Matthias Weidlich and Jan Mendling and Mathias Weske
Propagating Changes between Aligned Process Models


This version is available at: [http://epub.wu.ac.at/3497/](http://epub.wu.ac.at/3497/)

Available in ePubWU: April 2012

ePubWU, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.

This document is the version accepted for publication and — in case of peer review — incorporates referee comments. There are minor differences between this and the publisher version which could however affect a citation.
Propagating Changes between Aligned Process Models

Matthias Weidlich\textsuperscript{a,*}, Jan Mendling\textsuperscript{b}, Mathias Weske\textsuperscript{c}

\textsuperscript{a} Technion – Israel Institute of Technology, Technion City, 32000 Haifa, Israel
\textsuperscript{b} Wirtschaftsuniversit\"at Wien, Augasse 2-6, A-1090 Vienna, Austria
\textsuperscript{c} Hasso-Plattner-Institute, University of Potsdam, Prof.-Dr.-Heinert-Str. 2-3, D-14482 Potsdam, Germany

Abstract

There is a wide variety of drivers for business process modelling initiatives, reaching from organisational redesign to the development of information systems. Consequently, a common business process is often captured in multiple models that overlap in content due to serving different purposes. Business process management aims at flexible adaptation to changing business needs. Hence, changes of business processes occur frequently and have to be incorporated in the respective process models. Once a process model is changed, related process models have to be updated accordingly, despite the fact that those process models may only be loosely coupled. In this article, we introduce an approach that supports change propagation between related process models. Given a change in one process model, we leverage the behavioural abstraction of behavioural profiles for corresponding activities in order to determine a change region in another model. Our approach is able to cope with changes in pairs of models that are not related by hierarchical refinement and show behavioural inconsistencies. We evaluate the applicability of our approach with two real-world process model collections. To this end, we either deduce change operations from different model revisions or rely on synthetic change operations.

Keywords: Change Propagation, Model Synchronisation, Behavioural Analysis, Process Model Alignment

1. Introduction

Process models are an important mechanism for capturing requirements at the early stage of system development. An appealing characteristic of them is that they are easily understood not only by software engineers, but also by business people. Business process modelling languages like BPMN\textsuperscript{1} have been explicitly designed to leverage communication between both these groups of system stakeholders. Indeed, process models are increasingly utilised for pure business purposes, such as identifying potential for process improvements and facilitating organisational redesign.

The usage of process models for different purposes, within different departments, and at different degrees of detail in the same organisation poses considerable challenges for model maintenance. The result is often that there exist several models for the same real-world process, which are maintained either by the business units or the IT department, and which take a different perspective on the process. Most notably, different people work with and edit these models. Therefore, there is a strong need for techniques to efficiently transfer a change being made, e.g., in the technical process model to the business process model, and vice versa.

Change management for process models has recently emerged as a research topic entailing several non-trivial conceptual problems. According to\textsuperscript{2}, change management involves three steps, (1) detection of differences, (2) analysis of their relations (implications, contradictions), and (3) resolution of differences. In this article, we focus on the third step. Related research often assumes that two corresponding process models are in a strict refinement relation (also called decomposition)\textsuperscript{3,4,5}, where the more detailed model takes a white-box view on a set of black-box activities of the coarse-granular model. The problem in this context is that such a refinement is hardly found in practice. In a survey with 69 members of the SAP developers network, it was found that refinement-consistent notions like trace equivalence are often not suitable for assessing cases of complex correspondences\textsuperscript{6}. Case studies illustrate the relevance of 1: n and overlapping correspondences in practice\textsuperscript{7,8}.

In this article, we address the problem of propagating changes between two semantically overlapping process models for which elementary and complex correspondences have been identified. We explicitly formalise the problem in a way that overlapping and complex correspondences can be handled in the same way as hierarchical refinement. The general idea of our approach was presented in\textsuperscript{9}. This article develops our technique in formal detail and covers several aspects that were not included in our prior work. First, we incorporate complex correspondences in this work,
which is important for applying the technique in a practical setting. Second, we show how the technique is applied for inconsistent alignments, which is essential for increasing the applicability of the approach. Finally, we present a thorough evaluation of our technique. Our evaluation covers two aspects in particular, applicability in terms of change operations observed in practice and scalability for large process models.

The remainder of the article is structured as follows. Section 2 discusses the background of our work. We describe the essential concepts of a process model alignment and introduce a running example for the article. Section 3 defines the formal concepts of our approach, namely Petri nets, alignments, and behavioural profiles. Section 4 formalises our technique for change propagation. It utilises the notion of a change region, which can be narrowed down based on the behavioural profiles and correspondences. Section 5 provides an evaluation of the change propagation technique. Section 6 discusses related work before Section 7 concludes the article.

2. Background

To illustrate the background of our work, this section first clarifies terminology of process model alignments. We discuss the essential concepts and their relations. Then, we exemplify these concepts with an illustrative scenario for change propagation between process models.

2.1. Process Model Alignments

Process models are often used on different levels of granularity and with varying focus in the same organisation. Most prominently, the challenge of keeping semantically related process models in line shows up between business-oriented and system-oriented departments in a company. Typically, processes are described on a business level at a coarser degree of granularity and with a focus on activities that are conducted by humans. On the level of system specification, processes are captured in greater detail. Automatic activities, read and write operations to database, or web services calls are additionally defined while tasks that are purely executed by humans might be neglected. Therefore, there is a need for the definition of correspondences between a business-oriented and a system-oriented process model to keep the evolution of both in line.

The concepts used for describing process model alignment are inspired by the according notions in the field of schema and ontology matching [10 11]. We illustrate the main concepts and their relations in Figure 1. Process models are built of process model elements, e.g., activities or events. A correspondence relates two non-empty sets of elements to each other. Here, all elements of a set necessarily belong to the same process model, and two sets that form a correspondence must relate to distinct process models. We distinguish two types of correspondences. Elementary correspondences relate singleton sets of elements to each other with a certain confidence. For complex correspondences, in turn, at least one of the related sets comprises two or more elements. From a conceptual point of view, a complex correspondence is formed by multiple elementary correspondences, i.e., the Cartesian product of the associated sets of elements is captured by elementary correspondences. In other words, a certain set of elementary correspondences – the set is maximal with respect to set inclusion – is interpreted as a complex correspondence. Hence, it suffices to base the concept of an alignment on elementary correspondences. An alignment is built from a set of elementary correspondences for which the first elements all refer to one process model and the second elements all refer to another model. This way, an alignment relates two process models to each other.

A correspondence is a generic concept to express a relation between activities. It is defined on the model syntax. Still, semantics of a correspondence are induced by the behaviour of the aligned process models. Consider a complex 1:n correspondence between two process models that relates an activity (A) to the set of activities \( \{ (X), (Y), (Z) \} \). Semantics of this correspondence is deduced from the behavioural dependencies between the activities \( \{ (X), (Y), (Z) \} \). For instance, those may (1) be executed sequentially with no option to skip their execution, (2) be mutually exclusive, or (3) potentially in concurrency following an inclusive decision point. Then, the execution of activity (A) in the first model corresponds to the execution of (1) activities \( \{ (X), (Y), (Z) \} \) in sequential order, (2) of one of the three activities, or (3) at least one of the three. These three types have in common that, assuming trace semantics, we would expect that (a) the execution of a single activity like (A) would require at least one of the corresponding activities to be observed in the execution trace, and that (b) the execution of several of the \( n \) activities would result in observing the corresponding singular activity only once. In compliance with this requirement, correspondences can be potentially overlapping. Note that this interpretation of correspondences differs from those in
the area of schema and ontology matching. A correspondence between data schemas or ontologies is also defined structurally, it relates schema or ontology entities to each other. However, semantics need to be defined explicitly for all correspondences of an alignment, which yields a mapping. As an example, a complex 1:n correspondence between data entities that represent integer values may be defined, such that the value in one schema equals the sum of values of n data entities in the other schema.

Correspondences are given implicitly in some use cases. Consider, for instance, a process model that is derived from a reference model by customisation. However, in many cases, an alignment needs to be defined explicitly. Correspondences are either defined manually or constructed by (semi-)automated matching techniques. The latter are inspired by a large body of work on solving the matching problem for data schemas and ontologies, see [10] [11]. For process models, textual matching is often enriched with graph-based techniques that exploit the model structure, or the behavioural techniques that leverage the execution semantics. Regarding the former, sub-graph isomorphism and the graph-edit distance have been utilised [13] [8] [14]. To leverage the behaviour of process models, different behavioural models may be utilised. For instance, the similarity of state spaces, of n-gram representations of the language induced by a process model, or of behavioural abstractions may guide the matching procedure [13] [10] [14] [17]. As for the area of schema and ontology matching, however, there are only a few approaches, for instance, the ICoP framework [15], that aim at the detection of complex correspondences between process models.

2.2. An Example Alignment

We illustrate the concepts of process model alignments with Figure 2. It depicts two process models captured as Petri net systems. Both models describe a request handling process. We observe various commonalities between the models. In both cases, the request is either created and approved, or an existing request is fetched for processing. Also, tracking of the request status and the collection of request responses is handled in a similar way. However, we also observe differences between the models. Those relate to the level of modelling granularity and the model coverage. For instance, fetching an existing request is modelled with one activity in model (a), i.e., transition (B), whereas the functionality is split up in transitions (B1) and (B2) in model (b). With respect to the model coverage, e.g., the activity of requesting a notification in model (b), transition (L), is without counterpart in model (a).

For this article, we assume an alignment between process models to be given. In Figure 2, an alignment is visualised by the colouring of Petri net transitions and the transition labels. All transitions with a grey-striped background have been aligned by correspondences that relate the transitions with the same letter identifier in the label to each other. Groups of transitions that are part of complex correspondences are highlighted. For instance, there is a correspondence that relates transition (B) in model (a) to transitions (B1), (B2) in model (b). Also, there is a complex correspondence between transitions (C1), (C2) in model (a) and transitions (C3), (C4) in model (b). This correspondence is an example for a complex n:m correspondence. Due to a different distribution of functionality (the approval of the request is handled differently), we relate only the two sets of transitions to each other, but not any pair of single transitions.

Assume that model (a) has just been changed by inserting an additional activity, represented by transition X. Our approach supports change propagation by identifying a region in model (b), in which the change may be applied. This enables a process analyst to first assess whether change propagation seems to be appropriate. If so, the region sets the scope for updating the model accordingly.

3. Formal Preliminaries

This section clarifies formal preliminaries for our work. Section 3.1 first introduces net systems, the formalism used in this article to capture process models. Then, we elaborate on alignments of net systems in Section 3.2. Section 3.3 defines behavioural profiles as an abstraction of trace semantics of net systems.

3.1. Net Systems

Petri net systems are a widely used formalism for the description of concurrent systems in general, and business processes in particular. Certain classes of net systems have been utilised for process modelling over a decade [10]. Petri net based formalisations have been presented for most common process description languages, such as the Business Process Execution Language (BPEL), the Business Process Model and Notation (BPMN), Event-Driven Process Chains (EPCs), and UML activity diagrams (UML ADs) [20] [21] [22] [23]. A survey of Petri net based formalisations for BPMN, EPCs, and BPEL was presented in [24]. This work also highlights the limitations of Petri net based formalisations with respect to OR-join constructs and exception handling.

Introductions to Petri nets can be found in [25]. In the following, we recall basic definitions.

Definition 1 (Petri net). A Petri net, or a net, $N = (P,T,F)$ has finite disjoint sets $P$ of places and $T$ of transitions, and a flow relation $F \subseteq (P \times T) \cup (T \times P)$.

We write $X = (P \cup T)$ for all nodes of a net and identify the flow relation $F$ with its characteristic function on the set $(P \times T) \cup (T \times P)$. Further, we denote predecessors of a node $x \in X$ by $\text{pr}(x) = \{ y \in X \mid F(y,x) = 1 \}$ and successors by $\text{su}(x) = \{ y \in X \mid F(x,y) = 1 \}$. A tuple $N^* = (P',T',F')$ is a subnet of a net $N = (P,T,F)$, iff $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$.

We define semantics of Petri nets as follows.

Definition 2 (Net semantics). Let $N = (P,T,F)$ be a net.
As discussed in Section 2.1 and outlined in Figure 1, the correspondence relation to capture elementary correspondences for net systems. Since transitions represent the

\[ M : P \rightarrow N_0 \] is a marking of \( N \), where \( M(p), p \in P \), returns the number of tokens in place \( p \).

- For any transition \( t \in T \) and for any marking \( M \) of \( N \), \( t \) is enabled in \( M \), denoted by \((N,M)[t]\), iff \( \forall p \in \bullet M(p) \geq 1 \).

- If \( t \in T \) is enabled in \( M \), then it can fire, which leads to a new marking \( M' \), denoted by \((N,M)[t](N,M')\). The new marking \( M' \) is defined by \( M'(p) = M(p) - F(p,t) + F(t,p) \), for each place \( p \in P \).

- Let \( M_0 \) be a marking of \( N \). If \((N,M_0)[t_1](N,M_1)\ldots(N,M_{n-1})[t_n](N,M_n)\) are transition firings, then a sequence of transitions \( \sigma = t_1\ldots t_n, n \in N_0 \), is a firing sequence leading from \( M_0 \) to \( M_n \).

- For any two markings \( M \) and \( M' \) of \( N \), \( M' \) is reachable from \( M \) in \( N \), denoted by \( M' \in [N,M] \), iff there exists a firing sequence \( \sigma \) leading from \( M \) to \( M' \). Since \( \sigma \) can be the empty sequence, it holds \( M \in [N,M] \) for every \( M \) of \( N \).

- A net system, or a system, is a pair \( S = (N,M_0) \), where \( N \) is a net and \( M_0 \) is a marking of \( N \). \( M_0 \) is called the initial marking of \( S \).

3.2. Alignments of Net Systems

Having discussed basic concepts of alignments between process models in Section 2.1, we formally define those concepts for net systems. Since transitions represent the active parts of a business process, we define correspondences between transitions of net systems. We introduce a correspondence relation to capture elementary correspondences and a confidence function to capture their quality. As discussed in Section 2.1 and outlined in Figure 1, the notion of a correspondence – a relation between two sets of transitions – is derived from elementary correspondences.

**Definition 3** (Alignment). Let \( S_1 = (N_1,M_1), N_1 = (P_1,T_1,F_1) \), and \( S_2 = (N_2,M_2), N_2 = (P_2,T_2,F_2) \), be net systems.

- A correspondence relation \( \sim \subseteq T_1 \times T_2 \) associates corresponding transitions of both systems to each other.

- Let \( T'_1 \subseteq T_1 \) and \( T'_2 \subseteq T_2 \) be two sets of transitions such that \( T'_1 \times T'_2 \subseteq \sim \). Let \( T'_1 \) and \( T'_2 \) be maximal with respect to set inclusion, i.e., \( \forall t_1 \in (T'_1 \setminus T'_1) \), \( \forall t_2 \in (T'_2 \setminus T'_2) \), and \( \forall t_2 \in (T'_2 \setminus T'_2) \). Then, \( c = (T'_1,T'_2) \) is referred to as a correspondence and we also write \( T'_1 \sim T'_2 \).

- A confidence function \( \zeta : \sim \rightarrow [0,1] \) assigns confidence values between zero and one to pairs of corresponding transitions.

- A correspondence \( c = (T'_1,T'_2) \in \sim \) is called elementary, iff \( |T'_1| = |T'_2| = 1 \), and complex otherwise.

- Two correspondences \( c_1 = (T'_1,T'_2) \) and \( c_2 = (T''_1,T''_2) \) are overlapping, iff \( T'_1 \cap T''_1 \neq \emptyset \) or \( T'_2 \cap T''_2 \neq \emptyset \).

- A correspondence relation \( \sim \subseteq T_1 \times T_2 \) is overlapping, iff it induces at least two overlapping correspondences. Otherwise, it is non-overlapping.

- An alignment is a tuple \((\sim, \zeta) \) with \( \sim \) being a correspondence relation and \( \zeta \) being a confidence function for \( \sim \).

Given a correspondence relation \( \sim \subseteq T_1 \times T_2 \) that is defined between net systems \( S_1 = (N_1,M_1), N_1 = (P_1,T_1,F_1), \) and \( S_2 = (N_2,M_2), N_2 = (P_2,T_2,F_2) \), we use a short-hand notation to refer to all aligned transitions \( T_1^\sim \subseteq T_1 \) of \( S_1 \)
with \( T_2^- = \{ t_1 \in T_1 \mid \exists t_2 \in T_2 \ [ t_1 \sim t_2] \} \). The set of aligned transitions \( T_2^- \subseteq T_2 \) of \( S_2 \) is defined analogously.

For our example depicted in Figure 2, we discussed two complex correspondences, one between transition (B) of system (a) and transitions \( \{(B1),(B2)\} \) in system (b) and one between transitions \( \{(C1),(C2)\} \) in system (a) and transitions \( \{(C3),(C4)\} \) in system (b). The latter correspondence is manifested in the correspondence relation between both systems by four entries, i.e., \( (C1) \sim (C3), (C1) \sim (C4), (C2) \sim (C3), \) and \( (C2) \sim (C4) \). Even though most correspondences in Figure 2 are non-overlapping, there is also one exception. There is one correspondence that is defined between transition (G) in system (a) and transition (GI) in system (b), whereas another correspondence relates transition (I) in system (a) to the set \( \{(GI),(I1),(I2)\} \) of transitions in system (b). Both correspondences are overlapping, since transition (GI) is part of two correspondences. The meaning of overlapping correspondences is explained by a functional overlap. In our example, a part of transition GI corresponds to the transition (G) in system (a), whereas the remaining part, together with transitions \( \{(I1),(I2)\} \), corresponds to transition (I) in system (a).

### 3.3. Behavioural Profiles

A behavioural profile is an abstraction of the behaviour of a net system [20]. It captures the order of potential occurrence of transitions by relations over pairs of transitions. The definition of a behavioural profile is grounded on the notion of weak order. Two transitions \( t_1 \) and \( t_2 \) are in weak order, if there exists a firing sequence starting in the initial marking in which \( t_1 \) occurs before \( t_2 \).

**Definition 4 (Weak Order).** Let \( S = (N,M_0) \) be a net system, \( N = (P,T,F) \), and \( T' \subseteq T \) a set of transitions. A pair of transitions \( (x,y) \in (T' \times T') \) is in the weak order relation \( \succ \) over \( T' \), if there exists a firing sequence \( \sigma = t_1, \ldots, t_n \) with \( (N,M_0)[\sigma] \) and indices \( j,k \in N \), \( 1 \leq j < k \leq n \), for which holds \( t_j = x \) and \( t_k = y \).

Using the notion of weak order, we define three relations forming the behavioural profile.

**Definition 5 (Behavioural Profile).** Let \( S = (N,M_0) \) be a net system with \( N = (P,T,F) \) and \( T' \subseteq T \) a set of transitions. A pair of transitions \( (x,y) \in (T' \times T') \) can be in the following profile relations:

- The strict order relation \( \succ \), iff \( x \succ y \) and \( y \not\succ x \).
- The exclusiveness relation \( \prec \), iff \( x \prec y \) and \( y \not\prec x \).
- The interleaving order relation \( \parallel \), iff \( x \parallel y \) and \( y \parallel x \).\( B = \{\sim, +, \parallel\} \) is the behavioural profile of \( S \) over \( T' \).

Apparently, the relations of the behavioural profiles are mutually exclusive. Along with reverse strict order \( \sim^{-1} = \{(x,y) \in (T' \times T') \mid (y,x) \in \sim\} \), the relations partition the Cartesian product of transitions over which they are defined [26]. Exclusiveness and interleaving order are symmetric relations, whereas strict order is antisymmetric and irreflexive. Hence, a transition is either exclusive to itself or in interleaving order to itself. The former means that a transition may be fired at most once. The latter implies that a transition may be fired multiple times. The behavioural profile is a behavioural abstraction. Thus, two systems that are trace equivalent show equal behavioural profiles, whereas the equivalence of behavioural profiles does not imply equal trace semantics.

For illustration purposes, consider the system (a) of Figure 2. It holds \( (A) + (B) \), since both transitions cannot occur together in a firing sequence. For transitions (A) and (X), we observe strict order, \( (A) \sim (X) \). Further, potential concurrent enabling of transitions (X) and (G) leads to interleaving order as their relation in the behavioural profile, \( (X) \parallel (G) \). With respect to self-relations, it holds \( (A) + (A) \) and \( (H) \parallel (H) \).

Computation of behavioural profiles is done efficiently for the class of sound free-choice workflow systems. The free-choice property relates to the net structure and requires that conflict (a place is succeeded by multiple transitions) and synchronisation (a transition is preceded by multiple places) do not interfere, cf., [27]. Constructs of common process description languages can be mapped to free-choice net systems except for a few exceptions, see also [24]. A net system has a workflow structure, if it contains two special places: one to mark the initialisation and one to mark the completion of processing. As such, this class of net systems has been particularly advocated to be used for modelling business processes [19]. Soundness is a correctness criteria often used for process models. It guarantees absence of behavioural anomalies, e.g., deadlocks [25]. For sound free-choice workflow systems, the behavioural profile can be deduced from the net structure. This enables computation of the behavioural profile in \( O(n^3) \) time with \( n \) as the number of nodes of the respective net [26].

### 4. Change Propagation using Behavioural Profiles

This section introduces our approach to propagate changes. In Section 4.1, we will discuss the general idea of the approach using the example presented earlier. We elaborate on the notion of change in Section 4.2. Then, we define the major steps of our approach in detail in Sections 4.3 to 4.7.

#### 4.1. Overview

An overview of our approach is given by Figure 3. We assume that two process models, Original Model 1 and Original Model 2, have been aligned by correspondences. Given a change in the first model, we isolate a change region in the second model. In this way, a process analyst can quickly assess the necessity to propagate the change. If change propagation seems to be appropriate, the change region spots the position where to update the process model.

The general idea of our approach can be summarised as follows. A change operation in the first process model, also
This captures not only insertion of transitions or subnets, suffices to consider one of the three affected transitions as operation for process models. However, in our context, it reengineering and, therefore, represents a common change block. Such parallelisation may be applied during process that transforms a sequence of transitions into a parallel change transition. Figure 4 depicts a change operation addition or removal of nodes may be traced back to a single transition. Also, high-level changes that go beyond a single transition can be addressed by our approach. Such a change just requires selecting one transition as the change transition and to identify the transitions for which the relations of the behavioural profile are not affected by the change implementation. Hence, even complex change operations such as the parallelisation of a whole sequence of transitions can be addressed by our approach. Such change just requires selecting one transition as the change transition and to identify the transitions for which the relations of the behavioural profile are not changed by the parallelisation operation.

4.2. A Change & a Change Region

As a first step, we discuss how a change materialises in a net system and introduce the notion of a change region. In general, every change of the syntax of a net system may be considered for change propagation. Against the background of process models that serve different purposes, the question of what constitutes a change may be answered on a more abstract level. That is, multiple changes of the syntax of a net system are often semantically related and jointly realise a high-level change operation. Albeit defined in the context of process instance adaptation, common high-level changes have been classified as change patterns [22].

For our approach, we abstract from the actual implementation of a change. We assume that a change can be localised by a dedicated transition, called change transition. This captures not only insertion of transitions or subnets, but is also applicable for other changes. For instance, removal of a transition or a subnet may also be localised by a single transition. Also, high-level changes that go beyond addition or removal of nodes may be traced back to a single change transition. Figure 4 depicts a change operation that transforms a sequence of transitions into a parallel block. Such parallelisation may be applied during process reengineering and, therefore, represents a common change operation for process models. However, in our context, it suffices to consider one of the three affected transitions as a change transition, whereas the other affected transitions are neglected when propagating the change. For the example in Figure 4, therefore, change propagation would rely on transitions (D) and (E).

Since we exploit the relations of the behavioural profile for change propagation, we also define a notion of a consistent change. This notion is defined between the transitions of the original net system and the net system obtained after applying the change operation, i.e., Original Model 1 and Changed Model 1 in Figure 3. A consistent change is defined relative to a subset of transitions, for which the relations of the behavioural profile are not affected by the change implementation. Hence, even complex change operations such as the parallelisation of a whole sequence of transitions can be addressed by our approach. Such a change just requires selecting one transition as the change transition and to identify the transitions for which the relations of the behavioural profile are not changed by the parallelisation operation.

Definition 6 (Change Transition, Consistent Change). Let \( S = (N, M) \), \( N = (P, T, F) \), be a net system. Implementation of a change yields a system \( S' = (N', M') \), \( N' = (P', T', F') \), with \( T \cap T' = \emptyset \). The transition \( t \in T' \) representing the change is referred to as the change transition. The change from \( S \) to \( S' \) is consistent over \( T_c \subseteq T \cap T' \), if the behavioural profile \( B \) of \( S \) over \( T_c \) coincides with the behavioural profile \( B' \) of \( S' \) over \( T_c \).

For our initial example in Figure 3, the aforementioned change operation involved inserting a transition and two flows in system (a). This change is localised by the change transition (X). Further, the change is consistent with respect to all transitions of the net system. The relations of the behavioural profile have not been changed by the insertion of transition (X) for any pair of transitions.

To propagate a change from the source net system to the target net system, we identify a change region. Formally, such a change region is captured by a subnet of the target net system. We introduced the standard notion of a subnet in Section 3. For a net \( N = (P, T, F) \), a place-bordered subnet may be characterised by a set of transitions \( T' \subseteq T \). The subnet induced by \( T' \) is defined as \( N' = (P', T', F') \).
with $P' = \{ p \in P \mid \exists t \in T' [ p \in \bullet t \lor p \in t \bullet] \}$ and $F' = F \cap ((P' \times T') \cup (T' \times P'))$. Our approach builds upon change regions that are subnets induced by a set of transitions. For brevity, therefore, we refer to a set of transitions as a change region.

4.3. Consistent Alignments

To localise a change region in one system for a change in another system, we exploit the relations of the behavioural profile for corresponding transitions. We may only exploit correspondences for which the profile relations are consistent. Inconsistencies in these relations would lead to contradicting information on how to localise the change region. Hence, we first define a notion of consistency for an alignment. As an auxiliary concept, we introduce type equivalence of profile relations of different net system. This equivalence is defined for all relations of the behavioural profile along with the reverse strict order relation.

**Definition 7 (Type Equivalence of Profile Relations).** Let $B_1 = \{ \sim_1, +_1, ||_1 \}$ and $B_2 = \{ \sim_2, +_2, ||_2 \}$ be behavioural profiles. Two relations $R_1 \in B_1 \cup \{ \sim_1^{-1} \}$ and $R_2 \in B_2 \cup \{ \sim_2^{-1} \}$ are type equivalent, denoted by $R_1 \simeq R_2$, if and only if
\[
\begin{align*}
& o R_1 = \sim_1 \land R_2 = \sim_2, \\
& o R_1 = \sim_1^{-1} \land R_2 = \sim_2^{-1}, \\
& o R_1 = +_1 \land R_2 = +_2, \text{ or} \\
& o R_1 = ||_1 \land R_2 = ||_2.
\end{align*}
\]

Using type equivalence of profile relations, we define a notion of consistency of an alignment. Given two aligned net systems, we require that the profile relations observed for two transitions of different correspondences in one model are type equivalent for all pairs of corresponding transitions in the other model. We neglect self-relations of aligned transitions. Those are not leveraged for change propagation. Not every alignment, however, can be expected to satisfy our notion of consistency. On the one hand, different perspectives on a common business process may result in behavioural deviations between process models that serve different purposes. On the other hand, alignments that are derived automatically using matching techniques may contain false positives, correspondences established between activities with similar labels, which actually relate to different tasks.

We cope with inconsistencies using the notion of a weak consistent sub-alignment. Given an alignment between two net systems, a weak consistent sub-alignment is a restriction of the correspondence relation and the confidence function, such that the resulting alignment is weak behavioural profile consistent.

**Definition 8 (Weak Behavioural Profile Consistency).** Let $S_1 = (N_1, M_1)$ and $S_2 = (N_2, M_2)$ be net systems with $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$, be net systems, and $B_1 = \{ \sim_1, +_1, ||_1 \}$ and $B_2 = \{ \sim_2, +_2, ||_2 \}$ their behavioural profiles. Let $R_1 \in B_1 \cup \{ \sim_1^{-1} \}$ and $R_2 \in B_2 \cup \{ \sim_2^{-1} \}$. A correspondence relation $\sim \subseteq T_1 \times T_2$ is weak behavioural profile consistent, iff for all transition pairs $(t_x, t_y) \in (T_1' \times T_2')$, $t_x \neq t_y$, and transitions $t_s, t_t \in T_2'$, $t_s \neq t_t$, $t_x \sim t_s$, $t_y \sim t_t$, it holds that either (1) $(t_x t_r t_y \land t_s R_2 t_t) \Rightarrow R_1 \simeq R_2$ or (2) $t_x \sim t_s$ and $t_y \sim t_t$.

Referring to the alignment between the net systems in Figure 3, we observe that most profile relations between pairs of aligned transitions in one system are type equivalent to those observed for the corresponding transitions. Nevertheless, the alignment is not weak behavioural profile consistent. There are deviations in the two behavioural profiles that relate to transition $(E)$ in both systems. For instance, it holds $(F) \sim (E)$ and $(I) \sim (E)$ in system $(a)$, whereas we observe $(F) \| (E)$, $(GI) \sim (E)$, $(I1) \sim (E)$, and $(I2) \sim (E)$ in model $(b)$.

4.4. Extraction of a Consistent Sub-Alignment

We argued that only consistent alignments may be exploited for change propagation. Not every alignment, however, can be expected to satisfy our notion of consistency. On the one hand, different perspectives on a common business process may result in behavioural deviations between process models that serve different purposes. On the other hand, alignments that are derived automatically using matching techniques may contain false positives, correspondences established between activities with similar labels, which actually relate to different tasks.

We cope with inconsistencies using the notion of a weak consistent sub-alignment. Given an alignment between two net systems, a weak consistent sub-alignment is a restriction of the correspondence relation and the confidence function, such that the resulting alignment is weak behavioural profile consistent.

**Definition 9 (Weak Consistent Sub-Alignment).** Let $S_1 = (N_1, M_1)$ and $S_2 = (N_2, M_2)$ be net systems with $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$, and $(\sim, \zeta)$ an alignment with $\sim \subseteq T_1 \times T_2$. The tuple $(\sim', \zeta')$ is a weak consistent sub-alignment of $(\sim, \zeta)$, iff $\sim' \subseteq \sim$, $\forall (t_x, t_y) \in \sim'$, $\zeta'(t_x, t_y) = \zeta(t_x, t_y)$, and $\sim'$ is weak behavioural profile consistent.

An inconsistent alignment may be restricted in different ways to yield a weak consistent sub-alignment. It seems reasonable to rely on sub-alignments that comprise a maximal number of elementary correspondences. However, there may be multiple sub-alignments that show the maximal number of elementary correspondences. Further, identification of maximal alignments is an optimisation problem, which is computationally hard. Search algorithms, such as
as the \(A^*\)-algorithm \(30\), may be used to find maximal sub-alignments. Besides the number of elementary correspondences, the sum of the confidence values related to the correspondences may guide the selection of a sub-alignment. Again, this can be seen as an optimisation problem. As a non-optimal sub-alignment only lowers the amount of information exploited to support change propagation, we do not consider this to be a severe problem. Hence, non-optimal heuristics like steepest ascent \(31\) may be applied to select a weak consistent sub-alignment.

As discussed earlier, we assume a change represented by a change transition \(t\) to be consistent over a certain set of transitions \(T'\) of the source net system. All elementary correspondences \((t_1, t_2) \in \sim\) that relate to inconsistent transitions, \(t_1 \in T'\), are removed from the alignment before a weak consistent sub-alignment is selected. Hence, the set of correspondences that may be exploited for change propagation is reduced twice – once based on inconsistencies in the source system caused by the change implementation and once based on inconsistencies between the aligned net systems.

Consider, again, our running example depicted in Figure 2. We discussed that the behavioural profiles of both systems, \(N_1\) and \(N_2\), are used to support change propagation, we do not consider this to be a severe problem. Hence, non-optimal heuristics like steepest ascent \(31\) may be applied to select a weak consistent sub-alignment.

As discussed earlier, we assume a change represented by a change transition \(t\) to be consistent over a certain set of transitions \(T'\) of the source net system. All elementary correspondences \((t_1, t_2) \in \sim\) that relate to inconsistent transitions, \(t_1 \in T'\), are removed from the alignment before a weak consistent sub-alignment is selected. Hence, the set of correspondences that may be exploited for change propagation is reduced twice – once based on inconsistencies in the source system caused by the change implementation and once based on inconsistencies between the aligned net systems.

Consider, again, our running example depicted in Figure 2. We discussed that the behavioural profiles of both systems, \(N_1\) and \(N_2\), are used to support change propagation, we do not consider this to be a severe problem. Hence, non-optimal heuristics like steepest ascent \(31\) may be applied to select a weak consistent sub-alignment.

\[
\text{As discussed earlier, we assume a change represented by a change transition } t \text{ to be consistent over a certain set of transitions } T' \text{ of the source net system. All elementary correspondences } (t_1, t_2) \in \sim \text{ that relate to inconsistent transitions, } t_1 \in T', \text{ are removed from the alignment before a weak consistent sub-alignment is selected. Hence, the set of correspondences that may be exploited for change propagation is reduced twice – once based on inconsistencies in the source system caused by the change implementation and once based on inconsistencies between the aligned net systems.}
\]

4.5. Boundary Transition Reduction

Initially, we consider all transitions of the target system to form the change region, i.e., to induce a place-bordered subnet in which the according change shall be realised. The change region is narrowed by reducing the set of transitions. The first reduction requires the identification of the closest aligned transitions that are in strict order with the change transition in the source system. We refer to these transitions as preceding or succeeding boundary transitions. Here, closest means that there must not be any aligned transitions between a boundary transition and the change transition, which is also in strict order with the latter. This implies that these notions are non-transitive. We capture boundary transitions, either of the source system or of the target system, as follows.

**Definition 10 (Boundary Transitions).** Let \(S_1 = (N_1, M_1)\), \(N_1 = (P_1, T_1, F_1)\), and \(S_2 = (N_2, M_2)\), \(N_2 = (P_2, T_2, F_2)\), be net systems, \(B_1 = \{\sim_{1, +, 1}, ||1\}\) and \(B_2 = \{\sim_{2, -}, +, ||2, 2\}\) their behavioural profiles, and \(\sim \subseteq T_1 \times T_2\) a weak behavioural profile consistent correspondence relation. Let \(t_x\) be a change transition of a consistent change in \(S_1\).

- The set of preceding boundary transitions \(PT_{S_1} \subseteq T_1^*\) of \(S_1\) contains all aligned transitions that directly precede \(t_x\) in strict order, \(PT_{S_1} = \{t_1 \in T_1^* \mid t_1 \sim_1 t_x \wedge \forall t_2 \in T_1^* \{ t_2 \sim_1 t_x \Rightarrow t_1 \not\sim t_2 \}\}\). The set of succeeding boundary transitions \(SBT_{S_1}\) of \(S_1\) is defined accordingly.

- The set of preceding boundary transitions \(PBT_{S_2} \subseteq T_2^*\) of \(S_2\) contains all corresponding transitions for preceding boundary transitions that are not succeeded in strict order by another corresponding transition, \(PBT_{S_2} = \{t_1 \in T_2^* \mid \exists t_2 \in PBT_{S_1} \{ t_2 \sim_1 t_1 \wedge \forall t_3 \in T_2^* \{ t_2 \sim t_3 \Rightarrow t_1 \not\sim t_3 \}\}\}. The set of succeeding boundary transitions \(SBT_{S_2}\) of \(S_2\) is defined accordingly.

We illustrate the concept of boundary transitions using our running example depicted in Figure 2. System (a) shows three preceding boundary transitions, namely \((B), (C1)\), and \((C2)\). All of them are in strict order with the change transition \((X)\). Transition \((H)\) is the only succeeding boundary transition in system (a). Note that transition \((E)\) does not qualify to be a succeeding boundary transition, as we removed the respective correspondence from the alignment to ensure weak behavioural profile consistency.

Using boundary transitions, the change region in the target system is narrowed. The idea behind is that strict order between boundary transitions and the change transition in the source system has to be preserved for the boundary transitions and the change region in the target system. A transition is removed from the change region, if it meets one of the following requirements.

- It precedes a preceding boundary transition in strict order.
- It succeeds a succeeding boundary transition in reverse strict order.
- It is exclusive to or in interleaving order with a boundary transition.

Formally, we define this reduction of the change region as follows.

**Definition 11 (Boundary Transition Reduction).** Let \(S_1 = (N_1, M_1)\), \(N_1 = (P_1, T_1, F_1)\), and \(S_2 = (N_2, M_2)\), \(N_2 = (P_2, T_2, F_2)\), be net systems, \(B_1 = \{\sim_{1, +, 1}, ||1\}\) and \(B_2 = \{\sim_{2, -}, +, ||2, 2\}\) their behavioural profiles, and \(\sim \subseteq T_1 \times T_2\) a correspondence relation. Let \(t_x\) be a change transition in \(S_1\). \(PBT_{S_2}\) and \(SBT_{S_2}\) the sets of preceding and succeeding boundary transitions of \(S_2\), and \(B_2 = PBT_{S_2} \cup SBT_{S_2}\). The boundary transition reduction creates a change region \(T_C\), such that

\[
T_C = T_2 \setminus PT \setminus ST \setminus BT
\]

with

\[
PT = \{t_1 \in T_2 \mid \exists t_2 \in PBT_{S_2} \{ t_1 \sim_2 t_2 \}\} \\
ST = \{t_1 \in T_2 \mid \exists t_2 \in SBT_{S_2} \{ t_2 \sim_2 t_1 \}\} \\
BT = \{t_1 \in T_2 \mid \exists t_2 \in B_2 \{ (t_1 + t_2) \vee (t_1 ||2 t_2) \}\}
\]

Again, we refer to the running example for illustration. Figure 2 depicts the target system, i.e., system (b) in Figure 2 and highlights the change region once boundary transition reduction has been applied to the set of all transitions of
the net system. For instance, transitions \((A)\) and \((R)\) are removed as they precede a preceding boundary transition, or succeed a succeeding boundary transition, respectively. Transition \((Q)\) is removed from the change region, as it is exclusive to the boundary transition \((H)\). Further, all boundary transitions are not part of the obtained region either. Technically, they are removed by the reduction since their self-relation is either exclusiveness or interleaving order.

4.6. Inter-Boundary Transition Reduction

The second kind of reduction of the initial change region exploits transitions that are in strict order with preceding and succeeding boundary transitions in the source system, but not with the change transition. We refer to these transitions as \textit{inter-boundary transitions}. Those transitions are between the boundary transitions in terms of strict order and show exclusiveness or interleaving order with respect to the change transition. To keep the formalisation concise, we capture inter-boundary transitions only for the target system.

\textbf{Definition 12 \textit{(Inter-Boundary Transitions).}} Let \(S_1 = (N_1, M_1), N_1 = (P_1, T_1, F_1), \) and \(S_2 = (N_2, M_2), N_2 = (P_2, T_2, F_2), \) be net systems, \(B_1 = \{\sim_1, +_1, ||_1\}\) and \(B_2 = \{\sim_2, +_2, ||_2\}\) their behavioural profiles, and \(\sim \subseteq T_1 \times T_2\) a weak behavioural profile consistent correspondence relation. Let \(t_x\) be a change transition of a consistent change in \(S_1\) and \(PBT_2\) and \(SBT_2\) the sets of preceding and succeeding boundary transitions of \(S_2\).

- The set of \textit{exclusive inter-boundary transitions} \(EIT \subseteq T_2^\sim\) of \(S_2\) contains all aligned transitions that precede and succeed the boundary transitions and for which the corresponding transitions are exclusive to the change transition, \(EIT = \{t_1 \in T_2^\sim \mid \forall t_2 \in T_1^\sim \; [t_2 \sim t_1 \Rightarrow t_2 +_2 t_x] \land \forall t_p \in PBT_2, t_x \in SBT_2 \; [t_p \sim_2 t_1 \sim_2 t_x]\}\).

- The set of \textit{interleaving inter-boundary transitions} \(IIT \subseteq T_2^\sim\) of \(S_2\) contains all aligned transitions that precede and succeed the boundary transitions and for which the corresponding transitions are in interleaving order with the change transition, \(IIT = \{t_1 \in T_2^\sim \mid \forall t_2 \in T_1^\sim \; [t_2 \sim t_1 \Rightarrow t_2 + || t_x] \land \forall t_p \in PBT_2, t_x \in SBT_2 \; [t_p \sim_2 t_1 \sim_2 t_x]\}\).

For our example, see Figure 2 we identify one inter-boundary transition in system \((b)\), i.e., transition \((F)\).

It is located between the boundary transitions with respect to strict order. Further, the corresponding transition \((F)\) in system \((a)\) is exclusive to the change transition. Note that transitions \((GI)\), \((I1)\) and \((I2)\) do not qualify as inter-boundary transitions. Even though there is a strict order dependency from all preceding boundary transitions to these transitions, they are in interleaving order with the succeeding boundary transition \((H)\).

As for boundary transition reduction, inter-boundary transitions are leveraged to narrow the change region in the target system. We remove a transition from the change region, if it meets one of the following requirements.

- It is not exclusive to one of the exclusive inter-boundary transitions.
- It is not in interleaving order with one of the interleaving inter-boundary transitions.
- It is an exclusive or interleaving inter-boundary transition.

We define the inter-boundary transition reduction as follows.

\textbf{Definition 13 \textit{(Inter-Boundary Transition Reduction).}} Let \(S_1 = (N_1, M_1), N_1 = (P_1, T_1, F_1), \) and \(S_2 = (N_2, M_2), N_2 = (P_2, T_2, F_2), \) be net systems, \(B_1 = \{\sim_1, +_1, ||_1\}\) and \(B_2 = \{\sim_2, +_2, ||_2\}\) their behavioural profiles, and \(\sim \subseteq T_1 \times T_2\) a correspondence relation. Let \(t_x\) be a change transition of a consistent change in \(S_1\) and \(EIT\) and \(IIT\) the sets of exclusive and interleaving inter-boundary transitions of \(S_2\). The inter-boundary transition reduction creates a change region \(T_C\), such that

\[T_C = T_2 \setminus EIT \setminus IIT \setminus ET \setminus IT\]  

with

\[ET = \{t_1 \in T_2 \mid \exists t_2 \in EIT \; [\; (t_1, t_2) \notin +_2\;}\]  

\[IT = \{t_1 \in T_2 \mid \exists t_2 \in IIT \; [\; (t_1, t_2) \notin ||_2\;]}\]

Applying the inter-boundary reduction to our examples yields the change region highlighted in the target system in Figure 6. Four transitions, \{(M), (O), (P), (Q)\}, are exclusive to the inter-boundary transition \((F)\). The latter is not part of the change region. Hence, the change region is the place-bordered subnet induced by the transitions \{(M), (O), (P), (Q)\}.

4.7. Derivation of the Change Region

Having discussed the elementary steps of our approach, the complete algorithm to derive the change region is shown.
in Algorithm 1. Both reductions narrow the change region induced by a set of transitions in the target system. Thus, the final change region is the intersection of their results. Although both reductions are independent of each other, the result of one reduction may be used as the input for the second reduction. Both reductions, however, are not redundant. This has been illustrated by our running example. The change regions highlighted in Figure 5 and Figure 6 overlap only partly.

For implementing the change in the target system, the self-relation of the change transition in the source system may be exploited. Whether the change transition is exclusive to itself or in interleaving order to itself provides additional information on how to adapt the target system. Interleaving order hints at potential multiple occurrences of the transition related to the change in the source system. Such behaviour may be mirrored in the target system.

5. Experimental Evaluation

We evaluated our approach to propagate changes with two experiments. Both experiments relied on real-world process models and complement each other since they investigate different applicability aspects. First, we focus on applicability in terms of change operations observed in practice. We took process models that have been created as part of the BPM Academic Initiative (BPM AI). For these models, revisions are available, which allowed us to extract changes that have been applied during model evolution. The second experiment focusses on the merits of our approach in the large scale and utilised a model collection from a health insurance company. Models of this collection are significantly larger than those of the first set. For all pairs of semantically related models of this collection, we propagated them to the other model.

In this section, we describe both experiments in terms of the background, experimental setup, and obtained results. We close this section with a discussion of the results and a reflection on the limitations of our approach.

5.1. Models of the BPM Academic Initiative

The BPM Academic Initiative (BPM AI) is a joint effort of academic and industry partners that offers a process modelling platform for teaching and research purposes. As part of the BPM AI, an online model repository and a model editor are provided by the industry partner Signavio. More than 4500 people worldwide from roughly 450 universities are using the platform provided by the BPM AI. The snapshot of the BPM AI model collection used in our experiment dates from early 2011, see also [32]. It comprises

![Diagram](image-url)
1903 process models in different languages, such as the Business Process Model and Notation (BPMN), Event Driven Process Chains (EPCs), and Petri net systems. Most models were created by students as part of their curriculum.

Out of the models of the BPM AI, we identified a cluster of semantically related process models. We compared the labels of activities (or functions in EPCs, transitions in Petri net systems) of all process models. We considered only labels that have more than five characters in order to ignore non-sense and very abstract labels. Two labels have been found similar, if their string edit distance similarity was above the threshold of 0.8. The string edit distance compares two labels based on the minimal number of atomic operations on characters (insert, delete, replace) needed to transform one label into the other. All models that were grouped in one cluster were required to show at least five activities with similar labels. Even though this approach is rather simplistic, we were able to isolate several clusters of process models. We conducted our experiment with the largest of those clusters, comprising 11 process models captured in BPMN. Most of these models describe how assignments are conducted at universities. We transformed all models into Petri net systems following existing work on net-based formalisations of BPMN.

### Experimental Setup

To extract realistic changes that need to be propagated between the process models, we considered their past revisions. For the chosen process models, between one and 12 revisions have been available, 4.1 revisions on average. All revision were transformed to Petri net systems either. We took the head revision of each process model and compared it to the previous revision. All activities (transitions, respectively) that have been added or deleted as part of model evolution were considered as separate changes. As such, we obtained a set of change transitions for each net system. If no change transition was detected, we compared the head revision of the process model with the second-previous revision. We proceeded iteratively until we either reached the first revision or detected at least one change transition. Using the identified changes, we also identified the set of transitions over which the changes have been consistent, see Section 4.2. Here, we relied on the jBPT library for the computation of behavioural profiles.

For each process model, we propagated the identified changes to all other process models. We constructed alignments using the technique used to isolate model clusters. That is, we compared the labels of all transitions using string edit distance similarity with a threshold of 0.8. Then, the identified alignment was reduced until we obtained a weak behavioural profile consistent alignment, see Section 4.4. We adopted the following heuristic to perform this reduction. For all transitions, we counted how often they participate in an elementary correspondence that together with another correspondence has different behavioural relations in the source and the target model for the respective transitions. In this way, we obtained a measure for the impact of any transition on the violation of weak behavioural profile consistency. Then, we iteratively removed correspondences that relate to the transition with the highest impact on the violation.

Having ensured that the alignment is weak behavioural profile consistent, we conducted the presented approach to propagate changes. We identified boundary nodes and inter-boundary nodes, and applied the reductions accordingly.

### Experimental Results

Using the 11 net systems derived from the BPMN process models, we considered 110 alignments between them.
as part of our experiment. For these alignments, descriptive statistics can be found in Table 1. The systems had between 18 and 33 transitions with an average of around 30. Exploiting past revisions of these models, we identified transitions indicating a change for all models. On average, we identified around 10 such change transitions.

As a next step, we investigated the impact of the consistency assumption for changes between model revisions. As illustrated in Table 1, the assumption led to the exclusion of transitions. On average, the considered revisions shared around 21 transitions, whereas three transitions had to be removed since the change influenced their behavioural relations with other transitions.

Using the change consistent transitions, we constructed the alignments as mentioned earlier and verified whether they are weak behavioural profile consistent. If not, we extracted a consistent sub-alignment. On average, we had to decrease the size of the alignment in terms of elementary correspondences by two, so that around four correspondences were left to be exploited for change propagation. Note that, even though we measure the size of the alignment based on elementary correspondences, all encountered alignments comprised at least one complex correspondence.

The 112 identified change transitions for 110 alignments yield a total of 1120 change propagation runs. In each run, we applied the presented approach to identify a change region. Table 2 illustrates the obtained results in absolute and relative terms. On average, the identified change regions comprised 6.58 transitions. The results have been diverse, though. In some cases, the region was not comprising any transition, i.e., the part in the model where the according change would have to be incorporated was clearly identified. In other cases, however, the change region comprised 25 transitions, which only minimally restricts the part that may be affected by change propagation.

Table 2 also lists the results in terms of the relative size of the change region. This measure is defined as the ratio of transitions in the change region and all transitions of the target system. As such, a value close to zero indicates a small change region, whereas a value close to one means that the change region nearly spans the whole target system. On average, the change region was smaller than a quarter of the target system. We further illustrate these results in Figure 7. According to Figure 7a, in more than half of the change propagation runs, we have been able to reduce the change region to 10% or less of the size of the target system. Only for a few runs, the change region was larger than 50% of the size of the target model. Apparently, support for change propagation is particularly required for models of large size. For that reason, Figure 7b also shows the relative size of the obtained change regions relative to the size of the target system. The darker the colour of a data point, the more frequently was it observed in the experiment. We see that cases, in which the change region is only minimally reduced, involve target systems of little size. In contrast, for most of the larger models, the relative size of the change region is rather small.

5.2. Models of a Health Insurer

The models used in our second experiment stem from a health insurance company. The process models capture business functions of this company assuming an organisational perspective. They have been created for staff planning. The collection comprises 1026 process diagrams in EPC notation, which are expanded to 1042 process models. We neglected models that contain syntax errors or behavioural errors, are trivial (i.e., contain only a single element), or cannot be normalised so that we arrive at a dedicated entry and exit point without duplicating events or functions. This led to a selection of 1014 models. For this selection, we identified pairs of semantically related process models as follows. For all model pairs, we
Table 3: Health Insurer: Descriptive statistics on the considered alignments.

<table>
<thead>
<tr>
<th></th>
<th>Avg (Std)</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td># Transitions in System</td>
<td>68.67 (62.40)</td>
<td>5</td>
<td>456</td>
</tr>
<tr>
<td># Elementary Correspondences in Alignment</td>
<td>11.78 (15.75)</td>
<td>4</td>
<td>134</td>
</tr>
<tr>
<td># Elementary Correspondences in Consistent Alignment</td>
<td>9.62 (14.49)</td>
<td>1</td>
<td>117</td>
</tr>
</tbody>
</table>

Table 4: Health Insurer: Descriptive statistics on the identified change regions.

<table>
<thead>
<tr>
<th></th>
<th>Avg (Std)</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td># Transitions in Change Region</td>
<td>18.45 (32.50)</td>
<td>0</td>
<td>363</td>
</tr>
<tr>
<td>Relative Size of Change Region</td>
<td>0.14 (0.17)</td>
<td>0</td>
<td>0.90</td>
</tr>
</tbody>
</table>

compared the normalised labels of the EPC functions and EPC events. Normalisation involved removal of leading or trailing whitespaces and converting all characters to lowercase. Two models have been considered to be related, if they share at least five normalised labels of nodes. This approach led to the identification of 1294 pairs of related process models built from 282 unique models. All these process models were transformed into Petri net systems following existing formalisations of EPCs [34].

Experimental Setup

For the models of the health insurer collection, change operations could not be deduced from past model revisions. As such, we had to rely on synthetic change operations, which are built as follows. For each model of a pair of related process models, we considered an activity (transition, respectively) that is without counterpart, i.e., for which there is no node with an equal normalised label in the other model, to be a separate change. We also identified the set of transitions over which the changes have been consistent as outlined earlier.

For the 1294 pairs of related process models, we constructed alignments and propagated all changes between the two models. Construction of alignments was grounded on the normalised labels of nodes. Again, the identified alignment was reduced until we obtained a weak behavioural profile consistent alignment. For this reduction, we relied on the heuristic introduced for the first experiment. Using the weak behavioural profile consistent alignment, all changes have been propagated using the proposed approach.

Also, the alignments are larger. They consist of nearly 12 elementary correspondences on average. Ensuring that the alignment is weak behavioural profile consistent led to a slight decrease of the size of the alignment. As for the previous experiment, on average, two elementary correspondences had to be removed to meet the consistency requirements. It is worth to mention that also in this collection, all encountered alignments comprised at least one complex correspondence.

Using the aforementioned approach, we identified 12777 unique change transitions, which led to a total of 88032 change propagation runs. A summary of the results of these runs is shown in Table 4. In absolute terms, the identified change regions comprised around 18 transitions on average. The minimal value of zero hints at change propagation runs, in which the change location could be precisely determined. However, the maximal value also indicates that in the worst case, the change region comprised more than 360 transitions. To compare these results with those obtained for the BPM AI collection, we have to take the larger size of the process models into account. Table 4 also lists the results using the ratio of transitions in the change region and all transitions of the target system. We observe that, on average, the change region included 14% of the transitions of the target system.

Figure 8 provides further insights into the obtained results. As shown in Figure 8a, in more than half of the change propagation runs, the change region included only 10% or less of transitions of the target system. This confirms the observations done in the first experiment. Further, only in a very few cases, the change region was larger than 50% of the size of the target model. Overall, the share of change propagations runs that led to changes regions larger than 50% of the size of the target model is even smaller than observed for the case of the BPM AI models.

Finally, we also investigated the relation between the relative size of change regions and the size of the target model. Figure 8b reveals that for many large target systems, e.g., those that consist of more than 150 transitions, we
obtained small change regions. In most cases, the regions cover less than a fifth of the model, whereas in many cases even a precise change localisation is possible. Figure 8a also indicates that in some change propagation runs, the change region is rather large, though. In particular, for some of the very large models the change region covers around two thirds of the target system, which, given the size of the models, provides limited help in practice. Still, Figure 8a already showed that those are exceptional cases.

5.3. Discussion

The results indicate that the presented approach indeed supports the propagation of changes between aligned process models. Instead of investigating a complete model, a process analyst may focus on the identified change region, which is significantly smaller than the model. In many cases, we have been able to determine the location for change propagation with a high precision of a few nodes. This is remarkable since in particular in the first experiment, the size of alignments considered was relatively small.

To further illustrate the benefits and limitations of our approach, Figure 9 depicts two excerpts of process models from the BPM Academic Initiative. Both have been used in our experiment. Clearly, the labels of activities are rather homogeneous, so that various correspondences may be identified between the model excerpts. Consider the two activities highlighted in Figure 9a, which, as part of our experiments, shall be propagated to the model in Figure 9b. For the change related to activity ‘assign penalty points’, only boundary transitions may be exploited, i.e., activities ‘create one process model’ and ‘Extend their process to level 2 - 3’. As such, the change region in Figure 9b is rather large and comprises two activities (‘Check process model’ and ‘Fix until next session’), an XOR-gateway, and two events (‘penalzation’ and ‘4 days’). The lack of precision here stems from the control flow cycle. The abstraction of a behavioural profile does not allow for a precise characterisation of the activity ordering within this cycle. In contrast, our approach enables precise localisation of the change region for the second change, i.e., activity ‘assign roles in teams’. In this case, the change region is empty, whereas the preceding and succeeding boundary transitions along with the interleaving inter-boundary transitions hint at an exact spot in Figure 9b: the flow arc from activity ‘Analyse field and Identify processes’ to the AND-gateway.

We conclude that in many cases, our approach allows for the identification of a narrow change region. For certain settings the restriction of the model part that may be subject to change is modest, though. In these cases, few transitions do not qualify to be part of the change region. We illustrated that a certain imprecision may stem from the application of a behavioural abstraction.

6. Related Work

Our work relates to different streams of research. We already elaborated on matching techniques for process models that may be utilised to identify correspondences in Section 2.1. In this section, we focus on work on view-based process modelling, change management, and approaches to cope with process variability.

6.1. View-based Process Modelling

The existence of process models that overlap in content stems from the need to have process models that have been tailored for a certain purpose. Since those models evolve independent of each other, the relation between them needs to be managed explicitly. Alignments need to be constructed, consistency needs to be verified, and changes need to be propagated.

A different approach to cope with the variety of applications of process models is taken by view-based approaches.
View-based process modelling has its roots in Software Engineering techniques to design a system using viewpoints, see [35, 36, 37, 38, 39, 40]. Viewpoints capture different aspects of a system and, therefore, may serve different purposes. Since all viewpoints relate to a single point of truth — the system to be built — consistency between viewpoints is of central importance. To this end, behavioural equivalences may be leveraged [36, 37, 40] or a formula encoding behavioural dependencies of all viewpoints is checked for satisfiability [35, 39].

In the same vein, approaches to view-based process modelling assume a holistic model from which views are derived for different purposes [41, 42, 43, 44]. Typically, views are derived by a set of derivation rules that preserve a certain notion of consistency, e.g., the ‘activity orderings’ [41]. Note that the consistency obtained between the holistic model and the view is often grounded on behaviour equivalences and, therefore, stronger than weak behavioural profile consistency as required by the presented approach.

In summary, view-based modelling avoids the need to manage the relation between different models explicitly. The close coupling between a model and views on this model, however, also limits the flexibility when tailoring a model for a certain purpose. Views that are derived by transformation rules require that the relation between corresponding model elements is traced back to hierarchical refinements. Those transformations result in non-overlapping correspondences. Further, the groups of associated model elements, e.g., activities of process models, are also structurally restricted, e.g., activities have to form a connected single-entry single-exit subgraph. As such, view-based modelling avoids the challenge of propagating changes in a consistent manner. On the downside, view-based modelling is not able to cope with non-hierarchical relations between models, which are frequently observed ‘in the transition from high-level models of the application domain to the implementation model’ [45].

6.2. Process Model Change Management

We mentioned before that change management for process models comprises three steps, detection of differences, analysis of their relations, and the actual resolution of differences [2].

The identification of differences is closely related to similarity measures for process models, see [13, 14, 15, 16, 17]. Once differences are detected, they may be traced back to change operations [16, 47, 48, 49, 50]. Here, a change operation is typically not meant to be an atomic graph operation (e.g., insert edge), but refers to a compound change operation that is often observed in practice. Replacing an activity with a subprocess or adding the handling of an exception case would be examples for such operations. For these compound changes, dependencies, e.g., two changes are conflicting, may be investigated [49]. Then, the identified compound changes guide the adaptation of process models to propagate changes.

Change management along these lines is conceptually different compared to the presented approach. Our approach does not rely on a set of change operations. Consequently, we do not have to make any assumption on the structure of the correspondences that are leveraged for change propagation. As such, our approach is more generally applicable than approaches that are limited by a set of change operations. Apparently, the general applicability of our approach comes with a loss in precision for the resolution of differences. The presented approach is limited to localising a change region, the concrete implementation of the change is left to a process analyst, though.

6.3. Process Variability

Change management for process models is closely related to techniques for managing process variability. Models of process variants do not capture the very same business process, but very similar processes that slightly deviate from each other. Such deviations may be caused by different legal requirements in different countries, or differences between organisational units with respect to their organisational structure and supporting IT infrastructure [50]. Process variability may be addressed in the course of process automation [51, 52, 53, 54, 55]. However, we focus on variability on the model level in the following paragraphs.

Incorporating variability in the design of process models is inspired by work on Software Product Line Engineering (SPLE) [50, 57, 58]. Following this line, configurable reference models have been proposed. Based on such a reference model, models for process variants may be derived by model projection [50, 60]. Also, configuration mechanisms that go beyond model projection have been proposed for various process modelling languages, e.g., for EPCs [12, 51, 60].

Figure 9: Excerpts of two process models from the BPM Academic Initiative used in the experiment (graph layout has been changed, labels as in original).
WF-systems [63] and process models within the Provop framework [64, 65].

Configuration of a reference model may be based on blocking or hiding certain activities [12, 66, 62, 61], or restrictions of control flow routing elements [12, 61]. However, configuration is not necessarily limited to restricting behaviour. The Provop [67] framework advocates the adaptation of reference models by a collection of change patterns, the later are discussed in [29]. Also, stereotypes that mark variation points have been proposed to be incorporated into process models [61, 65].

As for work on view-based process modelling, approaches to managing process variability take a conceptually different approach compared to our work. Process models are closely coupled as they are derived from a single reference model by well-defined operations. Hence, changes may be applied to the reference model, which reduces the tendency of the models to drift apart. Again, however, the close coupling restricts the flexibility to tailor a model for a certain purpose. Also, all process stakeholders are required to agree on the maintenance of a configurable reference model. Once such a scenario is not feasible, our approach helps to maintain process models that are related but evolve independently of each other.

7. Conclusions

In this article, we have addressed the problem of propagating changes between process models, even if they are not in a strict hierarchical refinement relation or if they show behavioural inconsistencies. Our contribution is a general change propagation approach that utilises behavioural profiles of corresponding activities in order to determine a change region in another model. Our evaluation revealed that the approach is applicable for change operations observed in practice and also scales for large process models. We conclude that, apart from exceptional cases, the size of the region where a change has to be applied is only a fraction of the overall model.

In future research, we aim to work on some limitations of our approach. We currently utilise behavioural profiles as an abstraction. Earlier, we discussed that these profiles do not capture specific constraints between activities in a control flow cycle. Incorporating additional behavioural relations may enable us to further narrow down change regions.

Beyond that, the interplay of the presented technique and matching techniques for constructing alignments, as mentioned in Section 2.1, has to be further investigated. The construction of an alignment is of crucial importance for any approach to change propagation. In our experiments, we could rely on rather simplistic means for matching since both model collection showed a relatively homogeneous vocabulary. In the general case, more advanced matching techniques are needed. On the one hand, those may increase the number of correspondences and, therefore, the amount of information that may be exploited to reduce the change region. On the downside, however, such matching techniques may also increase the level of inconsistency observed for an alignment, e.g., by falsely identifying correspondences. Hence, further experiments are needed to investigate the robustness of our technique in such a setting.

Also, matching techniques for process models are rarely available for professional tools. Once such matching techniques have been incorporated, future work shall also explore how strongly a user benefits from our technique when applying changes. Testing of user acceptance was beyond the scope of this article.

References


