / Activity	Initiation	Planning	Analyze and Design	Implementation	Test	Closure	Asset Creation
-							-Create Asset
Basic	-Project Definition	-Planning	-Analyze and Design	-Implementation	-UAT	-Deployment -Closure	
Fast		-Planning -Scheduling				-Monitoring and Control -Closure	
Simple	-PM Assignment -Project Definition	-Planning -Planning Approval				-Deployment -Closure Approval -Closure	
Moderate	-Initiation Approval -Project Definition -Scope Approval	-Planning -Planning Quality Control -Planning Approval				-Deployment -Monitoring and Control -Closure	
Detailed	-Project Definition -Scope Approval -Initiation Announcement	-WBS Creation -Planning Approval	-Analyze and Design -Design Quality Control	-Implementation -Implementation Quality Control	-UAT -UAT Quality Control	-Monitor and Control -Closure -Closure Quality Control	
Complex	-PM Assignment -Project Definition -Initiation Quality Control -Scope Approval	-WBS Creation -Planning Quality Control -Planning Approval				-Closure Report Preparation -Closure Approval -Closure	

For example, in Company 1, "Initiate Project" task is handled with a sub-process which consists of "PM Assignment", "Project Definition", "Quality Control", and "Scope Approval" activities, while the same task is handled with a sub-process that contains "PM Assignment" and "Project Definition" activities for Company 3. While applying this decomposition, the granularity level is determined by considering the aim of the step that is identification of variations. In other words, the granularity level is deepened until the variations of activities are explicitly defined. As a result of the decomposition, the variations are identified on Variation Matrix (see Table 3).

Table 3. Variation Matrix

Software Project Management	Initiate Project	Plan Resources	Analyze and Design	Implement	Test	Close Project	Create Asset
Company 1	-Complex Initiation	-Moderate Planning	-Basic Analyze and Design	-Basic Implementation	-Basic Test	-Detailed Closure	
Company 2	-Moderate Initiation	-Complex Planning	-Detailed Analyze and Design	-Detailed Implementation	-Detailed Test	-Complex Closure	-Asset Creation
Company 3	-Simple Initiation	-Basic Planning	-Basic Analyze and Design	-Basic Implementation	-Basic Test	-Basic Closure	-Asset Creation
Company 4	-Detailed Initiation	-Simple Planning	-Detailed Analyze and Design	-Detailed Implementation	-Detailed Test	-Fast Closure	
Company 4 - 2	-Simple Initiation	-Fast Planning	-Basic Analyze and Design	-Basic Implementation	-Basic Test	-Simple Closure	-Asset Creation
Company 5	-Basic Initiation	-Detailed Planning	-Detailed Analyze and Design	-Detailed Implementation	-Detailed UAT	-Moderate Closure	

Step 5 – Perform similarity assessment of variants for each sub-process of the main process

In this step, we performed similarity assessment by analysing each action on variation matrix in Table 3. The identical actions are tried to be pointed out with the help of process automation system data. The collected and used data while performing actions, number of workflow steps and the role of actioner were main points to be investigated in order to decide actions are identical or not. Some inexplicit details of the actions are asked to the domain experts and obtained responses collected on a voting mechanism in which for each action, the remaining actions are graded with respect to the level of similarity to that action. For example, when the action "PM Assignment" investigated, it is observed that all project manager assignment actions are performed in same way on process automation system and it is easily labelled as a very similar action for all companies. The results obtained based on the grading method indicate that the very similar actions have identical characteristics for business drivers (i.e., companies). On the other hand, there are actions that do not have common specifications for business drivers. The action "Project Definition" is performed in different ways, as the collected data for "Project Definition" is mostly different for companies.

also includes the dissimilar actions that have different properties for each company. Furthermore, the task “Asset Creation” is an optional action on the main process, but when the task is investigated, it seems that it is similar for different companies and consequently.

Step 6 – Construct the Variation Map

As input from Step 4 and Step 5, we have the variants (actions causing variation) for each sub-process of the main process. The aimed action of this step was mapping these variants in a variation map (see Figure 4). For the case that we work on, we decided to model “Initiate Project”, “Plan Resources” and “Close Project” tasks of main process separately. According to variation matrix and similarity assessment, we conclude that these tasks are the reasons of variation and their decomposed application methods are not very similar. According to variation matrix of the process we proposed four separate variants of “Initiate Project”, five separate variants of “Plan Resources” and lastly five separate variants of “Close Project” activity. All of these sub-processes had a strong business driver and were not similar. Conversely, there were two process variants for “Analyse and Design”, “Implement” and “Test and one process variant for “Asset Creation”. These variants did not have strong variations and they were mostly similar. Thus, we modelled them together. The final version of variation map for each variants are shown in Figure 4. When a variant is modelled separately, the complexity decreases but the comprehensibility can also decreases. Therefore, with help of the Decomposition Driven Method, we aimed to balance this trade-off and we only separately modelled 3 tasks of the main process. The other tasks remained the same and modelled together, in order to increase the comprehensibility.

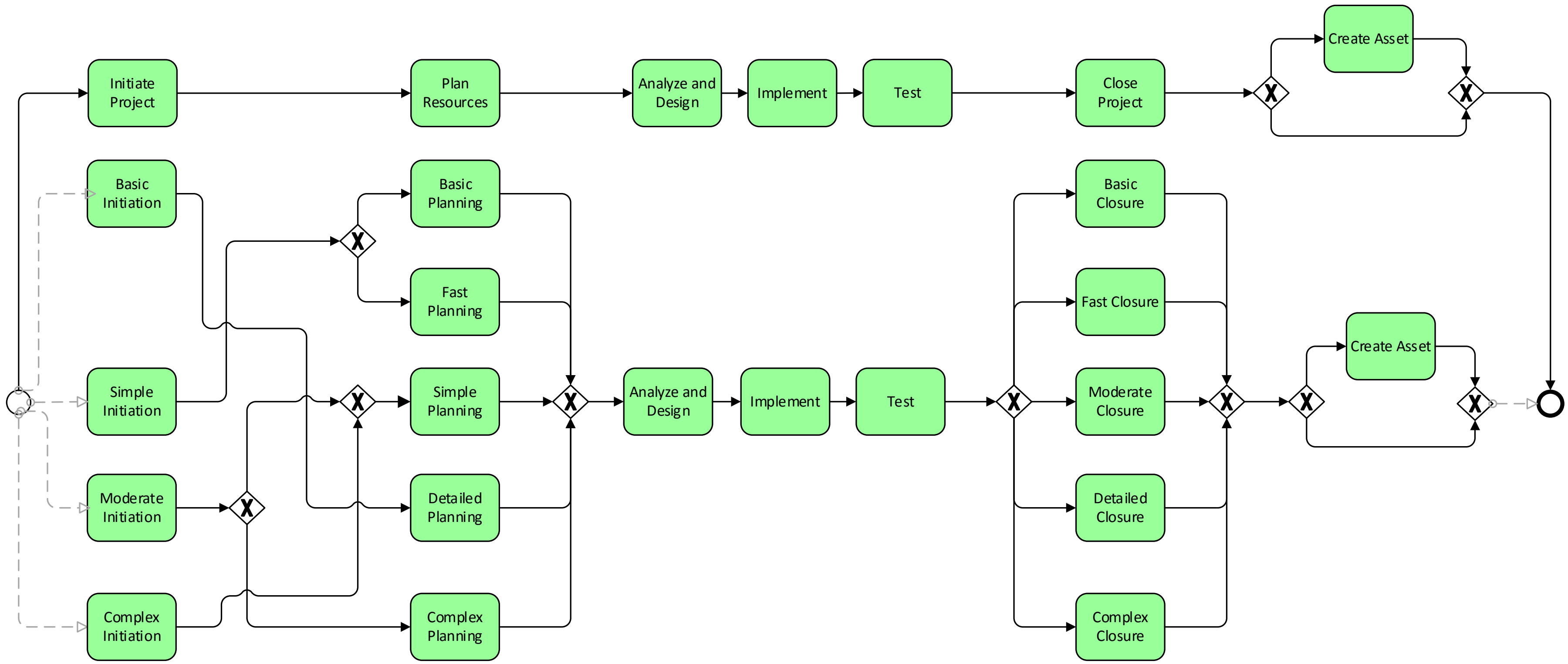

Figure 4. Variation Map

Step 7 – Configure a Specific Process Variant

The generated variation map acts as a reference model to observe both the process map and help 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 is done for Company 4 as shown with darker colored activities in Figure 5. We manually verified that we can generate all our variants as syntactically correct and sound.

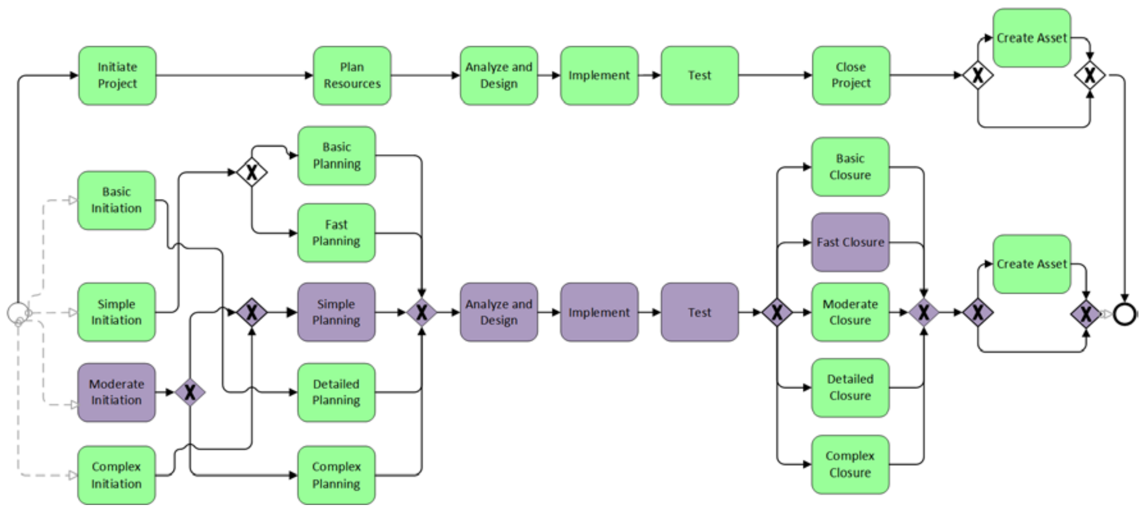

Figure 5. A specific variant on variation map

After this step, the Decomposition Driven method suggests the iteration of all steps for the sub-processes of the main process. We applied the Decomposition Driven method completely in the first level of decomposition in 4S. Moreover, we identified the activities to be placed in each sub-process and discussed a sketch of the variation maps with the experts. In this way, the experts were able to observe how the Decomposition Driven method provided a flexible way of variant modeling in different granularity levels. For example. For “Implement” process, variation in the high level is not found necessary. However, it is observed that variants of this sub-process need to be handled considering other business drivers such as project type.

3.2 PROVOP APPROACH

In the Provop method (A. Hallerbach et al., 2010a), (A. Hallerbach et al., 2010b), (a Hallerbach, Bauer, & Reichert, 2008), an approach similar to the single-model approach is utilized in order to configure variants based on a common process model. The common process model refers to a base process which can be configured in different ways to create specific process variants. The Provop approach provides operations in order to make the changes in the base process model. These operations are INSERT fragment, DELETE fragment, MOVE fragment and MODIFY attribute. In the base process, the *adjustment points* associated with a node enables to change the necessary parts with the help of change patterns. As the number of change operations is increased to create a variant model based on base model, the Provop approach groups some of the change operations to consider further reuse, which is called as *options*. This is important and useful since, the variants require similar change operations to be configured. By grouping of these similar change operations, the complexity to generate a variant decreases and the controllability of the model increases.

The main advantage of the Provop approach is that the variants are treated as first class objects. This implies that with the change operations and configuration of the process variants the Provop approach provides a solution for all phases of the process lifecycle. For example, when the requirements are changed in a period of the time, the method offers the options as a solution to make the necessary changes in the process model.

The method can be performed by applying the following two steps:

Step 1 – Designing a base process

In Provop approach, different policies are provided in order to design a base process.

- Policy 1: In this policy, the base process corresponds to a standard reference process. In our case study, base process includes best practice process provided by the process automation system with the consideration of Project Management Institute (PMI).

- Policy 2: This policy states that, if a particular process variant is used more frequently than the others, corresponding process variant can be chosen as a base process. This reduces the effort while as the number of necessary changes is less.
- Policy 3: With this policy, the optimized base process model is obtained with the minimization of average distance to other variant models. In other words, according to this policy, it is optimal to choose the variant model that requires the least number of operations to reach the other variant models.
- Policy 4: In order to apply this policy, all possible variants are merged in one process model using conditional branches. Then, only DELETE operations are employed to generate the process variants.
- Policy 5: This policy considers intersection of all variants as a base process. In this policy no DELETE operations are applied, however MOVE, MODIFY and INSERT operations are employed to generate process variants.

In our case study, the mixture of all policies are considered for the design of the base process since none of the policies described above is superior. In the first phase of the design, Policy 1 is employed by considering the best practice from process automation system. Also, Policy 2 is considered with Policy 1 to reduce configuration efforts. In the second phase of design, the approaches in Policy 3 and Policy 5 are examined and the base model in the design is evolved to obtain the sub-optimal minimal average distance to the other variants while analysing the superset of all process variants discussed in Policy 4.

The following model presents the best practice model obtained from Policy 1:

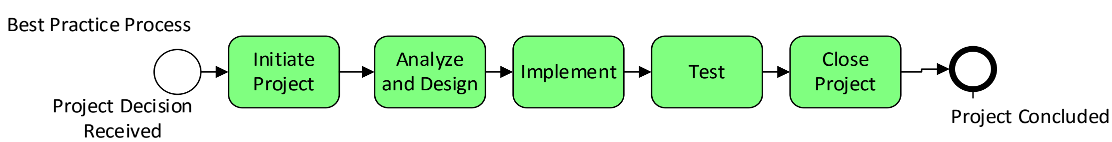

Figure 6. The best practice process model

The best practice process model in Figure 6 consists of 5 main activities. Based on the best practice process model and other policies, the following evolved version of the base process model is created:

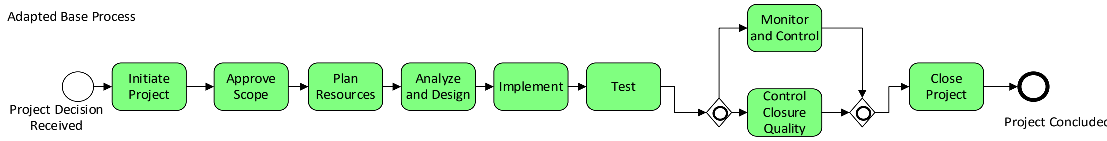

Figure 7. The adapted base process model

The adapted base process model in Figure 7 is obtained based on the variant models observed from Software Project Management processes of companies.

Step 2 – Defining Adjustment Points

While applying the Provop approach, the next action is determining the explicit positions of the adjustment points which are required by change operations in order to refer them. The Provop approach advices to define business-relevant reference points. For example, if we place an adjustment point to begging of the "Test" activity, the name of the adjustment point should be "Ready to Test". In this study, we investigate the company's existing Software Project Management processes as variants of the base model and we add the adjustment points to the base process model (see Figure 8).

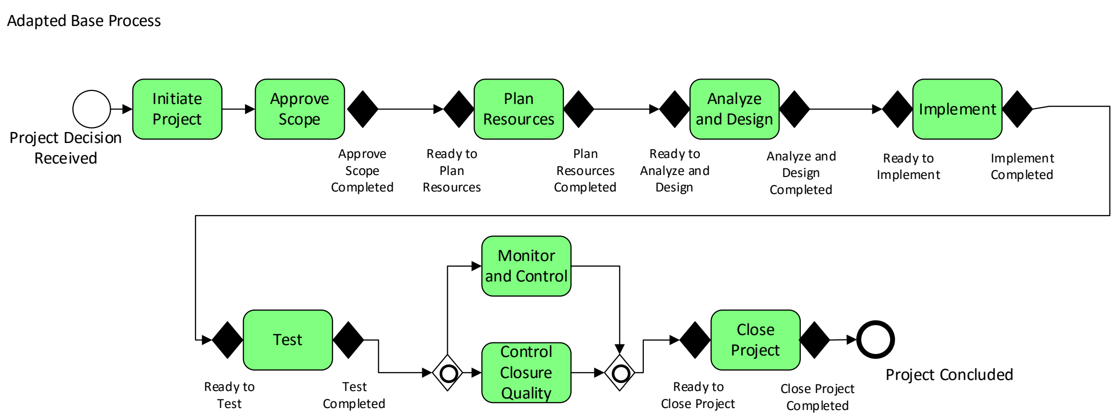

Figure 8. The base process model with adjustment points

Step 3 – Designing and Modelling the Options

So far, we have considered the concepts to the design of a base process model and adjustment points. During these steps the company processes are considered. (see Figure 1).

Now, according to the Provop approach, options of the process should be designed and modelled. To this end, the possible change operations for generating the variants based on the base process are investigated. Then, the conditional branches in the model are examined in order to determine that they are only variant specific or included in all variant models. Also, granularity selection is performed to increase efficiency of reusable options. While applying granularity selection the considered aspects are reusability, understandability and number of operations required for an option. For example, for Company 3, we could generate an option which consists of both Option 1 and Option 7 instead of splitting them. However, in that case, we could produce another option for removing Approve Scope from process model of Company 5. This level of granularity decreases both reusability and understandability while it increases the number of required operations. Therefore, we select more deepened granularity level and divided this option into two as Option 1 and Option 7.

The options designed in the consideration of the above concerns are provided below:

- **Option 1:**

  Context Rule: Company 3, 4-2, 5

  - DELETE Approve Scope
- **Option 2:**

  Context Rule: Company 4-1

  - INSERT Announce Initiation
    - *From Point:* Approve Scope Completed
    - *To Point:* Ready to Plan Resources
- **Option 3:**

  Context Rule: Company 3, 4-2

  - INSERT Assign PM(Project Manager)

- *From Point:* Approve Scope Completed
- *To Point:* Ready to Plan Resources

- **Option 4:**

  Context Rule: Company 2

  - o MODIFY Plan Resources as parallel activity
    - Gateway = Parallel gateway
  - o INSERT Prepare WBS parallel with Plan Resources
    - *From Point:* Ready to Plan Resources
    - *To Point:* Plan Resources Completed

- **Option 5:**

  Context Rule: Company 1, 2, 4-1, 5

  - o INSERT Approve Plan
    - *From Point:* Plan Resources Completed
    - *To Point:* Ready to Analyze and Design

- **Option 6:**

  Context Rule: Company 1, 2, 4

  - o INSERT Control Plan Quality
    - *From Point:* Plan Resources Completed
    - *To Point:* Ready to Analyze and Design
  - o MODIFY Approve Plan as optional activity
    - Gateway = Executive gateway

- **Option 7:**

  Context Rule: Company 3

  - o INSERT Control Plan Quality
    - *From Point:* Plan Resources Completed
    - *To Point:* Ready to Analyze and Design

- **Option 8:**

  Context Rule: Company 4-1, 5

  - o INSERT Control Design Quality
    - *From Point:* Analyze and Design Completed
    - *To Point:* Ready to Implement

- **Option 9:**

  Context Rule: Company 5

  - INSERT Control Implementation Quality
    - *From Point:* Implementation Completed
    - *To Point:* Ready to Test

- **Option 10:**

  Context Rule: Company 4-1

  - MODIFY Control Implementation Quality as optional activity
    - Gateway = Executive gateway

- **Option 11:**

  Context Rule: Company 3, 4-2, 5

  - DELETE Monitor and Control
  - DELETE Control Closure Quality

- **Option 12:**

  Context Rule: Company 1, 2, 4-1, 4-2

  - MODIFY Gateways = Parallel gateway
    - *From Point:* Test Completed
    - *To Point:* Ready to Close Project

- **Option 13:**

  Context Rule: Company 4-2

  - Option 11 is prerequisite to this option.
  - INSERT Approve Closure parallel with Create Asset
    - *From Point:* Test Completed
    - *To Point:* Ready to Close Project

- **Option 14:**

  Context Rule: Company 3, 4-1, 4-2

  - INSERT Create Asset
    - *From Point:* Close Project Completed
    - *To Point:* End Event

Step 4 – Configuring the Variants

For variant configuration, the Provop suggests the usage of three substeps. First, relevant options need to be selected to configure the relevant process variant. This can be done by asking users to manually choose specific variants, which is hard if there are a lot of options and specialized knowledge is required. To overcome the problem, the Provop suggests the definition of context rules by identifying, for each option, the context in which the options are applicable. In our case, the available knowledge on business drivers became useful to define the context. For each option, we identified the set of variants that are to be configured via this option. This can be seen in the list of options 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 relation between options, an option implying the usage of another one (A. Hallerbach et al., 2010a). We had an order constraint for options 6 and 7, as option 6 always needs to be applied before 7. We 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, we manually apply the set of options 4, 6, 7 and 12 to the base process Figure 8 to achieve the variant process of company 2 as in Fig. 9.

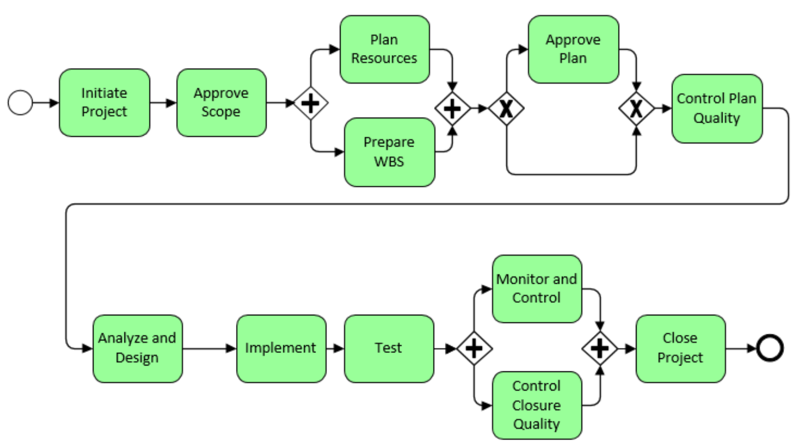

Fig. 9. Company 2 process model after configuration

CHAPTER 4

RESULTS

By applying the methods, we were able to answer our research questions which are determined at the beginning of the study. With the application of Decomposition Driven method, we aimed to address the question of how to manage the trade-off between modelling multiple variants of a business process together versus modelling them separately. Moreover, Decomposition Driven method can be mentioned as useful to address granularity management problem of process models. The low level modelling increases both comprehensibility and complexity while high level modelling decreasing both of them. With the help of the method, it can be decided which variants should be modelled together or separate. This provides a balance on complexity and comprehensibility trade-off and possible variations on process by proposing variation map (see Figure 4).

Results of the Decomposition Driven method application is interviewed with customer's process management responsibles. Proposed variation map for software project management process is evaluated and it is observed that proposed variation map is feasible for them especially with the proposed branches of "Initiate Project", "Plan Resources" and "Close Project" activities. They mentioned that, they have a classification for projects according to amount of planned man-day effort and with regard to defined project class, their initiation and closure activities are changing in practise but this variation is not modelled on process automation system. Moreover, they believe that, after modelling their variants separately they will be able to manage sub tasks more effectively and this will improve their both productivity and quality by decreasing rework and unnecessary monitoring efforts.

On the other hand, by applying Provop approach, similar change operations are grouped and the complexity to generate a variant decreases and the controllability of the model increases. Based on the options defined above and base process modelled in Figure 8, the Provop approach can be investigated and employed for the companies that require to apply business process management for their processes. It is obtained from the model that process variants can be generated via specified options in an easy and effective way. Also, this provides flexibility for companies while designing their processes. The flexibility of proposed process model can be obtained with the help of Provop method application results. According to results, number of user steps can be 6 in minimum and 18 in maximum while base model has 9 user steps. Moreover, 14 separate options are generated in order to handle many different process variation.

Some details of the results are grouped on Table 4.

Table 4. Results of the methods

Decomposition Driven Method	Provop Approach
In order to apply the method, total 22 hours spent in 5 sessions. With the application of the method, ~40 possible variations are handled with proposed variation map. Proposed variation map is accepted as feasible by customer. As a result of the real life application, reduced rework load and unnecessary monitoring effort with improved management perfection is expected.	✓ Total 15 hours spent in 4 sessions, in order to apply the method. ✓ As a result of the method, number of user steps can be 6 in minimum and 18 in maximum, while base model has exactly 9 user steps. ✓ 14 separate options are generated in order to handle many different process variations. ✓ The method provides enhanced controllability by enabling possible changes anytime.

CHAPTER 5

CONCLUSIONS

In this study, a case study is investigated based on the Decomposition Driven Method in (Milani et al., 2013) and the Provop approach in (A. Hallerbach et al., 2010a). The case study indicates that the process variants can be easily and effectively modelled by employing both of the methods. It is obtained from the models that the single-model approach is suitable for the processes of the companies analysed in the case study. It is expected that the process variant modelling improves the efficiency and the quality of the activities by reducing rework load and decreasing unnecessary monitoring effort. The case study concludes that the decomposition driven approach is suitable for the process variants, modelling of which requires an essential consideration of compact and clear design whereas the Provop approach is appropriate for the design of the process variants in which the controllability and manageability is a necessity.

In future work, we will completely apply the methods for low level processes of software project management as already initiated in current work. This will enable a thorough evaluation of the methods for hierarchical processes. Also, we plan to apply the methods to the demand request process, which even shows more variation with respect to customers and other factors. In parallel, 4S plans to start a gradual usage of variant modeling in its company. For this, new experts will be trained. Then, prototypes will be identified from the projects where the experts will use the outputs of this study to define processes of the new customers.

REFERENCES

Awad, A., Sakr, S., Kunze, M., & Weske, M. (2011). Design by selection: a reuse-based approach for business process modeling. In *Conceptual Modeling – ER 2011* (pp. 332–345). http://doi.org/10.1007/978-3-642-24606-7_25

Briand, L. C., Bunse, C., & Daly, J. W. (2001). A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. *Software Engineering, IEEE Transactions on*, *27*(6), 513–530. http://doi.org/10.1109/32.926174

Döhring, M., Reijers, H. A., & Smirnov, S. (2014). Configuration vs. adaptation for business process variant maintenance: An empirical study. *Information Systems*, *39*, 108–133. http://doi.org/http://dx.doi.org/10.1016/j.is.2013.06.002

Dumas, M. (2012). Consolidated Management of Business Process Variants BT. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), *Business Process Management Workshops* (p. 1). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-28115-0_1

Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., & Zugal, S. (2009). *Declarative versus Imperative Process Modeling Languages: The Issue of Understandability*. *Enterprise, Business-Process and Information Systems Modeling* (Vol. 29). http://doi.org/10.1007/978-3-642-01862-6_29

Gottschalk, F., Van Der Aalst, W. M. P., Jansen-Vullers, M. H., & La Rosa, M. (2008). Configurable Workflow Models. *International Journal of Cooperative Information Systems*, *17*(02), 177–221. http://doi.org/10.1142/S0218843008001798

Hallerbach, a, Bauer, T., & Reichert, M. (2008). Managing process variants in the process life cycle. *ICEIS 2008 - Proceedings of the 10th International Conference on Enterprise Information Systems*, *2 ISAS*, 154–161.

Hallerbach, A., Bauer, T., & Reichert, M. (2010a). Capturing Variability in Business Process Models: The Provop Approach. *Journal of Software Maintenance and Evolution: Research and Practice*, *22*(6-7), 519–546.

Hallerbach, A., Bauer, T., & Reichert, M. (2010b). Configuration and Management of Process Variants. In J. Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1 SE - 11* (pp. 237–255). Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-00416-2_11

Lind, M., & Goldkuhl, G. (2005). Designing business process variants. *… Third International Conference on Business Process …*, 27–28.

Milani, F., Dumas, M., Ahmed, N., & Matulevicius, R. (2013). Modelling Families of Business Process Variants: A Decomposition Driven Method. *CoRR*, *abs/1311.1*. Retrieved from http://arxiv.org/abs/1311.1322

Reijers, H., Mans, R. S. S., & van der Toorn, R. a. A. (2009). Improved model management with aggregated business process models. *Data & Knowledge Engineering*, *68*(2), 221–243. http://doi.org/http://dx.doi.org/10.1016/j.datak.2008.09.004

Rolon, E., Sanchez, L., Garcia, F., Ruiz, F., Piattini, M., Caivano, D., & Visaggio, G. (2009). Prediction Models for BPMN Usability and Maintainability. *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*. http://doi.org/10.1109/CEC.2009.53

Rosemann, M., & van der Aalst, W. M. P. (2007). A Configurable Reference Modelling Language. *Information Systems*, *32*(1), 1–23.

Wahler, K., & Küster, J. M. (2008). Predicting coupling of object-centric business process implementations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5240 LNCS, pp. 148–163).

Weidlich, M., & Weske, M. (2010). Structural and behavioural commonalities of process variants. In C. Gierds & J. Sürmeli (Eds.), *Proceedings of the 2nd Central-European Workshop on Services and their Composition, ZEUS 2010, Berlin, Germany, February 25--26, 2010* (Vol. 563, pp. 41–48). CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-563/paper5.pdf