Measuring the Compliance of Processes with Reference Models

Kerstin Gerke, Jorge Cardoso, and Alexander Claus

SAP AG, SAP Research, CEC Dresden,
Chemnitzer Str. 48, 01187 Dresden, Germany
mail@kerstin-gerke.de, jcardoso@dei.uc.pt, mail@alexander-claus.de

Abstract. Reference models provide a set of generally accepted best practices to create efficient processes to be deployed inside organizations. However, a central challenge is to determine how these best practices are implemented in practice. One limitation of existing approaches for measuring compliance is the assumption that the compliance can be determined using the notion of process equivalence. Nonetheless, the use of equivalence algorithms is not adequate since two models can have different structures but one process can still be compliant with the other. This paper presents a new approach and algorithm which allow to measure the compliance of process models with reference models. We evaluate our approach by measuring the compliance of a model currently used by a German passenger airline with the IT Infrastructure Library (ITIL) reference model and by comparing our results with existing approaches.

1 Introduction

Reference models have gained increasing attention, because they make a substantial contribution to design and execute processes efficiently. Obviously, reference models are useful, but to which extent are these best practices adopted and implemented in a specific business context? Process mining algorithms [1,17] have shown a considerable potential for assessing the compliance of instances with reference models. The instances are typically recorded by process-aware IS and serve as a starting point for reconstructing an as-is process model. The derived model can be compared with other models (e.g. reference models) using existing algorithms to determine the equivalence of processes. Nevertheless, the results of a former compliance analysis using process mining and equivalence algorithms are not sufficient [11]. Our previous studies have evaluated the compliance of an as-is process model of a passenger airline with a reference model, which had incorporated the fundamentals of ITIL [15]. We found that the techniques available yield low values of compliance which could not be confirmed by the passenger airline. This difference was mainly due to: (1) different levels of details, (2) partial view of process mining, and (3) overemphasis of the order of activities. First, the level of detail characterizing a process differs widely when comparing a reference model with an as-is or to-be process model. Second, the derived as-is model only partially represents the processes of the airline. The execution of the processes

R. Meersman, T. Dillon, P. Herrero (Eds.): OTM 2009, Part I, LNCS 5870, pp. 76–93, 2009.

does not only result in log files but it also results in written record files, manual activities as well as human knowledge. Information outside the reach of process mining algorithms may compromise the results of compliance. Finally, reference models typically do not state whether dependencies between activities are compulsory. During our former studies [11] on compliance using existing equivalence algorithms, we have changed the order of activities in a reference model. While the compliance should remain the same since the reference model did not enforce a specific order for the execution of the activities, the compliance yielded different results.

This paper motivates the reader for the importance of measuring the compliance of process models with reference models. We also discuss the differences between process equivalence and process compliance and argue for the need of specific algorithms to measure the compliance between processes. We show that two models can have different structures but one process can still be compliant with the other. Furthermore, we develop a new approach and algorithm to overcome the drawbacks identified. We measure the compliance of an as-is process model of a German passenger airline with a reference model. To validate our methodology, we compare our compliance results with two existing approaches and explain why current algorithms are not suitable to evaluate the compliance.

The remainder of our paper is organized as follows. Section 2 introduces the fundamentals of reference models. Section 3 explains our methodology to measure compliance. The following section investigates the requirements for determining compliance. Sect. 5 presents and evaluates our rational and concept to develop a new algorithm. Sect. 6 describes the main related work. Finally, Sect. 7 formulates our conclusions based on our findings.

2 The Importance of Reference Models

Reference models offer a set of generally accepted processes which are sound and efficient. Their adoption is generally motivated by the following reasons. First, they significantly speed up the design of process models by providing reusable and high quality content. Second, they optimize the design as they have been developed over a long period and usually capture the business insight of experts [25]. Third, they ease the compliance with industry regulations and requirements and, thus, mitigate risk. Fourth, they are an essential mean to create a link between the business needs and IT implementations [25].

Reference models can be differentiated along their scope, their granularity, and the views, which are depicted in the model [25]. We distinguish (1) reference models focusing on capturing domain-specific best practices like ITIL, COBIT, and SCOR, and (2) configurable reference models, such as SAP Solution Manager [18], which aim at capturing the functionalities of a software system. Although the focus of this paper is on the first class of models, we explain both classes shortly with respect to their characteristics and their contribution to compliance.

The Information Technology Infrastructure Library (ITIL) is a set of guidance published as a series of books by the Office of Government Commerce. These books describe an integrated best practice approach to managing and controlling IT services [15]. The Control Objectives for Information and related Technology (COBIT) has been developed by the IT Governance Institute to describe good practices, to provide a process framework and to present activities in a manageable and logical structure. The Supply Chain Operations Reference Model (SCOR) provides a unique framework, which links business process and technology features into a unified structure to support communication among supply chain partners and to improve the effectiveness of supply chains [19].

A process is compliant in terms of the introduced reference models if the process is implemented as described by the reference model and the process and its results comply with laws, regulations and contractual arrangements [21]. Other popular reference models include the APQC Process Classification Framework SM (PCF) [2] and the Capability Maturity Model Integration (CMMI) [6].

The SAP Solution Manager of SAP NetWeaver [18] provides configurable reference models for business scenarios. Their usage ensures quality of the IT solution and enables traceability of all changes and, thus, compliance to the organizational needs. Most of the ERP vendors have similar approaches to support the configuration and implementation procedure of an IS landscape.

3 Methodology to Analyze Compliance

Based on our experiences with business processes of the air travel industry, we devised a generic approach and methodology to analyze the compliance between processes. The methodology identifies 5 entities, illustrated in Fig. 1, which need to be considered when measuring the compliance with reference models: the meta reference model M_0, the adopted reference model M_1, the to-be process model M_2, the instances of a process model M_2, and the as-is process model M_3. Depending on the scope, a meta reference model M_0 may provide either generally accepted processes or a set of abstract guidelines. In both cases, and particularly in the latter case, the reference model M_1 needs to be adapted to the needs of an organization yielding a set of processes M_2. The execution of the processes generates a set of instances. The analysis of these instances provides an as-is process model M_3 which reflects how a process M_2 was executed. The level of compliance can be measured by analyzing process models M_0, M_1, M_2, and M_3. Since M_0 is generally specified in natural language, we will concentrate our study on analyzing models M_1, M_2, and M_3.

Model M_1 and M_2 are mainly constructed manually, whereas M_3 is usually inferred from log files. These log files serve as a starting point for process mining algorithms, which aim at the automatic extraction of process knowledge. Various algorithms [1,17] have been developed and implemented in ProM [16] to discover different types of process models, for instance Petri nets [22] or Event-driven Process Chains (EPCs) [26]. ProM is a process mining workbench offering algorithms to discover and verify process models [26].

The level of compliance is expressed by a quality indicator, which can be incorporated into a maturity model, e.g. the COBIT maturity model "Manage

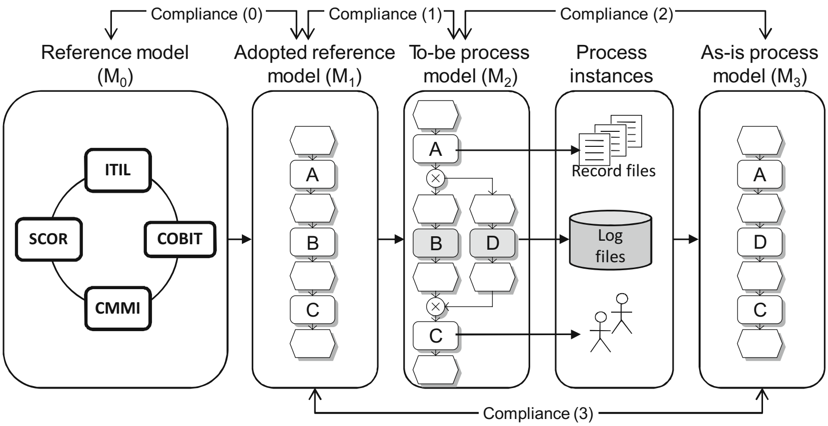

Fig. 1. Entities of a compliance analysis with reference models

Service Desk and Incidents" [21]. Such a model classifies the degree to which a process is aligned with a reference process. The level of compliance measured by the comparison of model M_2 or M_3 with M_1 serves as an initial estimate of the as-is compliance maturity. Opposing the as-is maturity and the to-be maturity supports the identification of potential improvements and contributes to determine alternative actions.

4 Requirements for a Compliance Analysis

We define process compliance as the degree to which a process model behaves in accordance to a reference model. The behavior is expressed by the instances, which can be generated by the model.

Figure 2 shows two EPCs capturing similar functionalities. Both are taken from the complaint handling process of a German passenger airline. The process is supported by the application "Interaction Center" (IAC) of the SAP Customer Relationship Management (CRM) system. The IAC facilitates the processing of interactions between business partners. Each interaction is registered as an activity. Besides a complaint description, further information, such as associated documents (e.g. e-mails), may be related to activities. Based on the characteristics of a complaint, an activity of the categories "Cust. Relations" or "Cust. Payment" is established. For example, complaints associated with payments are processed by the "Cust. Payment" department.

The EPC in the center of the figure shows model M_1, which depicts three activities: *Create incident*, *Categorize incident*, and *Prioritize incident*. The EPC on the right-hand side of the figure shows model M_2. Processing starts with an incoming complaint. Customers can complain by sending an e-mail or by filling an online form. In the latter case, the customer has to classify the complaint. In the former case, an employee has to read the e-mail to understand the complaint and determine the category manually. To measure the compliance, we need to discuss characteristics of business and reference models.

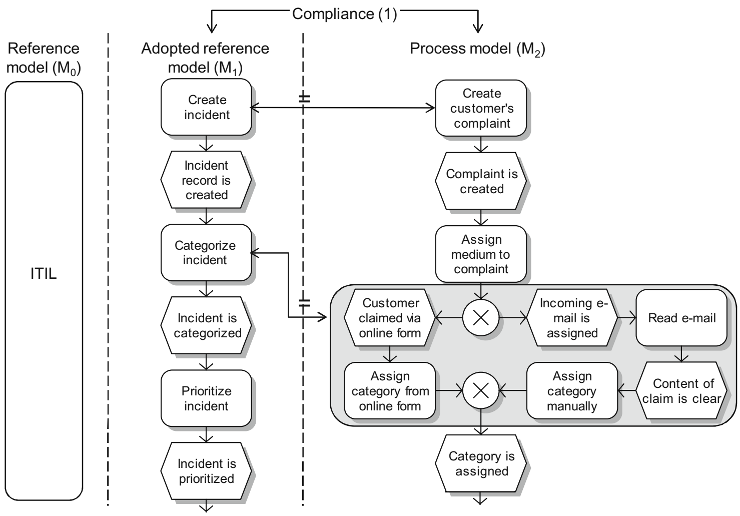

Fig. 2. The complaint handling process of a German passenger airline

Compliance Maturity and Degree. Our case study has identified two major concerns when it comes to evaluating compliance with reference models. First, the passenger airline wanted to learn if its processes followed the behavior recommended by the reference model. Second, the airline wanted to inquire if all the behavior recommended by the reference model was being implemented. In the context of compliance, we refer to the former as compliance degree and we denote the latter as compliance maturity. Let us consider the processing of incoming customer complaints. Model M_1 may recommend accepting complaints either via e-mail, letter or phone. If the airline accepts complaints via the first two mentioned communication channels only a part of the recommendations is implemented. We say that the airline is partially mature with respect to compliance maturity. But the ones currently being implemented (e-mail and letter) correspond to what the reference model M_1 recommends. In such a case, we say that the airline is fully compliant with respect to compliance degree.

Granularity of Models. Having two models M_1 and M_2 it may happen that the granularity characterizing the level of detail of activities varies. For example, in Fig. 2, activity *Prioritize incident* exists in model M_1, but no such activity exists in model M_2. Furthermore, it is possible that compliance applies to a set of activities, rather than individual activities. For example, activity *Categorize incident* of model M_1 corresponds to a set of activities in model M_2 highlighted in Fig. 2. In order to account for the granularity we have to identify the correspondence of activities. Correspondence is a mapping between activities of model M_2 to activities of model M_1 where the functionality of the activities is the same. Existing approaches, for example schema or semantic matching [26,9], assume that the correspondence can be established automatically based on the labels. The examples of our use case show that it is not realistic to only assume that equivalent activities may be identified by considering similarities of labels. For example, the activities *Create incident* in model M_1 and *Create customer's complaint* in model M_2 have the same functionality, but they have different labels. Since the automatic mapping is not applicable, we favor the manual mapping.

Customization of the Reference Model. It is often important to treat parts of model M_1 in a special way when measuring compliance. For example, since reference models do not typically state if the activities have to be executed exactly in a specified order, the order may not always be important. We refer to these special parts as partitions. A partition is a user-selected set of activities with a type, which can be "Order" or "Exclusion". Figure 3 shows that activities *Categorize incident* and *Prioritize incident* in partition P_1 may be executed in an arbitrary order. A partition of type "Exclusion" allows the definition of activities, which need to be excluded from the compliance analysis. Consider partition P_2. In our use case, the preprocessing of an incident is not supported by the IS right now. However, a manual activity corresponding to the functionality expressed by activity *Preprocess incident* is executed. To prevent the missing activity to erroneously affect the compliance, the activity is excluded.

Iteration. A special circumstance is the case in which an activity is part of an arbitrary cycle in process M_2 while it is not in model M_1. This means that this activity can be executed repetitively, while in model M_1 it must be performed correctly in only one iteration. For example, in our use case, the activities *Search for a solution* and *Inform Customer* are performed repeatedly until the customer accepts the processing of the claim. The existence of the cycle increases the quality of the process and contributes to a higher degree of the customer satisfaction. Thus, even if ITIL does not explicitly recommend a cycle, the airline feels that this cycle in model M_2 does not affect the compliance with model M_1 - a contrast with a cycle, which purely means to redo work. The latter cycle negatively affect the efficiency of a process. What makes it even more complicated is the fact that various reference models neither contain cycles nor state a precise number of recommended iterations. Without knowing the semantics of cycles it is not possible to state in general its effect on compliance.

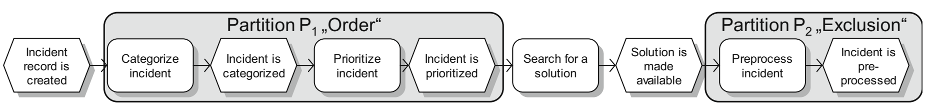

Fig. 3. Customization of reference model M_1

5 Sequence-Based Compliance

Based on requirements from Sect. 4 we have developed an algorithm to measure the compliance of model M_2 or M_3 with model M_1. Its main characteristic is that two models can have different structures but the algorithm can still judge one process to be compliant with the other. Figure 4, for example, clearly illustrates that the process models are different, but we will show that they are compliant.

5.1 Theoretical Foundations

Previous sections have used the EPC language to model processes since it is easy to understand and it is widely used in the industry (e.g. the common language of our use case). We use a more formal approach based on WF-nets [22] for the design of the compliance algorithm. It is a formalism well suited to analyze processes since there is a vast amount of research done in this area. We define the degree of compliance based on the firing sequences of WF-nets.

Definition 1 (WorkFlow net)
A WorkFlow net (WF-net) *is a tuple* $M = (P, T, F, i, o)$ *such that:*

- P *is a finite set of* places*,*
- T *is a finite set of* transitions*,*
- $P \cap T = \emptyset$*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a set of arcs,*
- $i \in P$ *is the unique* source *place such that* $\bullet i = \emptyset$*,*
- $o \in P$ *is the unique* sink *place such that* $o\bullet = \emptyset$*,*
- *Every node* $x \in P \cup T$ *is on a path from* i *to* o*,*

where for each node $x \in P \cup T$ *the set* $\bullet x = \{y|(y, x) \in F\}$ *is the* preset *of* x *and* $x\bullet = \{y|(x, y) \in F\}$ *is the* postset *of* x*.*

Transitions represent the activities of an instance. The input place (i) and the output place (o) of the WF-net express the entry point when instances are created and the exit point when instances are deleted. The last requirement ensures that there are no transitions and places which do not contribute to processing.

Definition 2 (Firing sequence)
Let $M = (P, T, F, i, o)$ *be a WF-net and let* $t \in T$ *be a transition of* M*.*

- *A* marking $K : P \to \mathbb{N}$ *is a mapping defining the number of tokens per place.*
- t *is* enabled *in a marking* K *if* $(\forall p \in \bullet t)\, K(p) \geq 1$*.*
- t fires from *marking* K to *marking* K'*, denoted by* $K[t\rangle K'$*, if* t *is enabled in* K *and* $(\forall p \in \bullet t)\, K'(p) = K(p) - 1$ *and* $(\forall p \in t\bullet)\, K'(p) = K(p) + 1$*.*
- $\sigma = \langle t_1, t_2, \ldots, t_n\rangle \in T^*$ *is a* firing sequence *leading from a marking* K_1 *to a marking* K_{n+1}*, denoted by* $K_1[\sigma\rangle K_{n+1}$*, if there are markings* $K_2, \ldots, K_n$*, such that* $K_1[t_1\rangle K_2[t_2\rangle \ldots K_n[t_n\rangle K_{n+1}$*.*

To capture relevant behavior we restrict ourselves to firing sequences representing process instances, which are terminated properly.

Definition 3 (Complete sound firing sequences). *Let* $M = (P, T, F, i, o)$ *be a WF-net and* $\sigma \in T^*$*.*

- K_i *is the* initial marking *with* $K_i(i) = 1$ *and* $(\forall p \neq i)\, K_i(p) = 0$*.*
- K_o *is the* final marking *with* $K_o(o) = 1$ *and* $(\forall p \neq o)\, K_o(p) = 0$*.*

– *σ is a* complete sound firing sequence*, if $K_i[\sigma\rangle K_o$.*
– *Let us use $S(M)$ to denote the set of all complete sound firing sequences.*

This definition ignores unsound behavior, for instance process instances running into a deadlock or a livelock. When no ambiguity occurs, we simply refer to σ as a firing sequence.

Since WF-nets can be considered as directed graphs, where $P \cup T$ is the set of nodes and F is the set of arcs, we use the standard graph-theoretical notion of a cycle.

Definition 4 (Cycle). *A* cycle *in a WF-net $M = (P, T, F, i, o)$ is a sequence of nodes $(x_1, \ldots, x_n) \in (P \cup T)^*$, such that $(\forall 1 \leq i < n)\,(x_i, x_{i+1}) \in F$ and $x_1 = x_n$.*

The existence of cycles causes the set $S(M)$ to be in general infinite. Therefore, we restrict the number of unroll factors for cycles by a variable parameter[1]. We end up with a finite subset of $S(M)$ denoted by $S'(M)$. The set $S'(M)$ grows exponentially in the number of transitions $|T|$. However, Sect. 5.5 will show that our approach can be used in practice. Our strategy to deal with cycles and their contribution to compliance among competing requirements (see Sect. 4) is to equate cycles having no correspondence in model M_1 with the action of redoing work. The superfluous work may have a negative effect on the compliance values.

5.2 Measuring Compliance

To account for the special characteristics of compliance with reference models, which we have identified in Sect. 4, we use several parameters to our algorithm.

Definition 5 (Granularity mapping). *Let be $M_1 = (P_1, T_1, F_1, i_1, o_1)$ and $M_2 = (P_2, T_2, F_2, i_2, o_2)$ two WF-nets where we refer to M_1 as the* reference model *and to M_2 as the* process model*. We use a mapping $\mathcal{G} : T_2 \rightarrow T_1$ to map activity labels in the process model to activity labels in the reference model. Since $\mathcal{G}$ can be non-injective, this mapping can handle granularity differences between the two models. Let us use the term* granularity mapping *for $\mathcal{G}$.*

Definition 6 (User-selected partition). *Let M_1 be a reference model as stated in Def. 5. A* user-selected partition *of M_1 is a set of transitions $p \subseteq T_1$ which can be of type* exclusion *or* order*. User-selected partitions of type exclusion are represented with $\bar{p}$ and those of type order with $\check{p}$. M_1 can have associated with it at most one user-selected partition of type exclusion and an arbitrary finite number of user-selected partitions of type order. Let us use $\mathcal{P}$ to denote the set of all user-selected partitions associated with M_1.*

Now that we have defined the parameters we deduce the compliance measures.

[1] We omit the parameter here and in subsequent equations since it has no significant effect to the equations and we want to keep them readable.

Definition 7 (Extended firing sequence set, Mapped firing sequence set). *Let M_1 and M_2 be the reference model and the process model as stated in Def. 5. Let $\mathcal{P}$ be the set of all user-selected partitions related to M_1 and let $\mathcal{G}$ be the granularity mapping between M_1 and M_2. Let $\sigma_1 \in T_1^*$ and $\sigma_2 \in T_2^*$.*

- $\sigma_1^{\text{ext}}(\mathcal{P})$ *is the* set of extended firing sequences *of σ_1, which is derived from σ_1 by applying two actions to σ_1: (1) remove the transitions in $\bar{p}$ from σ_1 and (2) generate the permutations of $\sigma_1 \setminus \bar{p}$ for all user-selected partitions $\check{p}$.*
- *Let us use $|\sigma_1|_{\text{ext}} = |\sigma_1'|$ $(\sigma_1' \in \sigma_1^{\text{ext}}(\mathcal{P}))$ to denote the length of an arbitrary extended firing sequence σ_1' of σ_1.*
- $\sigma_2^{\text{map}}(\mathcal{G})$ *is the* set of mapped firing sequences *of σ_2, which is derived from σ_2 by applying $\mathcal{G}$ to all transitions of σ_2, whereas for each subsequence of transitions of σ_2, which are mapped to the same transition $t_1 \in T_1$ only one occurrence of t_1 is placed in the resulting sequences, but possibly at different positions resulting in several mapped sequences.*
- *Let us use $|\sigma_2|_{\text{map}} = |\sigma_2'|$ $(\sigma_2' \in \sigma_2^{\text{map}}(\mathcal{G}))$ to denote the length of an arbitrary mapped firing sequence σ_2' of σ_2.*

Note, that $|\sigma_1|_{\text{ext}}$ is well defined. The length of all extended sequences $\sigma_1' \in \sigma_1^{\text{ext}}(\mathcal{P})$ is equal since they differ only in the order of transitions. The same holds for $|\sigma_2|_{\text{map}}$. Removing transitions by $\bar{p}$ guarantees $|\sigma_1|_{\text{ext}} \leq |\sigma_1|$ and the mapping of possible multiple transitions to one transition ensures $|\sigma_2|_{\text{map}} \leq |\sigma_2|$.

Definition 8 (Compliance measures). *Let M_1, M_2, $\mathcal{G}$ and $\mathcal{P}$ as stated in the definitions above. Let $\sigma_1 \in T_1^*$ and $\sigma_2 \in T_2^*$.*

- *The* firing sequence compliance (fsc) *of σ_2 w.r.t. σ_1 is:*

$$\text{fsc}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G}) = \max\{\text{lcs}(s, s') | s \in \sigma_1^{\text{ext}}(\mathcal{P}), s' \in \sigma_2^{\text{map}}(\mathcal{G})\} \ . \tag{1}$$

- *The* firing sequence compliance degree (fscd) *of σ_2 w.r.t. σ_1 is:*

$$\text{fscd}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G}) = \frac{\text{fsc}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G})}{|\sigma_2|_{\text{map}}} \ . \tag{2}$$

- *The* firing sequence compliance maturity (fscm) *of σ_2 w.r.t. σ_1 is:*

$$\text{fscm}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G}) = \frac{\text{fsc}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G})}{|\sigma_1|_{\text{ext}}} \ . \tag{3}$$

- *The* compliance degree (cd) *of M_2 w.r.t. M_1 is given by:*

$$\text{cd}(M_2, M_1, \mathcal{P}, \mathcal{G}) = \frac{\sum_{\sigma_2 \in S'(M_2)} \max_{\sigma_1 \in S'(M_1)}\{\text{fscd}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G})\}}{|S'(M_2)|} \ . \tag{4}$$

- *The* compliance maturity (cm) *of M_2 w.r.t. M_1 is given by:*

$$\text{cm}(M_2, M_1, \mathcal{P}, \mathcal{G}) = \frac{\sum_{\sigma_1 \in S'(M_1)} \max_{\sigma_2 \in S'(M_2)}\{\text{fscm}(\sigma_2, \sigma_1, \mathcal{P}, \mathcal{G})\}}{|S'(M_1)|} \ . \tag{5}$$

Function lcs in (1) calculates the length of the longest common subsequence of two firing sequences, thereby finding the maximum number of identical activities while preserving the activity order. The greater the value returned, the more similar the firing sequences are. See [4] for details on lcs. Since the firing sequences σ_1 and σ_2 can have various structures manifesting in their extended and mapped firing sequence sets, (1) will select the variation of σ_1 and σ_2 which will yield a greater similarity of σ_1 and σ_2. The compliance degree (2) of σ_2 indicates the extent to which the transitions of σ_2 are executed according to the specifications of a reference model expressed with σ_1. The compliance maturity (3) of a firing sequence σ_2 points at the extent to which the specification of a reference model expressed with σ_1 is followed by σ_2. In (4), (5), the degree and maturity of compliance express the ratio of instances, which can be produced by one model that can also be produced by the other model. From the viewpoint of compliance degree the process model is related to the reference model; from maturity vice versa. These compliance measures return a value in interval $[0, 1]$. For example, if the compliance degree is 1, the compliance is the highest since all firing sequences of model M_2 can also be produced by model M_1.

5.3 Industrial Application

This section applies the sequence-based compliance analysis to the case study introduced in Sect. 4 and compares the results with two existing approaches available in ProM: "Structural Precision/Recall" and "Footprint Similarity". We have chosen these two approaches since they are sometimes used to determine the compliance between models. We discuss the results of our study in Sect. 5.4.

Measuring Sequence-Based Compliance. Fig. 4 shows the starting point for the compliance analysis in ProM: two WF-nets. The left-hand side model portrays the reference model M_1, which was adopted from ITIL. Initially created as an EPC in the ARIS toolset, it has been converted into a WF-net and imported into ProM. The right-hand side model illustrates the as-is model M_3, which represents the complaint handling process of the passenger airline. It was extracted with the ProM plugin "Heuristic Miner" [16] from a log file containing 4,650 cases and 44,006 events being observed over a period of one year.

To adapt the reference model to the needs of the airline, model M_1 was customized as follows. The activity *Identify responsible employee* was excluded because the activity was not recorded by the IS. The airline assumes that the activities *Inform customer* and *Preprocess incident* may be executed in an arbitrary order. As a result, the airline has agreed on a user-selected partition of type exclusion ($\bar{p} = \{$*Identify responsible employee*$\}$) as well as on a partition of type order ($\breve{p} = \{$*Inform customer, Preprocess incident*$\}$). Besides the user-selected partitions, the left-hand side of Fig. 5 shows the granularity mapping. Please note, that the figure denotes the as-is model M_2. During the mapping, we found typical characteristics in the airline process discussed in Sect. 4: missing and additional activities and activities with different levels of detail. For example, the activity *Prioritize incident* is missing in model M_3 and the activities

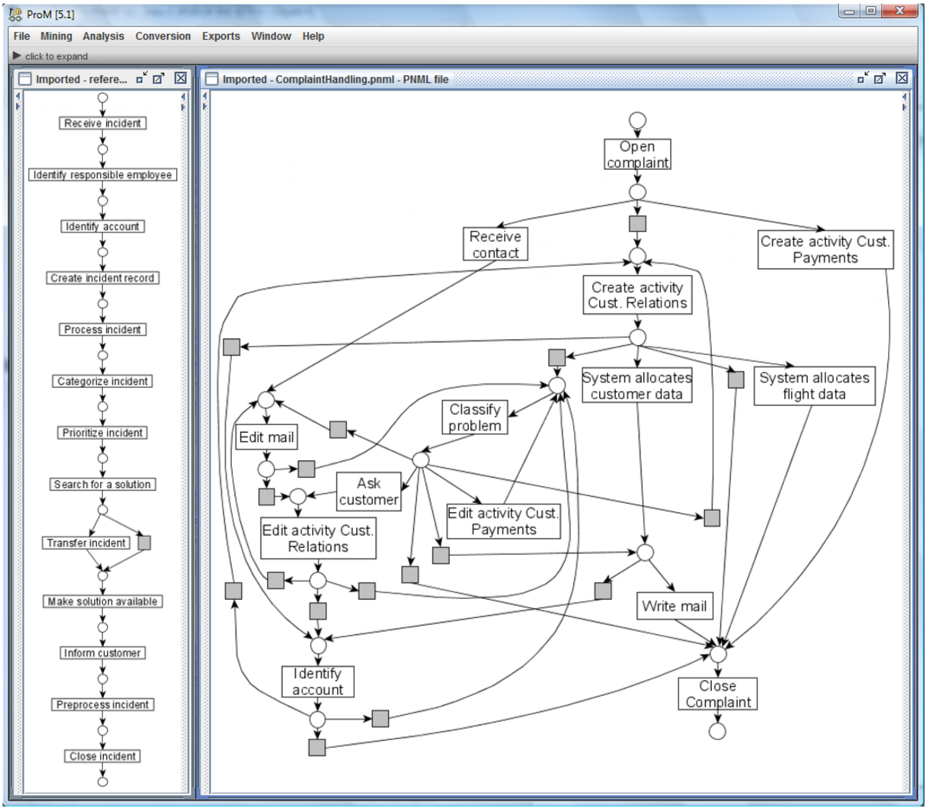

Fig. 4. Process models in ProM used for compliance analysis

Create activity Cust. Relations and *Create activity Cust. Payments* of model M_3 correspond to the activity *Create incident* in model M_1. Figure 4 shows that the airline uses iterations: model M_3 has cycles. Since the cycles are seen as quality improvement, the limit for cycle unrolling is set to 1. This limit ensures that all activities are considered but that the iteration of activities is not punished.

The right-hand side of Fig. 5 illustrates the results of our compliance analysis. Visible are the compliance degree and compliance maturity, which were computed according to Equations (4 and 5) per passed cycle as well as the extended firing sequences $\sigma_1^{\text{ext}}(\mathcal{P})$ of model M_1 and the firing sequences $\sigma_3^{\text{map}}(\mathcal{G})$ of model M_3. Unrolling a cycle once, yields the compliance degree $cd(M_3, M_1, \mathcal{P}, \mathcal{G})$ of 0.82 and the compliance maturity $cm(M_3, M_1, \mathcal{P}, \mathcal{G})$ of 0.52. To explain these values, we study the first line of the sequences σ_1 and σ_3, respectively. We consider the following extended firing sequence $\sigma'_{1-1} = \langle$*Receive incident, Identify account, Create incident record, Process incident, Categorize incident, Prioritize incident, Search for a solution, Make solution available, Inform customer, Preprocess incident, Close incident*$\rangle$ and $\sigma''_{1-1} = \langle$*Receive incident, Identify account, Create incident record, Process incident, Categorize incident, Prioritize*

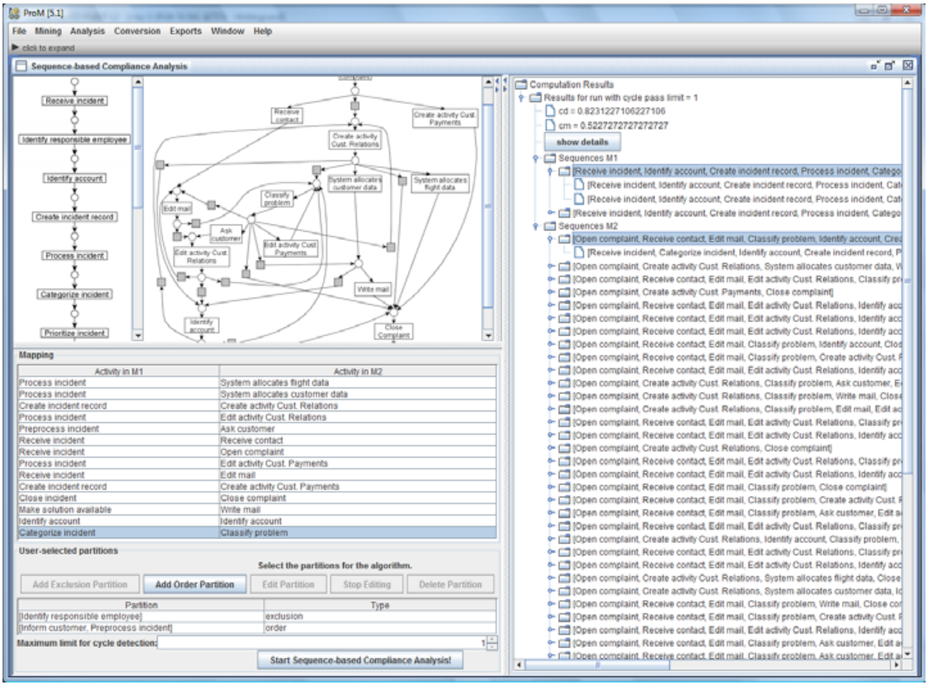

Fig. 5. Sequence-based compliance analysis plugin

incident, Search for a solution, Make solution available, Preprocess incident, Inform customer, Close incident$\rangle$, $\sigma'_{1-1}, \sigma''_{1-1} \in \sigma_1^{\text{ext}}(\mathcal{P})$. Let us also consider the firing sequence $\sigma_{3-1} = \langle$*Open complaint, Receive contact, Edit mail, Classify problem, Identify account, Create activity Cust. Relations, System allocates flight data, Close complaint*$\rangle$, which results in the firing sequence $\sigma'_{3-1} = \langle$*Receive incident, Categorize incident, Identify account, Create incident record, Process incident, Close incident*$\rangle \in \sigma_3^{\text{map}}(\mathcal{G})$. Since the maximum common longest subsequence of σ'_{1-1} and σ''_{1-1} with σ'_{3-1} corresponds to $\langle$*Receive incident, Identify account, Create incident record, Process incident, Close incident*$\rangle$, the firing sequence compliance $fsc(\sigma_{3-1}, \sigma_{1-1}, \mathcal{P}, \mathcal{G})$ is 5. The firing sequence compliance degree $fscd(\sigma_{3-1}, \sigma_{1-1}, \mathcal{P}, \mathcal{G})$ is $\frac{5}{6}$. This means that the instance σ_{3-1} of the as-is process model follows the order of the reference model with an overlap of 83%. The firing sequence compliance maturity $fscm(\sigma_{3-1}, \sigma_{1-1}, \mathcal{P}, \mathcal{G})$ is $\frac{5}{11}$. This means that only 45% of instance σ_{1-1} prescribed by the reference model are being followed by instance σ_{3-1} of the as-is process model. The result of the compliance degree of 82% indicates that the processes executed by the airline correspond to the recommendations of the reference model. We can say that, although the models M_3 and M_1 look different, the model M_3 is highly compliant with reference model M_1. The compliance maturity of 52% indicates that there are recommendations in reference model M_1 which are not implemented by the airline. Nonetheless, because of the maturity value of 52% we can conclude that model M_3 is also partially mature with reference model M_1.

Measuring Precision and Recall. In [24], the authors introduce the structural precision and recall. $Precision^{S}(M_1, M_2)$ is the fraction of connections in M_2 that also appear in M_1. If this value is 1, the precision is the highest because all connections in the second model exist in the first model. $Recall^{S}(M_1, M_2)$ is the fraction of connections in M_1, which also appear in M_2. If the value is 1, the recall is the highest because all connections in the first model exist in the second model. To analyze the compliance, model M_1 and M_3 of our use case need to be represented by a heuristic net. Therefore, we have converted model M_1, originally represented by an EPC, into a Heuristic net using ProM. Since the ProM plugin expects same labels, we have renamed the labels of model M_3 according to model M_1 and carried out the mapping depicted in Fig. 5. The structural precision obtained was 3% and the recall was 8%.

Measuring Causal Footprint. The causal footprint [26] is the second approach we have compared with our algorithm. The footprint identifies two relationships between activities: look-back and look-ahead links. This paper does not elaborate on the corresponding equation due to its complexity. We refer interested readers to [26]. Since the analysis of the causal footprint is based on comparing two EPCs, we have converted model M_3 into an EPC using a conversion plugin in ProM. The mapping was manually performed in accordance to the mapping shown in Fig. 5. To analyze the causal footprint, the ProM plugin "Footprint Similarity" was used and yielded a result of 27%.

5.4 Evaluation

This section discusses the compliance values, which we yielded in Sect. 5.3 based on the requirements from Sect. 4.

Precision and recall rely on the notion of equivalence and expect process models, which need to be compared, to be equal in their structure. This is the reason why the values obtained are relatively low: 3% and 8%, respectively. Similar to our approach these two measures allow to analyze the compliance from the perspectives compliance degree (i.e. precision) and compliance mature (i.e. recall). By contrast the approach neither offers a mapping functionality nor accounts for the necessary customization of the reference model: ordering or exclusion of activities. Expressing the behavior of a model in terms of connections results in the loss of information whether two connected transitions are part of a cycle and neglects the control flow of process models. However, these are relevant information when measuring the compliance with reference models.

The causal footprint also relies on the notion of equivalence. However, the approach assumes that process models with different structures may be similar. Therefore, the result of 27% is closer to the values obtained when using the algorithm we have developed (i.e. 82% and 52%). Since the formula is symmetric, measuring the compliance of model M_3 with model M_1 or of model M_1 with model M_3 yields the same value. It is clear that this situation is perfectly aligned with the notion of equivalence but fails to meet the requirements of determining compliance from the perspectives degree and maturity. Like our approach the notion of mapping is included. However, a non-injective mapping is not supported. Since the algorithm accounts for the ordering of activities, it partially fulfills the requirements for customization of reference models. Nonetheless, it does not account for the exclusion of activities. The authors [26] do not state the behavior of their formula with respect to cycles.

Using algorithms with the notion of equivalence, we are tempted to infer that the processes are not compliant. In contrast to the sequence-based compliance, the recall and precision and the causal footprint yield a value, which is little expressive and hard to explain. It is not possible to trace the missing or dissent instances. The solution proposed in this paper obtains two different values for compliance (i.e. degree and maturity) and also calculates intermediate results from instance compliance. This enables process designers to trace back which instances are affecting positively or negatively the compliance of the processes under analysis. The industrial application shows that the notion of equivalence cannot be used with satisfactory results to evaluate the compliance of processes with a reference model.

5.5 Feasibility Study

The sequence-based compliance algorithm is based on the generation of sets of firing sequences to describe the behavior of a process model. Unfortunately, in general, the size of these sets can grow exponentially with the size of the WF-net in terms of activities. This section shows the applicability of our algorithm in spite of its exponential complexity. Like Dijkman [8], we used a sample of EPCs of the SAP reference model to test whether our algorithm can be applied in practice by showing that the computation times are acceptable. The SAP reference model has been described in [20,10] and is referred to in many research papers (e.g. [8]). Since it is among the most comprehensive reference models covering over 600 business processes,we assume that these models can be regarded as a representative example. The study is performed by applying the sequence-based compliance algorithm to a subset of 126 pairs of EPCs from the SAP reference model, which we have converted to WF-nets. The pairs are put together based on their similarity computed by the ProM plugin "EPC Similarity Calculator". Our pairs are characterized with a similarity greater than 50%. Figure 6 shows the percentage of model pairs for which the compliance can be computed within a given number of milliseconds on a regular desktop computer. Ninety percent of the process models analyzed with our compliance algorithm took less than 62 milliseconds. In the experiment, the runtime of the algorithm takes on average 50.5 milliseconds with a standard deviation of 9.3 milliseconds. Figure 7 shows the runtime per activities in the processes of a model pair. The average number of activities in these processes is 16. We only found a weak correlation between runtime and the number of activities of a process. Therefore, we conclude that for the number of activities, which we found in the SAP reference models, the

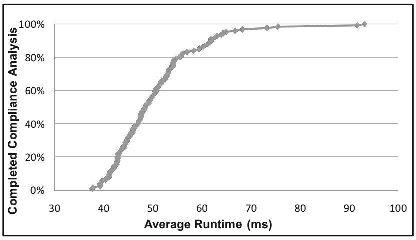

Fig. 6. Average runtime

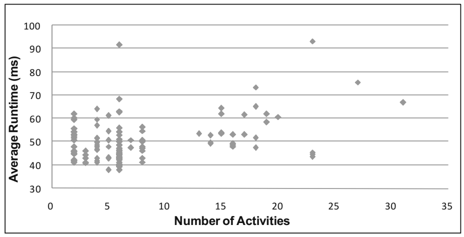

Fig. 7. Runtime as function of activities

sequence-based compliance analysis is applicable. These results show that, in theory we are confronted with exponential runtime when the complexity is measured in terms of the input size only, i.e. activities. However, in practice there are natural boundaries, e.g. the number of activities per process model is between a lower bound and an upper bound. Hence, the algorithm can be used in practice despite its exponential complexity.

An alternative to address complexity with regard to the input size of the algorithm is to capture the behavior of a model using the state space of a WF-net. A state space corresponds to the set of reachable markings of a WF-net [3]. The resulting graph is denoted as the reachability graph. Buchholz et al. [5] present a method focusing on optimizing the generation of the reachability graph of large Petri nets. The central idea is to decompose a net, to generate reachability graphs for the parts and to combine them. Furthermore, there exist various techniques for state space reduction [8], which may be exploited to improve the efficiency of the underlying algorithm of the sequence-based compliance algorithm. Corresponding approaches are referred to reduction rules. These rules aim at reducing the size of the state space by reducing the number of places and transitions preserving information relevant for analysis purpose. For example, it is possible to account for the significance of transitions. Transitions, which are rarely executed, can be left out using abstraction or encapsulation. Again, we found arguments for the applicability of state spaces in the context of the input size. For example, Verbeek et al. [28] argue that state spaces generating a reachability graph are often feasible for systems up to 100 transitions.

6 Related Work

Our work can be related to various research areas, namely process discovery and verification, process integration, and behavior inheritance.

Measuring compliance assumes the presence of a given model. Therefore, process mining, which aims at the discovery of such a model, is related to the work presented in this paper. Various algorithms have been developed to discover process models based on a log file [17,1].

In the literature, we have identified two ways to verify the compliance between processes and supporting IS: log-based verification and inter-model verification.

Since it is possible to verify if a model and a log file fit together, measuring the compliance can be seen as a very specific form of log-based verification. Thus, our paper is related to the work of Cook et al. [7] who have introduced the concept of process validation. They propose a technique comparing the event stream coming from the process model with the event stream from the execution log based on two different string distance metrics. The notion of compliance has also been discussed in the context of genetic mining [1]. Compliance checking is applied by using fitness, behavioral precision and recall. All these compliance measures propose some kind of replay of the instances in a Petri net. However, the applicability of the log-based verification presumes the existence of log files which are not always available. In the context of the inter-model verification van der Aalst introduces the delta analysis, which compares the real behavior of an IS with the expected behavior (e.g. a reference model) [23]. Different notions of equivalence of process models being subject to verification, such as trace equivalence [27], bisimulation [27], and behavioral equivalence [24], have been developed. The classical equivalence notions are defined as a verification property which yields yes or no, but do not provide a degree of equivalence [26]. Notions searching for behavioral similarity, for instance causal footprint [26] and structural appropriateness [17,24] are applicable in the context of process mining. However, they do not account for the characteristics of compliance with reference models. We introduced them in Sect. 5.3. For a detailed overview we refer to [24].

From a conceptual viewpoint, process integration and process inheritance are similar to our work. Comparing two process models in order to measure compliance in terms of corresponding behavior implies that there are distinctions. Common integration approaches for process models show how these distinctions can be integrated, for example to harmonize processes after an organizational merger [14]. In [8], Dijkman has categorized differences related to control flow, resource assignment, and activity correspondence and has presented a technique to diagnose these differences between process models. Juan [13] applied a string comparison approach of the firing sequences embedded in each process model to identify differences between process models. These works are complementary to our approach and can be considered together during the compliance analysis to locate the exact position of a difference between the models and analyze the type of a difference in the process models. However, since process integration approaches are designed for similar business situations, they typically focus on very similar processes on the same level of abstraction. Basten and van der Aalst [3] have introduced the relations of behavioral inheritance, which can also be used to identify commonalities and differences in process models. The approach is motivated by improving reusability and adaptivity of process models and concentrates on applying the idea of inheritance known from object-oriented modeling. The relations are based on labeled transition systems and branching bisimulation and correspond to the algebraic principles of encapsulation and abstraction [3]. Process inheritance assumes that process models originate from common sources and, therefore, are different yet very similar. Thus, notions of inheritance do not account for different level of granularities.

7 Conclusion and Future Work

Reference models provide valuable recommendations for the implementation of business processes. However, methods and solutions to determine how these guidelines are implemented in practice are non-existing. Known algorithms to evaluate the equivalence of processes have proven to be insufficient to measure compliance since many factors and characteristics related to compliance are ignored. In this paper, we have investigated the characteristics of compliance and we have devised a generic approach to analyze the compliance of process models with reference models. Our main contribution is an algorithm, called sequence-based compliance, which is based on the observation that process models can have different structures but one process can still be compliant with the other.

In order to validate our approach and our algorithm we have measured the compliance of a complaint handling process of a German passenger airline. The passenger airline has obtained transparency of its current customer support processes by carrying out process mining on their log files. Nonetheless, the next step, which needed to be executed, was to determine to which extent the process were aligned with a reference model (i.e. ITIL). This second step has been addressed in this paper.

We have further evaluated our methodology by comparing the results with two existing approaches. The validation was not trivial since we applied process mining and equivalence algorithms on real data. The results have shown that the sequence-based compliance yields more insightful values when compared to the results of existing algorithms based on analyzing the equivalence of processes.

In the future, we are planning to apply our approach and algorithm to other business and industry domains. We also aim to learn which additional types of customization of reference models are important and study how traceability can be incorporated into compliance analysis to enable organizations to quickly identify problematic parts of their running processes.

References

1. Alves de Medeiros, A.K., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic Process Mining: A Basic Approach and its Challenges. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 203–215. Springer, Heidelberg (2006)
2. APQC, American Productivity & Quality Center, `http://www.apqc.org/pcf`
3. Basten, T., van der Aalst, W.M.P.: Inheritance of Behavior. Journal of Logic and Algebraic Programming 47(2), 47–145 (2001)
4. Bergroth, L., Hakonen, H., Raita, T.: A Survey of Longest Common Subsequence Algorithms. In: 7th IEEE Intl. Symposium on String Processing Information Retrieval, pp. 39–48. IEEE Press, Los Alamitos (2000)
5. Buchholz, P., Kemper, P.: Hierarchical Reachability Graph Generation for Petri Nets. Form. Methods Syst. Des. 21(3), 281–315 (2002)
6. CMMI, Software Engineering Institute, `http://www.sei.cmu.edu/cmmi`
7. Cook, J.E., He, C., Ma, C.: Measuring Behavioral Correspondence to a Timed Concurrent Model. In: 17th IEEE Intl. Conf. on Software Maintenance, p. 332. IEEE Press, Los Alamitos (2001)

8. Dijkman, R.: Diagnosing Differences between Business Process Models. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 261–277. Springer, Heidelberg (2008)
9. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: 4th Asia-Pacific Conf. on Conceptual Modeling, pp. 71–80 (2007)
10. Teufel, T., Keller, G.: SAP R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley, Reading (1998)
11. Gerke, K., Tamm, G.: Continuous Quality Improvement of IT Processes based on Reference Models and Process Mining. In: 15th Americas Conf. on Information Systems (2009)
12. IDS Scheer AG, `http://www.ids-scheer.com`
13. Juan, Y.C.: A String Comparison Approach to Process Logic Differences between Process Models. In: 9th Joint Conference on Information Sciences (2006)
14. Mendling, J., Simon, C.: Business Process Design by View Integration. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 55–64. Springer, Heidelberg (2006)
15. Official Introduction to the ITIL Service Lifecycle. Stationery Office Books, London (2007)
16. ProM, `http://www.processmining.org`
17. Rozinat, A., Veloso, M., van der Aalst, W.M.P.: Evaluating the Quality of Discovered Process Models. In: Bridewell, W., et al. (eds.) 2nd Intl. Workshop on the Induction of Process Models, Antwerp, Belgium, pp. 45–52 (2008)
18. SAP AG, `http://www.sap.com`
19. Supply-Chain Council, Supply Chain Operations Reference Model, SCOR (2006)
20. Ladd, A., Curran, T., Keller, G.: SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Prentice Hall PTR Enterprise Resource Planning Series, Upper Saddle River (1997)
21. The IT Governance Institute, COBIT 4.1 (2007)
22. van der Aalst, W.M.P.: Verification of Workflow Nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
23. van der Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. Requir. Eng. 10(3), 198–211 (2005)
24. van der Aalst, W.M.P., Alves de Medeiros, A.K., Weijters, A.J.M.M.: Process Equivalence: Comparing Two Process Models Based on Observed Behavior. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 129–144. Springer, Heidelberg (2006)
25. van der Aalst, W.M.P., Dreiling, A., Gottschalk, F., Rosemann, M., Jansen-Vullers, M.: Configurable Process Models as a Basis for Reference Modeling. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 512–518. Springer, Heidelberg (2006)
26. van Dongen, B.F., Dikman, R., Mendling, J.: Measuring Similarity between Business Process Models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
27. van Glabbeek, R.J., Peter Weijland, W.: Branching Time and Abstraction in Bisimulation Semantics. Communications of the ACM 43(3), 555–600 (1996)
28. Verbeek, H.M.W.: Verification and Enactment of Workflow Management Systems. PhD thesis, University of Technology, Eindhoven, The Netherlands (2004)