Decomposition Driven Consolidation of Process Models

Fredrik Milani, Marlon Dumas, Raimundas Matulevičius

Institute of Computer Science, University of Tartu, Estonia, J. Liivi 2, 50409 Tartu, Estonia

{milani, marlon.dumas, rma}@ut.ee

Abstract. Oftentimes business processes exist not as singular entities that can be managed in isolation, but as families of variants that need to be managed together. When it comes to modelling these variants, analysts are faced with the dilemma of whether to model each variant separately or to model multiple or all variants as a single model. The former option leads to a proliferation of models that share common parts, leading to redundancy and possible inconsistency. The latter approach leads to less but more complex models, thus hindering on their comprehensibility. This paper presents a decomposition driven method to capture a family of process variants in a consolidated manner taking into account the above trade-off. We applied our method on a case study in the banking sector. A reduction of 50% of duplication was achieved in this case study.

Keywords: Variants, Variation Driver, Process Model Consolidation, Decomposition and Family of Process Variants.

1 Introduction

Every organisation, be it non-profit, governmental or private, can be conceived as a system where value is created by means of processes [15]. Oftentimes, these processes do not exist as singular entities but rather as a family of variants that need to be collectively managed [4, 19]. For example, an insurance company would typically perform the process for handling claims differently depending on whether it concerns a personal, vehicle or property claim [14]. Each of these processes for claims handling can be seen as variant of a generic claims handling process [6].

When it comes to modelling a family of process variants, one extreme approach is to model each variant separately. This *fragmented-model* approach [4] or "*multi-model approach*" [6] creates redundancy and inconsistency [6]. On the other hand, modelling multiple variants together in a *consolidated-model* approach [4] or "*single-model approach*"[6] leads to complex models that are hard to understand, analyse and evolve [6]. In addition to these comprehensibility and maintainability concerns, business drivers may come into play when determining whether multiple variants should be treated together or separately. Striking a trade-off between modelling each variant separately versus collectively in a consolidated manner is an open research question.

In this setting, the contribution of this paper is a decomposition driven method for modelling families of process variants in a consolidated manner. According to this

adfa, p. 1, 2011.

method, analysts start by incrementally constructing a decomposition of the family of process variants into sub-processes. At each level of the decomposition and for each sub-process, we determine if this sub-process should be modelled in a consolidated manner (one sub-process model for all variants or for multiple variants) or fragmented (one sub-process model per variant). This decision is based on two parameters: (i) the business drivers for the existence of a variation in the process; and (ii) the degree of difference in the way the variants produce their outcomes (syntactic drivers).

The rest of the paper is structured as follows. Section 2 introduces the conceptual foundation of our method. Next, Section 3 describes the proposed method step-by-step. Sections 4 and 5 discuss a case study where the proposed method was applied to consolidate a family of process models of a trading process in a bank. Finally, Section 6 discusses related work while Section 7 concludes and outlines future work.

2 Conceptual Foundation

The proposed method relies on two pillars: (i) a process decomposition method; and (ii) a decision framework for determining if variants of a process/sub-process should be modelled together or separately. Below we present these two frameworks in turn.

2.1 Decomposition of Process Models

A number of methods for process decomposition exist [7, 15, 18]. Although these methods differ in terms of the nomenclature and specific definitions of the various levels of the process decomposition, they rely on a common set of core concepts which we summarise below. A business process can be described at progressive levels of detail, starting from a top-level process, which we call the *main process* [18]. A main process is a process that does not belong to any larger process. The main process is decomposed into a number of *sub-processes* based on the concept of value chain introduced by Porter [7]. Sub-processes are processes on their own, and can be further decomposed into sub-processes until such a level where a sub-process consists exclusively of atomic activities (called tasks) that do not warrant further decomposition.

The above discussion refers to business processes, regardless of how they are represented. When modelling a business process, it is natural to model each of its sub-processes separately. Accordingly, the hierarchy of processes derived via process decomposition is reflected in a corresponding hierarchy of process models representing the sub-processes in this decomposition.

2.2 Business and Syntactic Drivers

By applying incremental decomposition on a family of process variants, we reduce the problem of determining whether a given process should be modelled in a fragmented or consolidated manner, to that of deciding whether each of its sub-processes should be modelled in a fragmented or consolidated manner. To guide this decision, we propose a decision framework based on two classes of variation drivers. On the one hand, there may be business reasons for two or more variants to be treated as separate processes (or as a single one) and ergo to model these variants separately (or together). On the other hand, there may be differences in the way two or more variants produce their outcomes, which make it more convenient to model these variants separately rather than together or conversely. We refer to the first type of drivers as *business drivers* while the second type of drivers is called *syntactic drivers*.

Business drivers can range from externally dictated ones such as legislative requirements to internal choices an organisation has made such as organisational divisions due to mergers for example [11]. By categorising the many business reasons of process variations into *classes of variation drivers*, a reduction in complexity is achieved [1]. This enables working with a few classes of drivers rather than a multitude of possible root causes [20]. To this end, we use our previously presented framework for classification of business drivers [10], which in turn is based on [15].

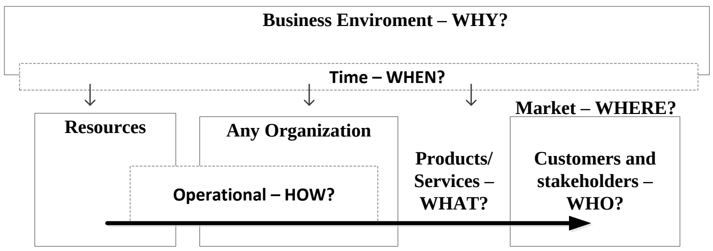


Fig. 1. Framework for classification of (business) variation drivers [10]

According to the adopted framework (Fig. 1), organisations operate within a context of external influences, to which they adapt their business processes in order to achieve competitive advantage. Organisations create an output by procuring resources in order to manufacture a product or a service (corresponding to *how* in Fig. 1). These products and services (*what*) are brought to a market (*where*) for customers (*who*) to consume. In some cases, an organisation might wish to adapt its processes depending on parameters in its external environment such as season (*when*). These factors lead to variations. Accordingly, the framework is based on the idea that drivers for process variation, based on their root causes, can be classified as *operational (how)*, *product (what)*, *market (where)*, *customer (who)* or *time (when)* drivers.

The second factor influencing whether to model two variants together or separately is the degree of difference in how the variants produce outcomes. If each variant was modelled separately, differences in the way variants produce outcomes would be reflected as differences between these separate models. If these models differ in significant ways, it is more convenient to keep them separate as consolidating them would increase complexity and reduce comprehensibility to such extent as rendering them of little use for users. However, if the variants are similar, it is more convenient to keep them together. Indeed, La Rosa *et.al.* [14] show empirically that the complexity of a consolidated model of two variants (measured by means of well-known complexity metrics such as size, density, structuredness and sequentiality) is inversely proportional to the similarity between the corresponding fragmented models, where similarity is measured by means of graph-edit distance between the process graphs. Hence, if we had a separate model for each variant, we could determine whether to merge them or not into a single model by computing their graph-edit distance. However, this requires that (i) the models of the separate variants are available; and (ii) that they are modelled using the same notation, at the same granularity and using the same modelling conventions and vocabulary. These assumptions are unrealistic in many practical scenarios. When these assumptions do not hold, we propose to assess the similarity between variants of a (sub-)process by means of subjective judgment of the expected differences between the separate models of these variants. Specifically, given two variants, we ask domain experts the question: How similar or different do you think the separate models of these two variants would be if they were available?

In the following section, we operationalise the concepts above in the form of a method for consolidated modelling of families of process variants.

3 Method

The method for process model consolidation consists of four steps as follows.

Step 1 – Model the main process.

The first step is to model the main process in terms of sub-processes as discussed in Section 2. The output of this step is a model of the main process in terms of sub-processes, but without any details of each of these sub-processes as illustrated in the top part in Fig. 2.

Step 2 – Identify business drivers and determine their relative strength.

In this step, the business drivers for variation in the process are elicited and classified in accordance with the framework described in Section 2 (see Fig. 1).

In this step, the business drivers for variation in the process are elicited and classified by asking two rounds of questions in accordance with the framework described in Section 2 (see Fig. 1). In the first round, questions are asked about the existence of drivers in each of the categories of the framework (such as how many markets or how many different customer segments are served). In the second set of questions, each of these categories of drivers are further clarified and refined. Concretely, this is achieved by means of a workshop or interview with business stakeholders.

Having identified the business drivers for the existence of variants in the process, a rating is assigned to each of these drivers to qualify their relative strength. The strength of a driver relative to a process is the perceived level of importance of managing the process variants induced by this driver separately, rather than together. The variants induced by a "very strong" driver are integral part of the business, whether for historical reasons or organizational reasons (e.g., different process owners or managers behind each variant). The variants induced by a "strong" driver are visible in the business, because for example the variants are supported by different IT systems or performed by different teams, though the differences are not ingrained in the business. The variants induced by a "somewhat strong" driver are considered to differ only at the level of minor details from a business perspective. The variants induced by "not strong" driver are completely irrelevant to the business; the variants should be treated as the same business process.

For example, a company that sells two similar services (e.g. individual and business travel insurance) in 10 countries with different sales and delivery channels, is likely to rate the driver ``geographic market'' as strong and the product driver as not strong. Meanwhile, a company delivering distinct products (e.g. motor and travel insurance) in a couple of similar markets is likely to rate the product driver as strong.

We propose a 4-point scale ("not strong'', "somewhat strong'', "strong'' and "very strong'') to rate the strength of business drivers but other scales could be chosen here. The output of this step is a variation matrix (see Fig. 2) wherein the rows correspond to business drivers (qualified by their relative strength) and the columns correspond to the sub-processes identified in step 1. A cell in this matrix lists the variants of a given sub-process (if any) induced by a given driver.

	Main Task 1	***Main Task 2***	***Main Task 3***
Driver 1 (Very Strong)	*Variant A* *Variant B*	*Variant A*	*Variant A*
Driver 2 (Very Strong)	*Variant C*	*Variant B*	*Variant B*

Fig. 2. Variation matrix

Step 3 – Perform similarity assessment for each sub-process of the main process.

In this step, we perform a similarity assessment for each subset of variants of each sub-process identified before. As discussed in Section 2, this similarity assessment is performed subjectively by domain analysts, given that we do not assume that detailed models of each sub-process are available for a detailed comparison. We use a 4-point scale for similarity judgements extensively used in the field of similarity assessment [24]: (1) identical, (2) very similar, (3) somewhat similar, and (4) not similar.

Step 4 – Construct the variation map.

From the previous steps, we know the strength of the business drivers and the degree of similarity between the variants of each sub-process induced by a driver. This information is used to manage the trade-off of modelling the variants in a consolidated versus fragmented manner. In making these decisions, the analyst will use the decision matrix depicted in Fig.3.

If the variants are very similar and there are no strong business drivers for variation (not strong or somewhat strong), then naturally the variants are modelled together. Conversely, if there are strong business drivers (strong or very strong) and the variants are syntactically different (somewhat similar or not similar), then they are modelled separately. If variants are similar and have strong business drivers, they are modelled together or separately depending on the current level in the process decomposition. At levels close to the main process, sub-process variants falling in this quadrant are modelled separately because the business driver for separating the variants prevails. Indeed, if the business driver is strong, it pre-supposes that the variants have different process owners and stakeholders and therefore the modelling effort has to be done separately for each variant. At lower levels of process decomposition, the business driver for modelling two variants separately weakens down and the incentive for sharing the modelling effort for variants increases. Therefore for sub-processes at lower levels of decomposition, the syntactic driver prevails, i.e. if these processes are similar, they are modelled together as a consolidated sub-process. Conversely, in the lower-right quadrant, variants of sub-processes at a high level of decomposition are modelled together, since these variants fall under the same ownership area and thus it makes sense to conduct a joint modelling effort for them. However, at the lower levels of decomposition, if two sub-process variants are not similar, the analysts can choose to model them separately. By high level of decomposition, we refer to level 3 (levels 1 and 2 refer to Business Model and the main process) of the value creation system hierarchy introduced by Rummler and Brache [15]. Using the same process architecture, low levels of decomposition refer to levels 4 and the lowest level 5.

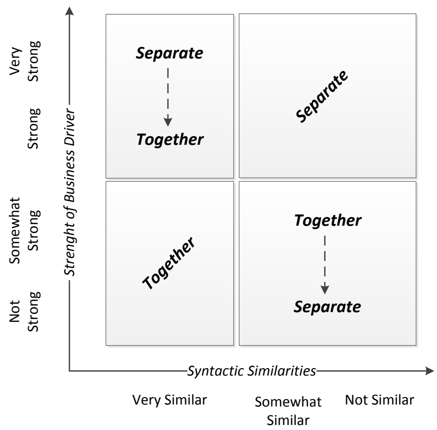


Fig. 3. Decision matrix for modelling variants separately or together

The output of this step is a variation map (see Fig. 4) showing the variants of each sub-process that ought to be modelled separately. The variation map contains one decision gateway per subset of variants of a sub-process that need to be modelled separately. If a sub-process does not have variants, it is not preceded by a gateway. Having constructed the variation map for the first level of process decomposition, we then consider each of the sub-process variants in the variation map in turn. Each of these sub-process variants is then decomposed into a lower-level process model and steps 2-4 are repeated at this lower level. In the decision matrix (Fig 3.), “very strong” and “strong” drivers are treated in the same manner as at this level, the variants have business impact. On the other hand, drivers that are “somewhat strong” or “not strong” are not considered to have business impact and therefore treated differently.

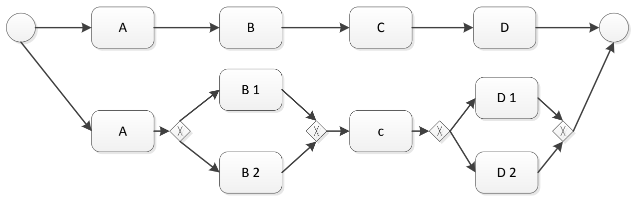


Fig. 4. Variation map

4 Case Study

4.1 Approach

The case study method allows researchers to investigate a phenomenon within its real-life context [16], particularly when the boundaries between what is studied and its context are unclear [23]. Case studies are often used for exploratory purposes, but they are also suitable for testing a hypothesis in a confirmatory study [5, 16] or to evaluate a method within the software and systems engineering domain [8]. These features make the case study method applicable to validate our proposed method.

When designing and creating a case study, Yin [23] argues for the necessity to define a research question. Our research question is: "*how can a family of process variants, based on managing its variations, be consolidated?"* Furthermore, Yin [23] states that there is a need for developing hypothesis. The purpose of our method is to produce consolidated process models that have less redundancy than a collection of fragmented models. Thus, our hypothesis is that *"if our method is applied on a family of process variants, then the same set of business processes can be modelled using fewer activities and sub-process models than if the same was done using a fragmented approach."* We do not expect, i.e. our alternative hypothesis is, that "*if applying our method, the size of the family of process variants is the same or larger in terms of total number of activities and sub-process models than with a fragmented approach*."

4.2 Setting

The case study setting is the foreign exchange (FX) and money market (MM) operations of a mid-sized European bank. FX covers financial products related to trade of international currencies. MM covers trade in short-term loans and deposits of financial funds between institutions. Currently, the bank is using a legacy system for managing these products. However it wants to replace it with an off-the-shelf system. For this purpose, the bank needs to elicit requirements, which primarily come from the corresponding business processes. The business processes had previously been modelled as separate process models by a team of consultants, several years before this case study. The existing models were flat (no decomposition had been made). Three of these models were for the variants of the process related to trading FX and MM with interbank counterparts and one for non-interbank clients who do not have an account with the bank. The bank aims at consolidating these process models prior to requirements elicitation. This case was selected as it fulfilled two main criteria we had defined namely (i) access to domain experts and (ii) process models that needed to be consolidated. The models were initially modelled as flowcharts.

4.3 Design

The case study (see Fig. 5) comprises 6 steps out of which the first 4 correspond to the steps in the consolidation method. The fifth step corresponds to constructing the consolidated models and the sixth step consists in verifying the consolidated models.

The method was applied in a workshop with 5 domain experts, led by the first author of this paper. In addition, two stakeholders from IT support were available for questions and clarifications. The workshop resulted in a variation map of the business processes. During the workshop, we first identified and modelled the main process for FX&MM trades (step 1). Then (step 2) we identified the variation drivers and determined their relative strength. We took the outputs of the first two steps to set up the variation matrix so we could populate the matrix with variants for each sub-process of the main process (step 3). Once the variation matrix was populated, we performed the similarity assessment, which gave us the input needed for constructing the variation map (step 4). We then consolidated the four end-to-end of process models (step 5) in accordance with the variation map. Finally the consolidated models were verified by domain experts (step 6) without involvement of any author of this paper.

The initial workshop took ca. 4 hours: one hour for modelling the main process, one hour for elicitation and classification of drivers, and two hours for similarity assessment. The construction of the variation map took ca. three hours. The consolidation of process models took ca. 80 man-hours. Verification of the consolidated models was done by the domain experts in a series of eight workshops of two hours each.

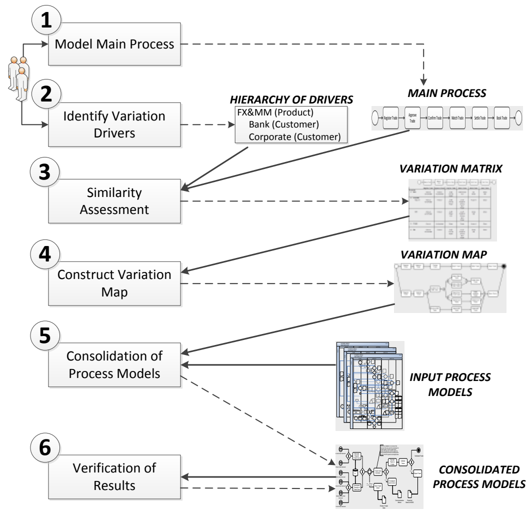


Fig. 5. Case Study Design

4.4 Execution

Step 1 - Model the main process of FX&MM trades.

In the first step, we modelled the main process for managing FX&MM trades (see Fig. 6). We started by asking what initiates the process and then, through a series of questions, modelled each step of the process until the end. We also clarified the purpose of each sub-process and summarised how they add value to the process. This step resulted in a model of the main process for FX&MM products (see Fig. 6).

The main process is initiated once an order is received. The first task is to "*register trade*" meaning entering the trade in the IS. The next task is "*approve trade*". Then, "*confirm trade*" takes place when the bank sends a confirmation of the trade details to the counterpart. Once the counterpart "*match trade*", i.e. agrees to the trade data, "*settle trade*" takes place (transfer of payment). The final task is "*book trade*" which is when the trade is booked in the accounting systems.

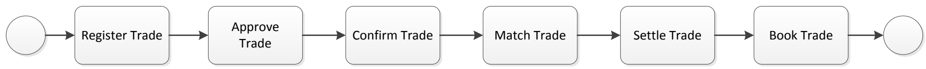


Fig. 6. Main process for managing FX & MM trades

Step 2 – Identify the variation drivers and determine their relative strength.

The second step (see Fig. 5) was to identify variation drivers of the process. We started by introducing the concept of variation drivers and the framework (see section 2) for their classification. We then gave some examples of variation drivers and asked if their business processes have occurrences of such variation drivers.

We observed that product and customer driven variations existed. The product driven variations were FX, MM and NDF (non-deliverable forward i.e. trading in restricted currencies). The customer driven variations were identified as Bank (other banks), Corporate (companies), Private (individuals) and Site (belonging to branches) clients. Furthermore, the corporate clients were of account (having an account agreement with the bank) or cash (do not have an account with the bank) client type.

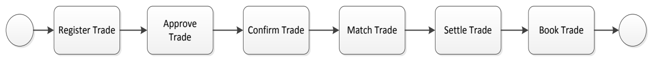
With the main variation drivers identified, we continued with determining their relative strength. Through discussions we understood that the product drivers were the strongest. It also became clear that FX & MM were similar enough to be treated as one. However, NDF is separate and on its own.

Finally we populated the variation matrix (see Fig. 7) from the drivers and the sub-processes identified in step 1 (see Fig. 6). First, we used the variation drivers and their relative strength to populate the first column of the variation matrix. Then, for each sub-process of the main process, such as *"match trade"*, we ask the domain experts, how is this process performed? For instance, for an FX trade done with another bank, the ways to match the trades are either Intellimatch (in-house trade by trade matching) or CLS (a centralised intra-bank platform). We thus enter these two variants in the matrix under sub-process *"match trade"* and for customer type *"bank"* (see Fig. 7).

Step 3 – Perform similarity assessment for each sub-process of the main process.

We performed the similarity assessment by visiting each cell of the variation matrix in turn. For example, the variation matrix shows that corporate and site clients have the same variants for matching a trade. We asked the domain experts to grade the level of similarity of these variants from 1 (identical) to 4 (not similar). The results showed that all SWIFT trades are very similar. The same applied to platform, online and paper. We also observed that matching in bulk (when several trades are matched at once) is very different compared to SWIFT, platform, online and paper.

Having established the degree of similarity among the corporate, private and site clients, we enquired about similarities between CLS and Intellimatch when the counterpart is a bank. These differed significantly compared to how trades are matched for non-bank counterparts. This step resulted in identifying two main variants for matching when the counterpart is a bank (Intellimatch and CLS) and two variants when trading with non-bank counterparts (bulk versus single-trade match).



	Register Trade	*Approve Trade*	*Confirm Trade*	*Match Trade*	*Settle Trade*	*Book Trade*
FX & MM **1. Bank**	*Manual* *Automated*	*Manual* *Automated*	*Swift* *Online* *Paper*	*IntelliMatch* *CLS*	*CLS* *Gross*	*Gross* *Net*
2. Corporate Account	*Manual* *Automated*	*Manual*	*Swift* *Online* *Paper*	*Swift* *Platform* *Online* *Bulk* *Paper*	*Account*	*Gross*
Cash	*Manual* *Automated*	*Manual*	*Swift* *Paper* *CLS*	*Swift* *Platform* *Online* *Bulk* *Paper*	*Gross* *Net*	*Gross*
3. Private	*Manual*	*Automated*	*Paper*	*Paper*	*Account*	*Gross*
4. Site	*Manual* *Automated*	*Manual*	*Swift* *Online* *Paper*	*Swift* *Platform* *Online* *Bulk* *Paper*	*Gross* *Net*	*Gross*

Fig. 7. Filled Variation Matrix (NDF excluded due to space limitation)

Step 4 – Construct variation map.

As input for step 4, we know the strength of the drivers and the perceived level of similarity of the variants for each sub-process of the main process. For instance, we had four separate process models of *"register trade"*. These sub-processes did not have a strong business driver and were similar. Referring to the decision framework (Fig. 3), we modelled them together. Conversely, there are two models describing "confirm trade", one for FX/MM and one for NDF trades. These sub-processes have very strong drivers and are not similar, and thus are modelled separately in accordance with the decision framework. The resulting variation map for each sub-process is depicted in Fig 8.

Step 5 - Consolidation of Process Models.

The original process models had been modelled as flat end-to-end process models. As a first step, we divided these models into sub-processes in accordance with the decomposition identified in step 3. That gave us four hierarchical process models, one for FX traded gross, one for FX traded via CLS, one for MM and one for corporate clients. In addition to these four process models, there were two additional processes described as text, one for NDF and one for bulk matching, which we modelled diagrammatically as part of the consolidation effort.

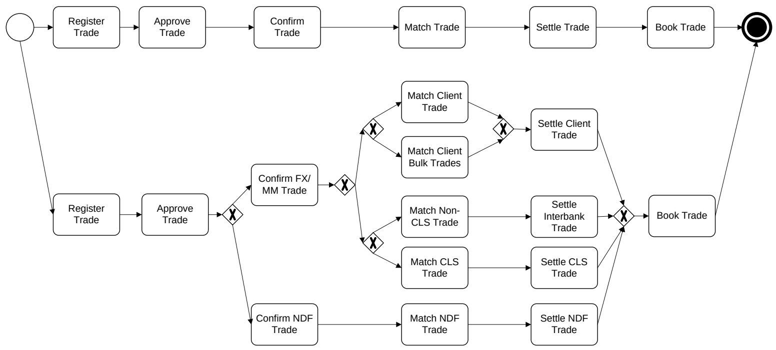


Fig. 8. Variation map for FX&MM main process

For each task of the main process, we compared and consolidated them in accordance with the variation map. We sought clarification from the domain experts and IT stakeholders when needed. The input process models had not been regularly updated with changes in the business processes during the past 3 years and therefore we observed minor discrepancies. We updated the consolidated process models accordingly.

Step 6 - Verification of results by domain experts.

Once the process models had been consolidated, they were verified by domain experts. An initial verification was made by one domain expert who examined the consolidated process models and noted minor issues (corrected by the researcher). Then, in a series of 8 workshops, the domain expert verified the models in detail with other four domain experts. Adjustments to the consolidated models were made by the coordinating domain expert during these workshops. After all workshops, the domain experts were asked about the usefulness of the models in terms of comprehensibility and if they will use the models for evaluating off-the-shelf systems. They stated that the consolidated models are easier to understand (compared to the input process models), already used for evaluating one vendor and they intend to reuse the models to evaluate future vendor products.

5 Findings

5.1 Comparison of Input versus Consolidated Process Models

As mentioned above, the original process models had been modelled flat (no decomposition into sub-processes). In order to make them comparable with the models produced after consolidation, we split each flat process model into sub-processes following the same sub-process structure that resulted from the consolidation. In this way, the input models and the consolidated ones are comparable.

The input process models did not include NDF and bulk matching. These processes had only been partially documented in textual format prior to the consolidation. During the consolidation effort, these two processes were modelled as well. However, to make the input and the consolidated process models comparable, we do not take into account NDF and bulk matching in any of the statistics given below.

The input process models contain 35 sub-process models and 210 activity nodes (not counting gateways and artefacts such as data objects or data stores). Out of these, 75 activity nodes were duplicate occurrences (an activity occurring N times across all sub-process models counts as N duplicate occurrences). Thus, it can be said that the duplication rate in the input models is 36 %. Note that the 35 sub-process models in the input were distinct models, although some of them had strong similarities.

The consolidated models contain 17 sub-process models and 149 activity nodes of which 22 are duplicate occurrences, corresponding to 15 % duplication. Thus the consolidated models contain 30% less activity nodes, half of the sub-process models and half of the duplication rate relative to the original model. These observations (summarised in Table 1) support the hypothesis of the case study formulated above.

Table 1. Size metrics before and after consolidation

Variable	**Input**	**Consolidated**
Main Process Models	4	1
Sub-Process Models	35	17
Activity Nodes	210	149
Duplicate Activity Occurrences	75	22
Duplication rate	36 %	15 %

It is reasonable to assume that the complexity of the process models will increase during consolidation since additional gateways are introduced to capture differences between multiple variants of a sub-process model. This trade-off between reduction in duplication and increase in complexity has been observed for example in [14].

To measure the impact of consolidation on complexity, we use the coefficient of network complexity (CNC) metric. CNC is the ratio between the number of arcs and the number of nodes. This simple metric has been put forward to be suitable for assessing the complexity of process models [2]. The input process models had a total of 350 arcs and 280 nodes (210 activity nodes and 60 gateways and start/end events). This gives a CNC of 1.25. The consolidated process models consist of 320 arcs and 240 nodes (149 activity nodes and 81 gateways and events) giving a complexity factor of 1.33. Thus, there is a marginal increase in complexity as a result of consolidation. This should be contrasted to the significant reduction in size and duplication.

5.2 Threats to Validity

Case studies come with several inherent threats to validity, particularly regarding external validity and reliability [16]. External validity concerns the extent to which the findings can be generalised beyond the setting of the study. Our method has been applied on one case study, and accordingly, the results are limited in the extent they can be generalised. As the results are dependent on the domain experts and the purpose of the study, there is also a limitation to repetitiveness. Hence, our method is replicable but results may vary due to the reasons above. It should be underscored though that the case study was conducted in an industrial setting and involved workshops with domain experts. Reliability concerns the level of dependency between the results and the researcher i.e. would the same results be produced if another researcher conducted the study? This threat was to some extent tackled by having verifications by the domain experts without the presence of the researcher. In addition, the consolidated models were used in a four-day workshop with a supplier of off-the-shelf solution to investigate the extent to which the solution could satisfy their needs.

6 Related Work

The presented study falls under the scope of process model consolidation. Process model consolidation is related to *process standardisation*, which seeks to merge several variants of a process into one standard process [12], as opposed to merging the models of the processes for documentation purposes. One of the steps in process standardisation is to identify suitable processes that can be standardised. Proposed methods to achieve this include assessing process complexity [17] or applying user-centred design approaches such as work practice design, which helps to identify candidate processes based on how employees perform their responsibilities [9]. Since our method focuses on model consolidation and not process standardisation, it does not touch upon the organizational change management issues that are central in standardisation. This having been said, process model consolidation and process standardisation share common concerns. In particular, we foresee that the business variation drivers identified via our method could serve as input for standardisation decisions.

Related to process standardisation is *process harmonisation*, which seeks to achieve a reduction in the differences between variants of a process [12] rather than aiming at one standardised process. Romero *et.al.* [13] propose a model-based technique to determine an optimal level of process harmonisation based on the identification of so-called influencing factors (i.e. variation drivers) and based on similarity metrics between the models of the individual variants. Their method however requires that the process models are represented at a low level of details. In contrast, our method can be applied when the process variants are not modelled at the same level of detail or when the models are incomplete (e.g., some processes have not been modelled or not modelled at the same level or using the same conventions as others).

Alternative methods to process model consolidation include process model merging methods such as the one proposed by La Rosa et.al [14]. In these methods, multiple variants of a process model are merged into a single model, essentially by identifying duplicate fragments and representing these fragments only once in the merged model. This and similar approaches have the limitation of being based purely on syntactic similarities across process models. They do not take into account business drivers. Also, their aim is to build a single consolidated model, but this might sometimes not be desirable since the consolidated model might be overly large and complex. Our method can be seen as an approach to answer the question of when it makes sense to merge, and when it is better to keep separate models. Thus, our contribution is upstream with respect to automated process model merging methods.

Other related work includes process model refactoring [3], where the aim is to rewrite process models in order to improve their comprehensibility, maintainability or reusability, but without altering their execution semantics. Weber *et.al.* [22] propose a catalogue of "smells" in process models that could be treated as candidates for refactoring. Dijkman *et.al.* [3] developed a technique that measures consistency of activity labels, degree and type of process overlap to identify refactoring opportunities in repositories of process models. Our method can be seen as identifying refactoring opportunities in a family of process models by optimising their structure. However, we take the business drivers for variation into consideration, whereas the methods mentioned above [3, 22] focus on semantic and structural aspects of process models.

Finally, our work is related to variability modelling in software product lines, where methods based on feature diagrams have been studied extensively [21]. However, feature diagrams take the viewpoint of the product and are geared towards describing product variations. Our method transposes ideas behind feature diagrams to process modelling. Indeed, variation matrices and variation maps can be seen as integrated views of process models and the features that drive variations in these models.

7 Conclusion

We have presented a decomposition driven method for consolidating models of process variants. In comparison to existing approaches, which handle consolidation on the basis of syntactic differences, we consider also business drivers for variation. This reduces the risk of distancing the models from the processes they aim at representing.

We have validated the method by applying it on an industrial case study. Although not fully generalisable, the findings show that the method can help analysts to significantly reduce duplication in a family of process variants, with a relatively small amount of effort and a minor penalty on model complexity.

Currently, we are working on applying the method on a second case study. We also plan to develop a semi-automated tool for construction of variation matrices and similarity assessment of variants from logs (process mining). This would combine our method with BPM tools and cover additional aspects such as traceability.

Acknowledgement. This research was supported by the European Social Fund via the Doctoral Studies and Internationalisation Programme – DoRa.

8 References

1. Bailey, K.: Typologies and Taxonomies: An introduction to Classification Techniques. Sage, (1994)

2. Cardoso. J., Mendling, J., Neumann, G., Reijers, H.A.: A Discourse on Complexity of Process Models. In: Proceedings of BPM Workshops pp. 115-126, Springer (2006)
3. Dijkman, R.M., Gfeller, B., Kuster, J.M., Völzer, H.: Identifying refactoring opportunities in process model repositories. Information & Software Technology, 53(9):937–948 (2011)
4. Dumas, M.: Consolidated Management of Business Process Variants. In: BPM Workshops (2), pp 1-1, Springer (2011)
5. Flyvbjerg, B.: Five misunderstandings about case-study research. Qualitative inquiry 12(2): 219-245 (2006)
6. Hallerbach, A.: Bauer, T., Reichert, M.: Configuration and Management of Process Variants. In: Handbook on Business Process Management 1, pp. 237–255, Springer (2010)
7. Harmon, P.: The Scope and Evolution of Business Process Management. In: Handbook on Business Process Management 1 pp. 37–81, Springer (2010)
8. Kitchenham, B., Pickard, L., and Pfleeger, S.: Case studies for method and tool evaluation. IEEE Software 12(4):52–62 (1995)
9. Ludwig, H., Rankin, Y., Enyedi, R., Anderson, L.C.: Process Variation Analysis Using Empirical Methods: A Case Study. In 9th Intl Conference on Business Process Management, Clermont-Ferrand, France, pp. 62-65, Springer (2011).
10. Milani, F., Dumas, M., Matulevičius, R.: Identifying and Classifying Variations in Business Processes. In Enterprise Business Process and Information Systems Modeling, pp. 136-150, Springer (2012)
11. Pascalau, E., Rath, C.: Managing Business Process Variants at eBay. In 2nd International Workshop on BPMN, Potsdam, Germany, Springer (2010)
12. Richen, A., Steinhorst, A.: Standardization or harmonization? You need both. In BPTrends Newsletter (November 2005)
13. Romero, H., Dijkman, R., Grefen, P., Weele, A.: Harmonization of Business Process Models. Business Process Management Workshops (2), pp. 13-24, Springer (2012)
14. La Rosa, M, Dumas, M, Uba, R, & Dijkman, R. M. Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transactions on Software Engineering and Methodology, to appear (2013)
15. Rummler, G., Brache, A.: Improving Performance: How To Manage the White Space on the Organization Chart. The Jossey-Bass Management Series (1995)
16. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2):131-164 (2009)
17. Schafermeyer, M., Grgecic, D., Rosenkranz, C.: Factors Influencing Business Process Standardization: A Multiple Case Study. Proceedings of the 43rd Hawaii International Conference on System Sciences, IEEE (2010)
18. Sharp, A., McDermott, P.: Workflow modeling: tools for process improvement and applications development, Artech House (2009)
19. Simidchieva, B.I., Clarke, L.A., Osterweil, L.J.: Representing process variation with a process family. In International Conference on Software Processes, Minneapolis MN, USA, pp. 109–120. Springer (2007)
20. Smith E.E., Medin D.L.: Categories and concepts. Harvard University Press (1981)
21. Svahnberg, M., van Gurp, J., Bosch, J.: A taxonomy of variability realization techniques. Softw., Pract. Exper. 35(8): 705-754 (2005)
22. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry 62(5):467–486 (2011)
23. Yin, R.K.: Case study research: Design and methods. 5th edition. Sage Publications, (2008)
24. Young, F. W., Hamer. R.M.: Multidimensional scaling: History, theory, and applications. Lawrence Erlbaum Associates (1987)