Summer 6-15-2016

AN EXECUTION-SEMANTIC APPROACH TO INDUCTIVE REFERENCE MODEL DEVELOPMENT

Jana-Rebecca Rehse
Institute for Information Systems (IWi) at the DFKI, jana-rebecca.rehse@iwi.dfki.de

Peter Fettke
Institute for Information Systems (IWi) at the DFKI, peter.fettke@iwi.dfki.de

Peter Loos
Institute for Information Systems (IWi) at the DFKI, loos@iwi.uni-sb.de

AN EXECUTION-SEMANTIC APPROACH TO INDUCTIVE REFERENCE MODEL DEVELOPMENT

Research

Jana-Rebecca Rehse, Institute for Information Systems (IWi) at the DFKI, Saarbrücken, Germany, Jana-Rebecca.Rehse@iwi.dfki.de

Peter Fettke, Institute for Information Systems (IWi) at the DFKI, Saarbrücken, Germany, Peter.Fettke@iwi.dfki.de

Peter Loos, Institute for Information Systems (IWi) at the DFKI, Saarbrücken, Germany, Peter.Loos@iwi.dfki.de

Abstract

Reference models are a cost- and time-saving approach for the development of new models. As inductive strategies are capable of automatically deriving a potential reference process model from a collection of existing process models, they have gained attention in current research. A number of promising approaches can be found in recent publications. However, all existing methods rely on graph-based similarity measures to identify commonalities between input models. Since behaviourally similar process models can have different graphical structures, those approaches are unable to find certain commonalities. To overcome these shortcomings, we propose a new approach to inductive reference model development based on an execution-semantic similarity measure. Since a naïve solution to the intuitive idea does not yield productive results, the proposed approach is rather elaborate. By capturing the commonalities of the input models in a behavioural profile, we are able to derive a reference model subsuming the input models' semantics instead of their structure. In our contribution, this approach is outlined, implemented and evaluated in three different scenarios. As the evaluations show, it is capable of handling complex process models and overcome most restrictions that structural approaches pose. Thus, it introduces a new level of flexibility and applicability to inductive reference modelling.

Keywords: Reference Modelling, Inductive Reference Model Development, Execution Semantics, Behavioural Profiles

1 Introduction

The utilization of reference process models is a cost- and time-saving approach to the design of new models. They serve as blueprints for best-practice processes used in the respective industry and can be adapted to fulfil the individual needs of an organization (Becker and Meise, 2011). The use of reference models is associated with a higher quality of the business process and the related process model, as it simplifies internal communications by introducing a common terminology and considerably reduces the resources required for business process management (Fettke and Loos, 2007, p. 5). However, in order to exploit these benefits, organizations require access to existing high-quality reference models.

Reference models may be constructed both deductively and inductively (Becker and Schütte, 1997). Deductive methods, which are also known as "top-down" approaches, employ generally accepted theories and principles. Models are constructed on a theoretical base and gradually substantiated along the way. In contrast, inductive or "bottom-up" development of reference models makes use of real-world data such as existing process models or execution logs. It focuses on similarities and commonalities within the input data and abstracts from individual features of the single models. As inductive strategies are capable of automatically deriving a potential reference model from a collection of existing process models, they are gaining attention in current research activities.

A number of promising inductive approaches to reference model development can be found in recent literature (e.g. Ardalani et al., 2013; Li et al., 2009; Martens et al., 2014; Rehse et al., 2015; Yahya, Bae, et al., 2012; Yahya and Bae, 2011). All of them have certain assets and drawbacks and the respective evaluations show promising results. However, all of them rely on a structural notion of process equality in order to identify similarities between the input models. This implies that model similarities are only discovered if the models obtain a similar structure, although most semi-formal modelling languages allow for a great degree of structural variations. In addition, a lot of emphasis is placed on given node relations.

This contribution is set out to pursue a different approach to inductive reference model development. In contrast to the existing approaches, it does not rely on the underlying graph structure of the provided input models. Instead, it employs execution-semantic aspects, constructing a consolidated model out of semantic relations between nodes. Understanding "semantic" as related to process execution (opposed to structure), the resulting reference model subsumes the behaviour of the input models rather than their design. Consequently, this method should offer greater flexibility and pose fewer restrictions on the input data.

Since the main objective of our research is the design of a new approach to inductive reference model development, we follow a design-oriented research approach (Hevner et al., 2004; Peffers et al., 2007). Driven by the research objective to devise a more flexible, more capable and less restrictive approach to inductive reference modelling, we deduce to follow a different paradigm for the identification of process model commonalities. Following this paradigm, we design a new approach to inductive reference model development as a novel artefact in the area of information systems. It is implemented as a proof-of-concept and extensively evaluated in several scenarios.

Following this design-oriented approach, this contribution is structured as follows. In section 2, we present a motivating example, which demonstrates the deficiencies of structure-based inductive reference modelling in general and illustrates the restrictions and shortcomings of a number of existing approaches. Section 3 introduces our newly developed approach to inductive reference modelling, named RMM-2. It is implemented as a proof-of-concept and evaluated in multiple scenarios, which are presented in section 4. The evaluation results as well as the potentials and limitations of our approach are critically assessed in section 5. We discuss related work in section 6. Section 7 concludes the paper with a summary and an outlook on future work.

2 Motivational Example

Figure 1 shows an example of four small models, each depicting a simple trip booking process by means of an Event-driven Process Chain (EPC). Although the models are quite similar, none of the existing approaches to inductive reference modelling is capable of deriving a meaningful reference model for this input, since the approaches are all based on a structural notion of process similarity.

Behaviourally similar process models can exhibit very different model structures. As an example, a typical result of a graph-based approach is shown in figure 1. While the two subgraphs that form the start and end of all process models can easily be found, the middle block, containing different nodes in different relations and a loop, cannot be reconstructed. In particular, conflicting order relations, loops, and differing scopes prevent or falsify the computation of a valid reference model.

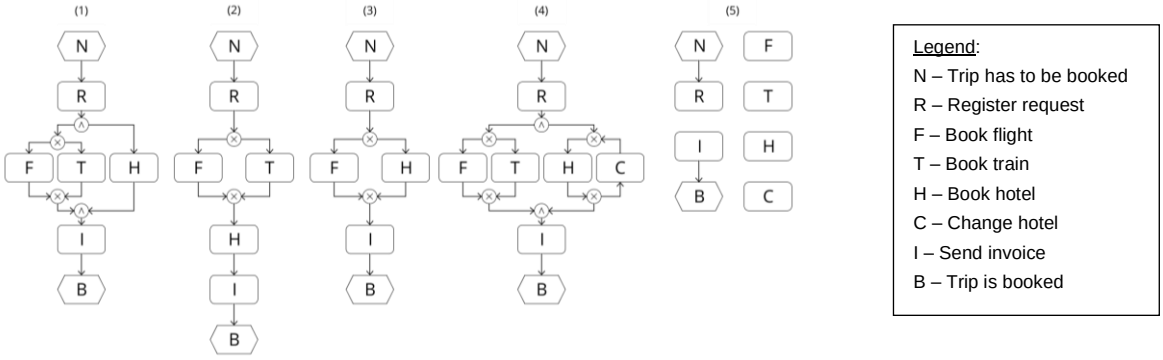

Figure 1. Four example models [1](1) – (4) and a typical result of a graph-based approach (5)

Table 1 summarizes some important requirements to inductive reference modelling by analysing the shortcomings of six existing approaches. Identified by means of a literature review, they are compared to the proposed approach by indicating whether they fulfil (✓) or not fulfil (×) a certain requirement. In addition to the listed restrictions and shortcomings, all of the approaches assume a uniform level of granularity and the absence of duplicate labels. The listed shortcomings impede the application graph-based approaches in many practical use cases, as the problematic features and properties are contained in most real-world models.

Approach	Models may contain loops	Models may be arbitrarily structured	Models may contain errors	Models may have multiple startnodes	Approach supports external mapping	Reference model is connected	Frequent relations are re-tained
Ardalani et al. (2013)	×	✔	×	✔	✔	×	✔
Li et al. (2009)	×	×	✔	✔	×	✔	✔
Martens et al. (2014)	✔	✔	×	✔	✔	✔	×
Rehse et al. (2015)	×	✔	✔	✔	✔	×	✔
Yahya & Bae (2011)	×	✔	✔	×	×	✔	✔
Yahya et al. (2012)	×	✔	✔	×	×	✔	×
RMM-2	✔	✔	✔	✔	✔	✔	✔

Table 1. Requirements of approaches to inductive reference model development

[1] Originally depicted as Petri Nets (Van der Aalst et al., 2006), the models are converted to EPCs by inserting a function for a Petri Net transition. An AND-split (join) is inserted for a transition with multiple inputs (outputs) and an XOR-split (join) is inserted for a place with multiple inputs (outputs). Intermediate events are omitted for the sake of comprehensibility.

3 The RMM-2 Approach

3.1 Limitations of a Naïve Solution

From the apparent shortcomings of structural approaches to inductive reference model development we deduce an execution-semantic principle for our approach. This means that we derive a reference model based on the execution traces of the individual models. By constructing a completely new model from these execution traces, we make sure that the behaviour of the individual models is represented in the reference model. Structurally equivalent models will always generate the same execution traces, however, the same holds for structurally different, but behaviourally equivalent models. Similarly, execution semantics are able to handle loops, since every execution trace is finite.

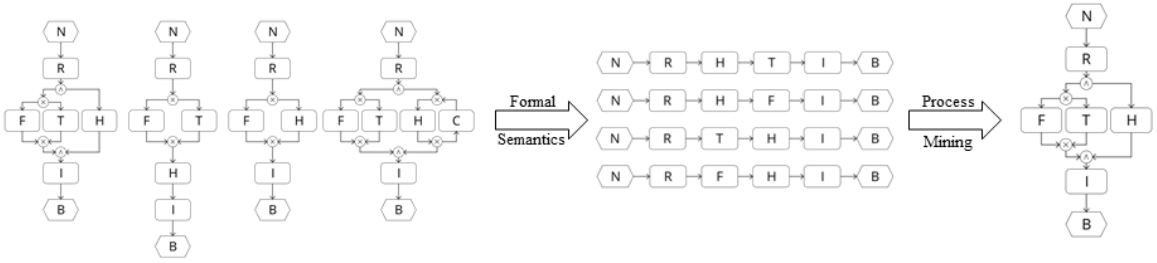

Figure 2. Naïve solution to the intuitive idea

Figure 2 outlines a naïve execution-semantic approach to inductive reference model development. Execution traces are generated from the input models, based on formal process model semantics. These traces are used to derive a reference model by means of Process Mining techniques. Applying parameters as well as pre- and post-processing, the model set behaviour, as represented by the traces, is contained the resulting reference model. This idea explains the concept of inductively developing a reference model based on execution-semantic similarities. However, due to the high computational complexity of computing and storing execution traces, it is not feasible in practice. To illustrate this, we consider one single operator block with *n* branches. If this block is connected by an XOR-connector, it yields *n* different execution traces. For an AND-connector, every single branch has to be executed. Since the branches are not causally dependent, all possible execution orders are allowed, yielding $n!$ possible execution orders. Every OR-connector allows the execution of any subset of subsequent branches (except the empty set). Hence, for every OR-connector with n branches, there are $2^n - 1$ executable subsets of branches. Within each of these subsets, every possible order is allowed. This yields a total $\sum_{i=1}^{n}\binom{n}{i} * i!$ different execution traces. To clarify the significance of these formulas for our computational task, Table 2 lists the number of possible execution traces for an operator block with *n* branches, with *n* ranging from 1 to 10. It is evident that computation of all possible traces will be impossible in a reasonable computation time, especially since most models contain more than one operator block.

n	1	2	3	4	5	6	7	8	9	10
AND	1	2	6	24	120	720	5,040	40,320	362,880	3,628,800
OR	1	4	15	64	325	1,956	13,699	109,600	986,409	9,864,100

Table 2. Numbers of possible execution traces for AND- and OR-connectors

Hence, the RMM-2 approach is able to deduce the behavioural similarities of the input models without explicitly computing and storing execution traces. Instead, the behaviour of a model set is represented by so-called behavioural profiles (Weidlich et al., 2010, 2011), matrices containing semantic relations for every pair of contained nodes. These behavioural profiles can be derived directly from the process model itself and serve as the basis for computing a reference model.

3.2 Outline and Specification of the Proposed Approach

Based on existing approaches by (Ardalani et al., 2013; Li et al., 2009; Martens et al., 2015; Rehse et al., 2015), we derive a procedure model for the RMM-2 approach, which is shown in Figure 3. The three stages marked grey are particularly relevant to our approach and are thus described in detail in the following sections. This does not mean that the other stages are irrelevant; our approach would not be able to compute a meaningful reference model without collecting the input models and computing an appropriate matching between them. However, these stages are important to almost all inductive approaches to reference model development. The relevance of collecting a meaningful set of input models and computing a mapping between them as well as the different measures of postprocessing a reference model are extensively covered by Rehse et al. (2015).

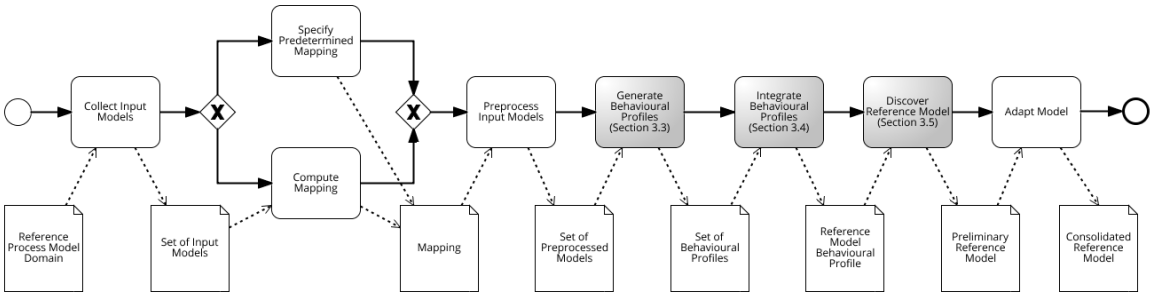

Figure 3. Outline of the Refined RMM-2 Approach

Before a user is able to develop a reference model, the input data has to be gathered. The boundaries of the reference model domain are determined by choosing a set of appropriate process models. While some organizations may decide to split their domain into several shorter processes, others may decide to include all activities into a single model. The combined scopes of all input models define the process domain, from which the reference model scope is derived.

The input models provide the computational basis of every inductive development. Users have to adhere to the restrictions an approach imposes on its input models. Unlike many other approaches, our approach allows models to include loops; however, the models may not include duplicate nodes or exhibit differing degrees of abstraction. As a second input, we require a mapping of the input models. Such a mapping of semantically equivalent functions can either be constructed manually or computed by a state-of-the-art process matching algorithm (Antunes et al., 2015). Based on the identified analogies, the approach is able to derive semantic similarities in the set of input models.

The scope of the new process model is determined within the boundaries of the process domain, which is given by the chosen input models. It defines which activities are to be included in the final model. Our approach provides two parameters, which impact size and structure of the resulting reference model. Depending on the chosen threshold values, models can be smaller or larger, "Spaghetti-like" or hardly connected at all. However, these characteristics also depend on the variability of the process domain, so their values should be chosen for each application individually.

- Frequency Threshold (t): The frequency of a node is determined by dividing the transitive match frequency of the node by the total number of models. The scope parameter $t \in [0; 1]$ is then defined as a frequency threshold, setting the minimum frequency in order for a node to appear in the resulting reference model.
- Noise Level (n): The noise level determines how often a relation has to appear in the set of input models to be adopted into the reference model. It is defined as a threshold percentage $n \in [0; 1]$, specifying the minimum percentage of models a relation has to appear in. This ensures that infrequent relations are not included in the reference model, such that its behaviour truly represents the common behaviour of the set of input models.

Applying the frequency threshold *t*, the scope of the reference model is derived from the scope of the input models. If a node's relative frequency is at least as high as *t*, the node is part of the reference model. If the frequency is lower than *t*, the node has to be removed from the input models. For an internal node, we add an additional arc from its predecessor to its successor node before removing it. If the removal results in a connector block with only one branch or an empty loop, the respective edges and connectors have to be removed as well.

3.3 Generate Behavioural Profiles

After identifying the scope of the reference model, its behaviour has to be determined in terms of process model semantics. One way to represent the pairwise semantic relations within a set of nodes are so-called behavioural profiles (Weidlich et al., 2010, 2011). A behavioural profile for a set of nodes S with $|S| = n$ is a n×n matrix, which contains a semantic order relation for every pair of nodes $a,b \in S$. In Process Mining, behavioural profiles are usually derived from a set of process traces. As we explain in section 3.1, it is not computationally feasible to compute all possible execution traces in order to compute a behavioural profile for every one of the input models. Instead, we compute the behavioural profiles directly from the process models themselves by inspecting their formal execution semantics.

All following definitions define an EPC as a tuple P = (E, F, C, A, τ), according to the definition by Weske (2012, p. 162), where E is the set of events, F the set of functions, V = E ∪ F the set of vertices, C the set of connectors, A the set of arcs and τ the connector typing function.

Definition (Model-based Ordering Relations): Let P = (E, F, C, A, τ) be an event-driven process chain. For every pair of vertices $v_1, v_2 \in V$, one of the following relations holds.

- $v_1 \#_P v_2$ if there exist two process execution where one connector enables v_1 and disables v_2 in one execution and vice versa in the other (XOR-split) or there exist two process executions where one connector in solely enabled by v_1 in one execution and by v_2 in the other (XOR-join)
- $v_1 \|_P v_2$ if for every process execution, v_1 and v_2 are enabled by the same connector (AND-split) or jointly enable the same connector (AND-join)
- $v_1 * v_1$ if there exists a process execution where v_1 is enabled by itself (one-node loop)
- $v_1 \bullet v_2$ if there exists a process execution where v_1 is enabled by v_2 and vice versa (two-node loop)
- $v_1 \rightarrow_P v_2$ if for every process execution, v_1 is enabled by v_2 (possibly via connectors)
- $v_1 -_P v_2$ otherwise

The ways in which nodes are enabled and disabled by each other depend on the formal semantics that is used. Because the EPC is a semi-formal modelling language, there exist a few different approaches to formalizing its semantics. The contribution at hand refers to the semantics defined by Mendling (2007) and Mendling and Van der Aalst (2006)[2].

For every input model of the approach, a behavioural profile is computed. Model-based ordering relations are identified by traversing the model, imitating an execution by means of the formal EPC semantics. Whenever a node is fired, we check which of the above relations hold and denote it in the behavioural profile. Such a traversal has a complexity linear to the model size. After executing this step, we obtain a set of behavioural profiles, which represent scope and behaviour of every input model. However, to derive a reference model, we need one single behavioural profile subsuming those of the input models. Hence, in the next step, the set of profiles is parametrized and integrated accordingly.

[2] Formal definitions of the above relations based on the given semantics are available upon request.

3.4 Integrate Behavioural Profiles

In this stage, we determine both the set of nodes the reference model will consist of and their relation towards each other by integrating the individual behavioural profiles. The integrated profile will serve as a basis for discovering the actual reference model in the next step.

Since the models were already preprocessed and infrequent nodes were removed, all nodes that are contained in the input models at this point will be represented in the reference model. For every (transitive) match between the input models, one representative node is chosen. This set of representatives will constitute the reference model.

For every pair of representatives, we have to determine their ordering relation, according to the subset of input models they are contained in. Therefore, we apply the following definition.

Definition (Integrated Behavioural Profile): Let $P = P_1,\ldots,P_m$ be the set of input models, $V_k = E_k \cup F_k$ the set of nodes for model P_k, $V \subset \bigcup_{k=1}^{m} V_k$ the set of nodes that constitute the reference model and $v_i, v_j \in V$. The integrated behavioural profile is a matrix $M_{|N|\times|N|}$, such that for every M_{ij} one of the following relations holds.

- $v_i * v_i$ if there exists P_k with $v_i \in P_k$ and $v_i *_{P_k} v_i$
- $v_i \bullet v_j$ if there exists P_k with $v_i, v_j \in P_k$ and $v_i \bullet_{P_k} v_j$
- $v_i \,||\, v_j$ if either (there exists P_k with $v_i, v_j \in P_k$: $v_i \,||_{P_k} v_j$) or

  (there exist P_k, P_m with $v_i, v_j \in P_k$ and $v_i, v_j \in P_m$: $v_i \rightarrow_{P_k} v_j \wedge v_i \leftarrow_{P_m} v_j$)
- $v_i \rightarrow v_j$ if for all P_k with $v_i, v_j \in P_k$: $v_i -_{P_k} v_j \vee v_i \rightarrow_{P_k} v_j \vee v_i \#_{P_k} v_j$
- $v_i \# v_j$ if for all P_k with $v_i, v_j \in P_k$: $v_i -_{P_k} v_j \vee v_i \#_{P_k} v_j$
- $v_i - v_j$ otherwise

The integrated behavioural profile is parametrized by the Noise Level parameter *n*. It specifies the minimum number of models in which a relation has to appear in order to be included in the behavioural profile. This cancels out infrequent behaviour and allows for clearer relations and thus higher chances for a connected, meaningful model.

3.5 Discover the Reference Model

The integrated behavioural profile and its corresponding set of nodes serve as the basis for determining the final preliminary reference model. To perform this task, we define a basic algorithm for process discovery, i.e. the construction of a new process model, represented as an EPC, encompassing the behaviour as specified in the behavioural profile. Similar ideas are used in some algorithms for Process Mining (Van der Aalst, 2011).

This algorithm has to account for the syntactic and semantic specificities of EPCs. EPCs make their control relation explicit by the use of split- and join-connectors, associated with a type and respective semantics. Hence, explicit AND-, OR- and XOR-semantics have to be considered separately. EPCs allow nested operators, which add another level of complexity to the algorithm. Also, loops that consist of only one or two loops have to be identified separately.

The following algorithm takes all of those features into account and computes an EPC, which allows for the semantics expressed in the behavioural profile. Given an integrated behavioural profile M and its corresponding set of nodes V, we can define the following sets.

- $E = \{e \mid e \in V \text{ and } e \text{ is an } event\}$ is the set of events
- $F = \{f \mid f \in V \text{ and } f \text{ is a } function\}$ is the set of functions

- $X_{\wedge}^{S} = \{(a,B) |\ a \in V \wedge B \subset V \wedge B \neq \varnothing \wedge \forall_{b \in B} a \rightarrow b \wedge \forall_{b_1,b_2 \in B} b_1 || b_2\}$
- $Y_{\wedge}^{S} = \left\{(a,B) \in X_{\wedge}^{S} \mid \forall_{(a,B') \in X_{\wedge}^{S}} B \subseteq B' \Rightarrow B = B'\right\}$ (AND-splits)
- $X_{\times}^{S} = \{(a,B) |\ a \in V \wedge B \subset V \wedge B \neq \varnothing \wedge \forall_{b \in B} a \rightarrow b \wedge \forall_{b_1,b_2 \in B} b_1 \# b_2\}$
- $Y_{\times}^{S} = \left\{(a,B) \in X_{\times}^{S} \mid \forall_{(a,B') \in X_{\times}^{S}} B \subseteq B' \Rightarrow B = B'\right\}$ (XOR-splits)
- $X_{\wedge}^{J} = \{(A,b) |\ A \subset V \wedge b \in V \wedge A \neq \varnothing \wedge \forall_{a \in A} a \rightarrow b \wedge \forall_{a_1,a \in B} a_1 || a_2\}$
- $Y_{\wedge}^{J} = \left\{(A,b) \in X_{\wedge}^{J} \mid \forall_{(A',b) \in X_{\wedge}^{J}} A \subseteq A \Rightarrow A = A'\right\}$ (AND-joins)
- $X_{\times}^{J} = \{(A,b) |\ A \subset V \wedge b \in V \wedge A \neq \varnothing \wedge \forall_{a \in A} a \rightarrow b \wedge \forall_{a_1,a \in B} a_1 \# a_2\}$
- $Y_{\times}^{J} = \left\{(A,b) \in X_{\wedge}^{J} \mid \forall_{(A',b) \in X_{\times}^{J}} A \subseteq A \Rightarrow A = A'\right\}$ (XOR-joins)
- $X_{1L} = \{(P,a,S) |\ a \in V \wedge P,S \subset V \wedge a * a \wedge \forall_{v \in V} v \rightarrow a \Rightarrow v \in P \wedge \forall_{v \in V} a \rightarrow v \Rightarrow v \in S\}$
- $X_{2L} = \{(P,a,b,S) |\ a,b \in V \wedge P,S \subset V \wedge a \bullet b \wedge \forall_{v \in V} v \rightarrow a \Rightarrow v \in P \wedge \forall_{v \in V} a \rightarrow v \Rightarrow v \in S\}$

All of those sets are necessary such that we can compute an EPC that allows for the specified behaviour. The first two sets of events and functions are derived directly from the corresponding set of nodes and constitute the nodes of the reference model. The four sets $X_{\wedge}^{S}, X_{\times}^{S}, X_{\wedge}^{J}, X_{\times}^{J}$ are necessary to define the control flow of the resulting EPC. Each one represents one possible connector type, i.e. AND-splits, XOR-splits, AND-joins, and XOR-joins. The sets contain pairs of a node and a set of nodes, which constitute the input and output nodes of a connector in the final EPC. For each set *X*, a second corresponding set *Y* is specified, which contains only the maximum pairs, such that only one connector is inserted for the entire set of successors or predecessors instead of one for every subset.

Loops of one or two nodes have to be treated separately to be correctly discovered. Therefore, we define the two sets X_{1L}, X_{2L}. X_{1L} contains all nodes that are contained in an individual loop. Similarly, X_{2L} contains pairs of nodes that are jointly contained in a loop. To correctly define a loop, its sets of predecessors P and successors S have to be determined as well.

For a meaningful EPC, we also have to define a set of connectors *C* as well as a control flow relation *A*. A connector of the respective type is defined for every pair contained in the sets $Y_{\wedge,\times}^{S,J}$. For each loop, two XOR-connectors are defined one as entry and one as exit point. The three sets *E*, *F* and *C* are connected by the set of edges *A*. To implement the control flow relations specified in $Y_{\wedge,\times}^{S,J}$, we define four arc sets $A_{\wedge,\times}^{S,J}$. In case of a split-connector, one edge is inserted from the common predecessor node to the corresponding connector and one from the connector to every successor node. The arcs are inversely defined for the join nodes of both types. The arc sets A_{1L}, A_{2L} compose the loop structure. All predecessor nodes are connected to the entry point connector. In both cases, this connector has node *a* as an output. In case of a two-loop, node *b* is inserted after node *a* before connecting it to the exit point connector. The loop itself is defined by an arc connecting the exit point to the entry point. Finally, arcs are inserted from the exit point connector to all successor nodes. For sequentially ordered nodes, the according edges are inserted by the set A_S, which defines a simple edge between every pair of nodes that have exactly one predecessor respectively successor.

Finally, connectors are nested in order to obtain a correct control flow. A nested connector is a connector, which has another connector as an output or input. Nested split operators can be identified by examining their successor sets. If one successor set is a real subset of another successor set, the connectors have to be nested, since the outer one has the same successors as the inner one, but also additional ones. The nesting is achieved by altering the set of already defined arcs. For each pair of nested connectors, we add an edge from the outer to the inner one. Then, we remove all edges between the outer connector and the successors of the inner connector, because this connection is now defined

indirectly by the connection to the inner connector. Analogously, the procedure is inversed for join connectors.

In the final stage of the development process, the discovered model can be manually changed to fit the needs of its stakeholders. This may include expanding, aggregating or replacing, adding or removing nodes, adding deductively developed model parts, correcting modelling mistakes, as well as removing, changing or reordering model parts.

4 Proof-of-Concept and Evaluation

In order to demonstrate and evaluate the capabilities of RMM-2, the approach, as specified above, is prototypically implemented and integrated into a research prototype developed in our research group. This Java-based prototype provides functionalities for loading and saving XML-based representations of business process models, such as EPML (EPC Markup Language) or AML (ARIS Markup Language) files. In addition, it contains several state-of-the-art matching algorithms that can be used to compute a high-quality matching between the input models. This proof-of-concept implementation is then used to evaluate the approach in three different scenarios, which are all derived from real-world datasets. The scenarios are summarized in table 3.

	Domain	#Models	Avg. Size	Modelling type
Scenario 1	Third-party Funding	4	29.0	Manually Designed Models
Scenario 2	Residents' Registration	10	22.0	Synthetically Generated Data Set
Scenario 3	Frequent Flyer Programs	8	28.5	Controlled Manual Modelling Exercise

Table 3. Summary of evaluation scenarios

The first scenario originates from a BPM project regarding the processes around third-party funded research projects at German universities (Gröger and Schumann, 2014). Processes from given domains were collected at four different universities and manually modelled as BPMN process diagrams, containing harmonized labels. For our evaluation scenario, they were transformed into EPCs and translated from the original German into English. The original publication suggests a reference model for the given set of process models, which is used as the benchmark for our evaluation. Figure 4 shows the input models for scenario 1 as well as the suggested and computed reference models.

For the second scenario, we choose a model from the CoSeLoG project, which was set out to document and compare business processes from several Dutch municipalities, as the basis for a synthetic modelling case. By randomly deleting, moving, and renaming nodes, we compute a set of ten process models, each depicting a variation of the given model (Ardalani et al., 2014). Our approach is then applied to these variations, whereas the original model serves as the reference model benchmark.

In the third evaluation scenario, we use data from a controlled modelling scenario. A textual representation of a business process is given to a number of candidates, which then model the described process as an EPC. From the resulting process models, we exclude those that do not adhere to the restrictions of the approach, i.e. those that are not fully connected, do not contain at least one start and end node, and contain events or functions with more than one ingoing and outgoing edge. From the remaining model set, eight models are chosen randomly. The sample solution to the controlled modelling exercise is used as the expected reference model.

Every scenario consists of a number of models that describe different business processes from the given domain. For all three sets, we compute a reference model using the implementation described above. Since there exists a given reference model for each scenario, we use this model as a benchmark for our evaluation. The reference model that is computed by the RMM-2 approach is compared to the given reference model to determine the capability, flexibility, and the restrictions of the approach.

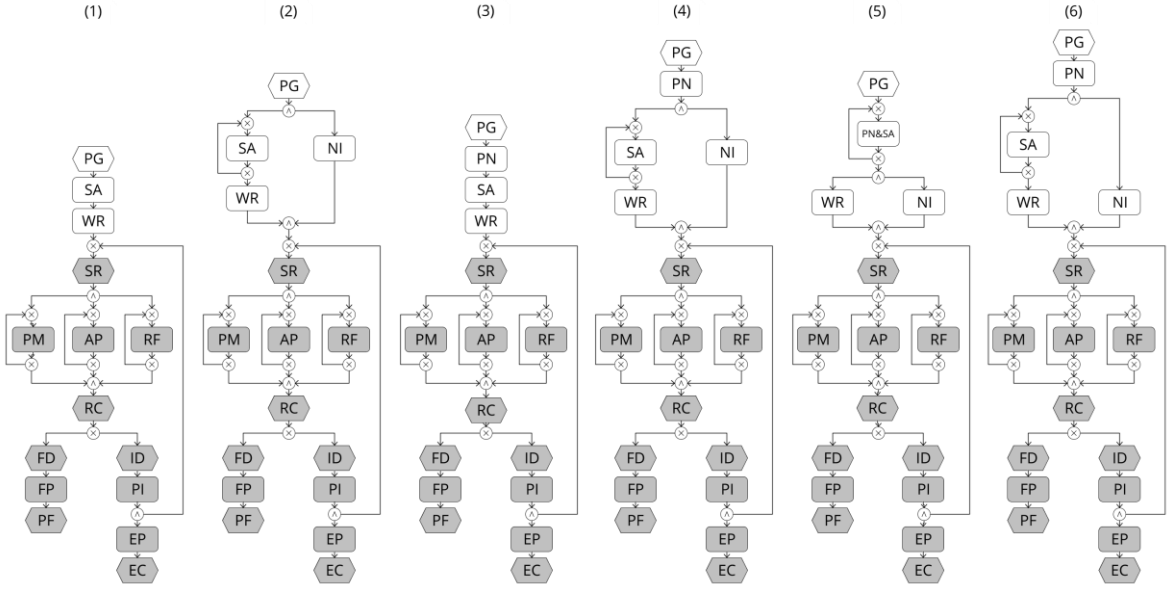

Figure 4. Four process models (1) – (4) depicting the execution stage of a third-party funded research project at German universities (adapted from (Gröger and Schumann, 2014)), the suggested reference model (5) and the computed reference model (6)

This comparison makes use of several numeric measures that originate in the field of Information Retrieval. Precision, Recall and F-measure are for example used to determine the quality of a computed process matching in comparison to a predefined gold standard (Antunes et al., 2015). Here, we use it to determine the similarity between the expected reference model *R** and the reference model *R* that is computed by the respective inductive approach (Martens et al., 2015). Precision measures the percentage of relevant elements in the computed reference model, whereas recall measures the percentage of elements in the expected reference model that are correctly discovered. The F-measure is defined as the harmonic mean of precision and recall, integrating and balancing them into a single measure.

$$precision = \frac{|R^*| \cap |R|}{|R|} \qquad recall = \frac{|R^*| \cap |R|}{|R^*|} \qquad F - measure = \frac{2 * precision * recall}{precision + recall}$$

Precision, Recall and F-measure can be computed individually for all elements of an EPC, i.e. events, functions, connectors, or edges, or for the EPC as a whole. For all measures, a high value indicates a high degree of conformity between the computed and the expected reference model. Regarding the execution time, none of the computations for this evaluation took longer than a few seconds on a standard PC hardware architecture.

	Measure	Events	Functions	Connectors	Edges	All
Scenario 1	Precision	1.00	0.80	0.80	0.79	0.82
	Recall	1.00	0.89	0.80	0.82	0.83
	F-measure	1.00	0.84	0.80	0.80	0.82
Scenario 2	Precision	1.00	1.00	0.80	0.30	0.61
	Recall	1.00	1.00	0.67	0.29	0.59
	F-measure	1.00	1.00	0.73	0.30	0.60
Scenario 3	Precision	0.82	0.86	0.00	0.46	0.66
	Recall	0.69	0.60	0.00	0.22	0.40
	F-measure	0.75	0.71	n/a	0.30	0.49

Table 4. Summary of evaluation results

5 Discussion

The evaluation results reveal some major assets and drawbacks of the RMM-2 approach. The first evaluation scenario illustrates the capabilities of automated versus manual approaches to inductive reference model development as well as the advantages of semantic in comparison to structural approaches. While manual approaches require a lot of resources, they are capable of considering context information. Automated approaches are fast and do not require additional resources, however, they have to rely on the provided information and cannot incorporate additional domain knowledge. Semantic approaches do not rely on structural similarity measures or uniform graph structures to represent process model similarity. However, an increased flexibility does not guarantee an optimal reference model. In addition, they rely on semantically correct process models to be able to produce results.

The RMM-2 approach is able to reconstruct the rather complicated second part of the reference model of scenario 1 (marked grey in figure 4). It consists of nested and interrelated loops and operator blocks, which pose a remarkable challenge to graph-based approaches. The ability to focus on model behaviour and reconstruct it is one of the major strengths of RMM-2. We still need to account for the differences between the computed reference model and the benchmark model. The latter contains specific optimizations instead of subsuming the contents and behaviour of the given input models. Since our approach does not consider this type of context information, it cannot perform this kind of merge operations. However, these kinds of adaptions can easily be included in a manual post-processing step.

While the first evaluation scenario yields very good results in terms of high precision and recall values, not all of these numbers can be reproduced in the two remaining scenarios. First, for both of them, less than half the edges of the reference model can be discovered correctly. Second, whereas a high percentage of connectors are correctly discovered in the second scenario, not one of them is found in the third one. This means that the discovered structure differs considerably from the expected one. We can deduce that the quality of the resulting model is highly dependent on the quality of the input data. If the models differ greatly in terms of size, scope, or structure, RMM-2 is unable to identify many similarities between them and does not construct a meaningful model. This is at least in part due to the rather simple process discovery algorithm. More elaborate discovery techniques would probably yield better results. In addition, the mapping quality also influences the final reference model in terms that automated mapping approaches generally produce mappings of lower quality. If model analogies are not captured in a mapping, they cannot be included in the reference model.

Readers may notice that the above definitions do not contain OR-connectors. This does not have technical reasons, as the used semantics present a concise definition of the complex semantics of an OR-join. However, including OR-joins in our reference model would require additional conceptual considerations about the intended semantics. Since this contribution is meant to be a proof-of-concept, these considerations are left for future work.

On first sight, it seems counterintuitive that the first evaluation scenario yields so much better results than the second and third one. After all, the first scenario is the only one based on real-world process models, whereas the other two scenarios are constructed synthetically. However, the models from the first scenario exhibit the highest degree of similarity towards each other, while the models from the other two sets vary in terms of scope and structure. Hence, the applicability of inductive reference modelling approaches to specific use cases is not determined by how closely they resemble reality. Instead, difficulty is introduced by a higher degree of variation within the model set, independent from its modelling method. Hence, while RMM-2 produces viable reference models for scenarios 2 and 3, these models differ from the benchmark model, as they include more process variants. This is a general problem for practical use cases of inductive reference modelling, which needs to be addressed.

The higher the variation within the input model set, the harder it is to find an appropriate value for the parameters *t* and *n*. As users may put more relevance to less frequent model parts and vice versa, the manual post-processing step gains more relevance, as it enables users to adapt the reference model according to their preferences by adding or removing nodes and models parts.

6 Related Work

Several authors have already proposed methods for the inductive derivation of reference process models, employing different paradigms and techniques. The Graph-Edit Distance (GED) as a measure of process model similarity serves as a basis for some of them. Li et al. (2009) suggest an algorithm that reduces the average edit-distance between a given reference model and a set of process variants. Similarly, Ardalani et al. (2013) extend the minimal graph-edit distance towards a minimal cost of change. Non-heuristic approaches may work by identifying similarities among process models in terms of frequent substructures (Rehse et al., 2015). Genetic algorithms are the basis for another approach (Martens et al., 2014), which defines a fitness function, based on the graph-edit distance. Integer Programming (IP) provides the formal background for several approaches. One iteratively constructs a new reference model based on the pairwise degree of node proximity (Yahya and Bae, 2011). The second describes a genetic algorithm, performing random mutations based on node proximity (Yahya, Bae, et al., 2012). This approach is extended towards multi-objective reference models (Yahya, Wu, et al., 2012). Other authors, such as Song et al. (2008) or Gottschalk et al. (2008), rely on event log data from different systems rather than individual process models to mine their reference models.

Apart from reference modelling, some authors have already considered execution semantics as the base for process model comparison. For example, Mendling et al. (2007) introduce the idea of causal footprints to determine a degree of similarity between two process models. The contribution by Becker et al. (2011) defines a semantic process distance, completely abstracting from the model structure.

The techniques and algorithms of our proposed approach resemble some ideas that originate from the field of Process Mining. Conceptually, they follow different objectives and purposes. Process Mining aims at extracting As-Is-Processes in order to "pay attention to the alignment of model and reality" (Van der Aalst, 2011, p. 7). So, while Process Mining intends to monitor processes on an execution level, Inductive Reference Modelling focuses on the design stage, improving the reference model quality rather than treating it ex post. The developed reference model is not meant to depict a realistic model of the given domain or to represent every possible process instance or configuration. Instead, it should illustrate typical, but abstract process behaviour. On a technical level, Process Mining depends on the availability of event log data, whereas the approach on hand is based on the transformation of existing models. Many techniques such as behavioural profiles can be applied in both fields, however, appropriate parameters, pre- and post-processing steps are required to obtain meaningful results.

Similarly, our contribution is set apart from configurable process models (Rosemann and van der Aalst, 2007), which contain many different process variants, providing an integrated view on the domain. Approaches to Process Model Merging (La Rosa et al., 2013) use configurable models to consolidate a set of process models. Since such a model gives way to a multitude of different process configurations, Schunselaar et al. (2014) automatically prune the model for the optimal configuration. Methodically, while they are set out to find the optimal process instance from a plethora of configurations, we subsume the commonalities of a set of process models into a single reference model.

Configurable reference models are often interpreted as generally applicable, hence incorporating all possible model variants. Vom Brocke (2007) argues that the intended and required character of a reference model in fact depends on the situational context. He introduces several additional design principles for reference models that introduce a new level of flexibility to both the design and the application of reference models. Depending on the specific use case, process models may be configured, instantiated, aggregated, specialized or designed in analogy to the employed reference models. Accordingly, reference models may be intended to represent a superset, merge, subset or extract of their respective domain. This characterization gives way to a different interpretation of reference model development towards reference model mining, i.e. an opportunity to automatically derive reference models for a specific application context. Table 5 relates some of the articles mentioned in this section to the described design principles and suggests situations for their application. As shown, most techniques are not specific to one design principle. Since they are not sharply distinguished from one another, several principles may apply to a situation.

Technique	Situation	Design Principle
Mining Reference Model Configurations (Gottschalk et al., 2008)	A configurable reference model can be derived by analysing the intended system behaviour. In addition, situational configurations can also be suggested.	Configuration, Analogy
Optimization of Process Configurations (Schunselaar et al., 2014)	In a specific domain, an optimal process configuration can usually be determined according to the situationally relevant parameters.	Instantiation
Process Model Merging (La Rosa et al., 2013)	Merging a multitude of separate process variations into a consolidated model may be used to gain a better understanding and simplify the management of the respective application domain.	Aggregation, Configuration
Process Model Intersection (La Rosa et al., 2013)	In contrast, focussing on the intersection (or digest) of a set of variants enables stakeholders to analyse recurring patterns and manage common fragments.	Specialisation
Proposed Approach	A reference model representing the typical model behaviour for a given domain contains solution patterns that may be replenished and reused.	Specialisation, Analogy

Table 5. Potential of Mining Approaches for Reference Model Adaptation Techniques

7 Conclusion

This contribution is set out to introduce, define and evaluate a new approach to inductive reference model development, which is based on execution-semantic instead of structural identification of model commonalities. We motivate the development of a new approach with the apparent restrictions and shortcomings of existing approaches, which are mainly based on structural notions of process-model similarity. Our new approach is intended to be more flexible and pose fewer restrictions to the input models, for example by allowing loops and external mappings. As the evaluation shows, the RMM-2 approach fulfils these design objectives and is able to reconstruct even fairly complex graph structured, solely based on the intended process semantics.

Still, the approach, as presented here, holds a lot of potential for improvements on both a conceptual and a technical level. Technically, the discovered model does not allow for all possible control flows and the discovery algorithm could be designed to be more flexible and more elaborate. The computation of behavioural profiles could be both simplified and optimized by employing the Refined Process Structure Tree (Vanhatalo et al., 2009; Weidlich et al., 2010). Conceptually, including OR-connectors would introduce a new level of flexibility into the approach, but would also require some additional conceptualisation. The inability to consider models with varying degrees of abstraction is one of the major drawbacks of all existing approaches. Whereas this does not rule out all potential use cases, it limits them considerably. Developing approaches to cope with varying degrees of abstraction is one of the major challenges when applying inductive reference model development in real-world scenarios. Examples of such use cases include aligning service processes in public administrations, integrating parallel ERP systems, or standardizing processes in multinational companies (Rehse et al., 2015).

The design principles for reference models by Vom Brocke (2007) are an additional point of future research. As we have seen, there are numerous different techniques and approaches that could be used for situational and adaptive reference model mining. Table 5 gives an impression of a potential new research direction. Existing techniques could be related to different design principles and characterizations of reference models, such that we obtain a toolbox for mining reference models for specific use cases and contexts. This would not only introduce a new level of applicability to inductive reference modelling, but also widen the acceptance and distributions of reference models themselves.

Acknowledgement: The research described in this paper was partly supported by a grant from the German Research Foundation (DFG), project name: "Konzeptionelle, methodische und technische Grundlagen zur induktiven Erstellung von Referenzmodellen (Reference Model Mining)", support code GZ LO 752/5-1. The authors would also like to thank the anonymous reviewers for their valuable comments which helped to improve this paper.

8 References

Antunes, G., Bakhshandelh, M., Borbinha, J., Cardoso, J., Dadashnia, S., De Francescomarino, C., Gragoni, M., et al. (2015), "The Process Model Matching Contest 2015", in Kolb, J., Mendling, J. and Leopold, H. (Eds.), *Proceedings of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures*, presented at the EMISA-15, Köllen Druck+Verlag GmbH, Bonn.

Ardalani, P., Houy, C., Fettke, P. and Loos, P. (2013), "Towards a Minimal Cost of Change Approach for Inductive Reference Model Development", *Proceedings of the 21st European Conference on Information Systems*, AIS, Utrecht.

Ardalani, P., Thaler, T., Fettke, P. and Loos, P. (2014), "EPC Generator: A concept to generate Event-driven Process Chains based on an original process model", in Suhl, L. and Kundisch, D. (Eds.), *Tagungsband Multikonferenz Wirtschaftsinformatik 2014*, Universität Paderborn.

Becker, J., Bergener, P., Breuker, D. and Räckers, M. (2011), "On Measures of Behavioral Distance between Business Processes.", *Wirtschaftsinformatik Proceedings*, presented at the Paper 48, available at: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1070&context=wi2011 (accessed 4 April 2016).

Becker, J. and Meise, V. (2011), "Strategy and Organizational Frame", *Process Management. A Guide for the Design of Business Processes, J. Becker, M. Kugeler and M. Rosemann (eds.), Springer, Berlin*, pp. 91–132.

Becker, J. and Schütte, R. (1997), "Reference Information Systems for Retail: Definition, Use and Recommendations for Design and company-specific Adaption of Reference Models", *Wirtschaftsinformatik*, Springer, pp. 427–448.

Fettke, P. and Loos, P. (2007), "Perspectives on Reference Modeling", in Fettke, P. and Loos, P. (Eds.), *Reference Modeling for Business Systems Analysis*, Idea Group Publishing, pp. 1 – 20.

Gottschalk, F., Van Der Aalst, W. and Jansen-Vullers, M. (2008), "Mining Reference Process Models and Their Configurations", in Meersman, R., Tari, Z. and Herrero, P. (Eds.), *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, Vol. 5333, pp. 263–272.

Gröger, S. and Schumann, M. (2014), *Entwicklung eines Referenzmodells für die Gestaltung des Drittmittel-Prozesses einer Hochschule und Ableitung von Einsatzgebieten für Dokumenten- und Workflow-Management-Systeme*, Professur für Anwendungssysteme und E - Business, Georg - August - Universität Göttingen.

Hevner, A., March, S., Park, J. and Ram, S. (2004), "Design science in information systems research", *MIS Quarterly*, Vol. 28 No. 1, pp. 75–105.

La Rosa, M., Dumas, M., Uba, R. and Dijkman, R. (2013), "Business process model merging: an approach to business process consolidation", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol. 22 No. 2.

Li, C., Reichert, M. and Wombacher, A. (2009), "Discovering Reference Models by Mining Process Variants Using a Heuristic Approach", in Dayal, U., Eder, J., Koehler, J. and Reijers, H. (Eds.), *Business Process Management : 7th International Conference, BPM 2009, Ulm, Ger-*

many, September 8-10, 2009. Proceedings, Vol. 5701, Springer, Berlin, Heidelberg, pp. 344–362.

Martens, A., Fettke, P. and Loos, P. (2014), "A Genetic Algorithm for the Inductive Derivation of Reference Models Using Minimal Graph-Edit Distance Applied to Real-World Business Process Data", in Kundisch, D. and Suhl, L. (Eds.), *Tagungsband Multikonferenz Wirtschaftsinformatik 2014*, Universität Paderborn.

Martens, A., Fettke, P. and Loos, P. (2015), "Inductive Development of Reference Models Based on Factor Analysis", in Thomas, O. and Teuteberg, F. (Eds.), *Proceedings Der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015)*, Vol. 12, presented at the Internationale Tagung Wirtschaftsinformatik (WI-2015), Universität Osnabrück, Osnabrück, Osnabrück, Germany, pp. 438 – 452.

Mendling, J. (2007), *Detection and Prediction of Errors in EPC Business Process Models*, Wirtschaftsuniversität Wien.

Mendling, J. and van der Aalst, W. (2006), "Towards EPC Semantics based on State and Context.", in Nüttgens, M., Rump, F.J. and Mendling, J. (Eds.), *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains*, German Information Society, Vienna, pp. 25–48.

Mendling, J., van Dongen, B.F. and van der Aalst, W.M. (2007), "On the Degree of Behavioral Similarity between Business Process Models.", in Nuettgens, M., Rump, F.J. and Gadatsch, A. (Eds.), *Proceedings of 6th Workshop on Event- Driven Process Chains (WI-EPK '07)*, Vol. 303, Gesellschaft für Informatik, pp. 39–58.

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007), "A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems*, Vol. 24 No. 3, pp. 45–77.

Rehse, J.-R., Fettke, P. and Loos, P. (2015), "A graph-theoretic method for the inductive development of reference process models", *Software & Systems Modeling (Online First)*, available at: http://link.springer.com/article/10.1007/s10270-015-0490-0 (accessed 21 September 2015).

Rosemann, M. and van der Aalst, W.M. (2007), "A configurable reference modelling language", *Information Systems*, Vol. 32 No. 1, pp. 1–23.

Schunselaar, D.M.M., Verbeek, H.M.W., Reijers, H.A. and Aalst, van der W. (2014), "Using Monotonicity to find Optimal Process Configurations Faster", in Accorsi, R., Ceravolo, P. and Russo, B. (Eds.), *4th International Symposium on Datadriven Process Discovery and Analysis*, Vol. 1293, pp. 123 – 137.

Song, W., Liu, S. and Liu, Q. (2008), "Business process mining based on simulated annealing", *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, IEEE, pp. 725–730.

Van der Aalst, W.M. (2011), *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer.

Vanhatalo, J., Völzer, H. and Koehler, J. (2009), "The refined process structure tree", *Data & Knowledge Engineering*, Vol. 68 No. 9, pp. 793–818.

Vom Brocke, J. (2007), "Design principles for reference modeling: reusing information models by means of aggregation, specialisation, instantiation, and analogy", in Fettke, P. and Loos, P. (Eds.), *Reference Modeling for Business Systems Analysis*, Idea Group Publishing, pp. 47–75.

Weidlich, M., Mendling, J. and Weske, M. (2011), "Efficient consistency measurement based on behavioral profiles of process models", *Software Engineering, IEEE Transactions on*, Vol. 37 No. 3, pp. 410–429.

Weidlich, M., Polyvyanyy, A., Mendling, J. and Weske, M. (2010), "Efficient computation of causal behavioural profiles using structural decomposition", *Applications and Theory of Petri Nets*, Springer, pp. 63–83.

Weske, M. (2012), *Business Process Management: Concepts, Languages, Architectures*, 2nd ed., Springer-Verlag, Berlin, Heidelberg.

Yahya, B.N. and Bae, H. (2011), "Generating Reference Business Process Model Using Heuristic Approach Based on Activity Proximity", *Intelligent Decision Technologies*, Springer, pp. 469–478.

Yahya, B.N., Bae, H., Bae, J. and Kim, D. (2012), "Generating valid reference business process model using genetic algorithm", *International Journal of Innovative Computing, Information and Control*, Vol. 8 No. 2, pp. 1463–1477.

Yahya, B.N., Wu, J.-Z. and Bae, H. (2012), "Generation of Business Process Reference Model Considering Multiple Objectives", *Industrial Engineeering & Management Systems*, Vol. 11 No. 3, pp. 233–240.