Maintaining business process compliance despite changes: a decision support approach based on process adaptations

Tobias Seyffarth & Stephan Kuehnel

Published online: 29 Dec 2020.

Submit your article to this journal ↗

Article views: 320

View related articles ↗

View Crossmark data ↗

Taylor & Francis
Taylor & Francis Group

Maintaining business process compliance despite changes: a decision support approach based on process adaptations

Tobias Seyffarth and Stephan Kuehnel

Department of Management Information Systems and Operations Research, Martin Luther University Halle-Wittenberg, Chair of Information Management, Halle (Saale), Germany

ABSTRACT

The term compliance essentially refers to ensuring that business processes, operations and practices conform to an agreed set of rules. Such rules can influence both business processes and components of an information technology (IT) architecture, resulting in relationships between (1) compliance requirements, (2) process elements and (3) IT components. Whenever one element of these three classes is changed, e.g. when outsourcing decisions are made, a relationship analysis becomes necessary in order to identify demanding and violated compliance requirements. Since a manual relationship analysis is a complicated and elaborate task, the paper at hand presents methods to 1) automatically identify potential compliance violations in the context of changes and 2) automatically propose process adaptions for maintaining or re-establishing compliance. The methods are implemented as a software artefact, evaluated as useful in the context of an expert survey and contribute to the support of process adaptation decisions for maintaining compliance following changes.

1. Introduction

Business process compliance refers to the modelling and execution of business processes in accordance with regulatory requirements (Governatori & Sadiq, 2009). In an age of increasingly digitised business activities or even completely digital business processes, a great number of companies are confronted with increasing compliance requirements originating from the IT environment, and these regulations are both laborious and costly (Becker et al., 2016; Sackmann et al., 2018). To keep the effect of compliance on business activities minimal and to avoid a negative impact on profitability, compliance violations have to be prevented, especially in digital business fields that are characterised by frequent changes (Sackmann et al., 2018). However, the prevention of such violations is only possible if the dependencies between compliance requirements, business processes and IT components are known and can be continuously analysed.

A compliance requirement is a constraint or assertion that prescribes a desired result or purpose to be achieved by incorporating actions or control procedures in processes (Turetken et al., 2011). Compliance requirements not only place demands on business

---

CONTACT Tobias Seyffarth ✉ tobias.seyffarth@wiwi.uni-halle.de ▣ Martin Luther University Halle-Wittenberg, Chair of Information Management, Halle (Saale) 06108, Germany

processes, but also on components of information technology (IT) architecture, such as data protection and information security laws addressing the operation of software and/or hardware (Committee of Sponsoring Organizations of the Treadway Commission, 2012; Knackstedt et al., 2013; Sadiq et al., 2007; The Institut der Wirtschaftsprüfer in Deutschland e.V. [Institute of Public Auditors in Germany, Incorporated Association], 2002b).

Compliance requirements imposed on an IT infrastructure have at the least an indirect influence on business processes if an IT component is a prerequisite for process execution, as is the case with non-manual process activities. Consequently, there are direct and indirect dependencies between compliance requirements, IT components and process activities. Each of them require consideration as part of a comprehensive management of business process compliance. Outsourcing decisions, business process re-engineering, new technologies and many other factors can lead to changing compliance requirements, business activities, or IT components (Fdhila et al., 2015; Rudzajs & Buksa, 2011).

In dynamic markets, the rapid adaption of business activities and processes to such changes is seen as a competitive advantage (Rinderle et al., 2004). Thus, the fast detection of demanding compliance requirements (prior to a compliance breach) and violated compliance requirements (subsequent to a compliance breach) as well as the related adaption of business processes for maintaining compliance are important tasks (Sackmann & Kittel, 2015). However, due to the steadily rising level of regulation and increasingly complex business process models and IT architectures, this is becoming a challenging and time-consuming manual task (Elgammal et al., 2010; Ghanavati et al., 2009). In this context, IT support and automation open up the potential for time and cost savings.

There is already a body of literature, which provides a range of approaches that allow for querying the relationships between compliance requirements and business processes, compliance requirements and IT components as well as interrelated requirements (e.g. (Fdhila et al., 2012; Rudzajs & Buksa, 2011)). However, an analysis of these relationships is a prerequisite for the identification of demanding and violated compliance requirements. It is also necessary for process adaptations in order to maintain or re-establish business process compliance. To the best of our knowledge, so far there is no approach that allows for a comprehensive and automatic analysis of all direct and transitive relationships as well as subsequent process adaptation proposals.

Consequently, the goal of this research is a decision support approach that allows for identifying demanding and violated compliance requirements by means of design-time information, and for maintaining compliance despite changes in process activities, IT components, or regulations. In this study, we focus on the compliance change patterns 'replace' and 'delete', as their impact on compliance can be determined based on information from the underlying design-time models, i.e. previously modelled dependencies between compliance requirements, business processes, and IT components. In order to reach that goal, we raise the following research questions (RQ):

- RQ1: How to automatically determine compliance requirements and compliance violations by replacing or deleting business activities, IT components, or regulations?
- RQ2: How to automatically generate proposals for compliant business process models at design time?

To investigate the research questions, we apply the well-known design science research methodology proposed by (Peffers et al., 2006). In this study, we bring together models and methods to recommend compliant business processes based on integrating previously modelled alternative compliance processes of organisations. In addition, we demonstrate our artefacts in a proof-of-concept implementation called BCIT, which helps to eliminate potential naming ambiguities when integrating compliance processes into business processes. Finally, we evaluate our artefacts using a case study.

The paper is structured as follows: In Section 2, we introduce required preliminaries and present a motivating example. In Section 3, we present the related work and briefly describe the extent to which the related work is able to implement the motivation example. Section 4 addressed our applied research method, and Section 5, introduces a method that allows an automatic identification of demanding and violated compliance requirements triggered by changes. In Section 6, we introduce a method that allows for the automatic suggestion of business process adaptations to maintain or re-establish compliance subsequent to a compliance violation. Section 7 provides evaluation episodes of this research project and presents a case study based on which domain experts evaluated the perceived usefulness of our approaches. After discussing the implications of this contribution for research and practice in Section 8, we conclude the paper in Section 9.

2. Theoretical background

2.1. Theories

From a broader perspective, compliance is about unambiguously ensuring conformance to a set of prescribed and/or agreed upon requirements (Turetken et al., 2011). In this context, a compliance requirement is a constraint or assertion that prescribes a desired result or purpose that derives from a compliance source, such as a regulation, legislation or law (Turetken et al., 2011). As previously stated out, compliance requirements may place demands on business processes and/or information technology (IT).

The execution of business processes in adherence to applicable compliance requirements is called business process compliance (BPC) (Governatori & Sadiq, 2009). The operation of IT in adherence to applicable compliance requirements such as COSO (Committee of Sponsoring Organizations of the Treadway Commission, 2012) and COBIT (ISACA, 2013) is called IT compliance. From a business process management perspective, business process cannot be viewed separately from IT (Weske, 2019). Therefore, compliance requirements for IT must also considered. In this context, this includes BPC and IT compliance. In the following, we will only use the terms 'compliance' and 'compliance requirement' regardless of whether the core concept is BPC or IT compliance.

2.2. Interrelations between compliance, business processes and IT Components

Figure 1 shows our compliance meta-model which illustrates the relations between business activities, IT components, compliance requirements and other elements to which we refer throughout the paper.
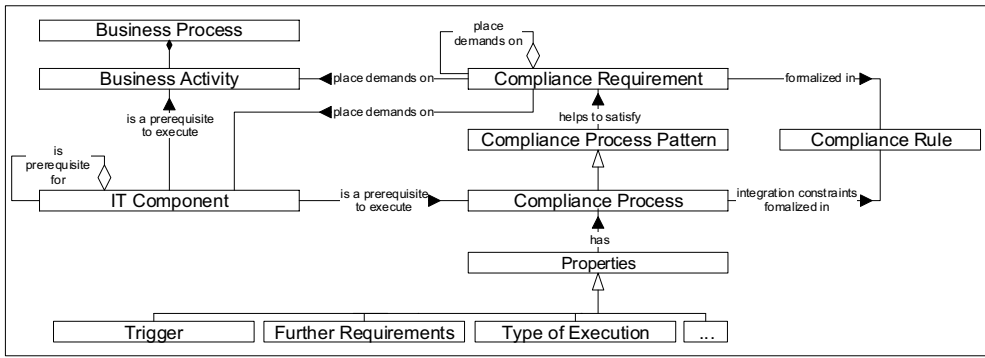
Figure 1. Compliance meta-model (based on (Seyffarth et al., 2016; 2017a)).

For reasons of simplicity, we refer to a single element within an IT architecture as an IT component and we do not distinguish between different types of IT components, which can be e.g. application services, application components or network devices such as special types of hardware (Winter & Fischer, 2006).

Various approaches check or ensure BPC. As an example, BPC can be checked after process execution by analysing log files (e.g. (El Kharbili et al., 2008)) or ensured at the design time of business processes, e.g. during modelling. A possible solution to ensure BPC at the design time of the business processes is the separate modelling of so-called compliance processes and its integration into business process models. In this context, a compliance process is defined as an independent process (part) consisting of at least one compliance-related activity that satisfies a compliance requirement (Seyffarth et al., 2017a). Whenever a business process activity is affected by a compliance requirement, a corresponding compliance process can be integrated in the business process (Sackmann & Kittel, 2015; Schumm et al., 2010a). Further, a compliance requirement can be satisfied by numerous alternative compliance processes. These can differ in a number of properties such as the kind of conditions to be verified, further necessary requirements for execution, and the type of execution (Seyffarth et al., 2017a).

The idea of separating compliance and business processes corresponds to a special kind of process modularisation which has the potential for reusing compliance processes or compliant process fragments to meet compliance requirements in various business processes (Schumm et al., 2010a). It is based on the assumption that the amount of business and compliance activities are disjointed, i.e. that an activity meets either a business or a compliance objective. If the pursuit of a compliance goal serves only value generation, it becomes difficult to distinguish between compliance and business activities. This might be the case when complying with requirements constitutes the unique selling proposition of a product or service. However, our approach focuses on cases where modularisation is possible.

2.3. Example model

Figure 2 shows a simplified purchase to pay process, which is modelled as a Business Process Model and Notation (BPMN) model, including perspectives on compliance requirements, a business process, a compliance process, and IT components. We are
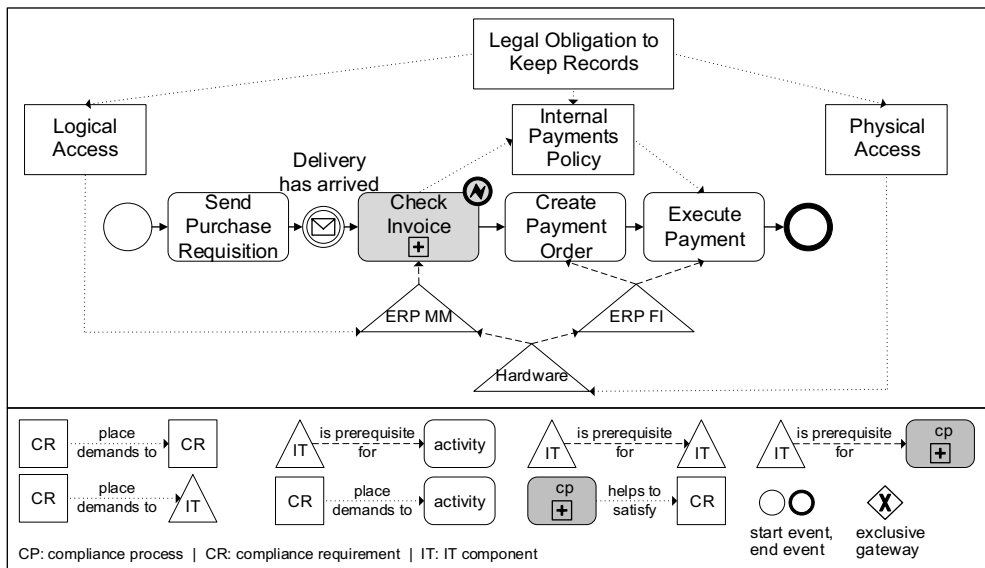
Figure 2. Simplified purchase to pay process (based on (Frank et al., 2009; Namiri & Stojanovic, 2007; Seyffarth et al., 2018; 2019)).

aware that the process might be overly simplified for any real scenario; however, the goal of the example model is to explain our methods that also works for more realistic and complex processes. Further explanations are based on this example.

In the example, some activities are supported by IT components that are modelled as triangles. On the one hand, we assume that the material management module of an enterprise resource planning system (ERP MM) is a prerequisite for the compliance process 'check invoice'. On the other hand, a financial module of an ERP system (ERP FI) is a prerequisite for the activities 'create payment order' and 'execute payment'.

In addition, some business activities and IT components are affected by compliance requirements. In the example, the requirement 'legal obligation to keep records' obliges German merchants to do accounting. According to IDW RS FAIT 1, this requirement is also related to the proper operation of IT that supports accounting activities (The Institut der Wirtschaftsprüfer in Deutschland e.V. [Institute of Public Auditors in Germany, Incorporated Association], 2002a). In this particular case, the compliance requirements 'physical access' and 'logical access' are prerequisites of 'legal obligation to keep records'. The compliance requirement 'physical access' requires a regulated access to physical IT components while 'logical access' requires the identification and authentication of the users of an application (The Institut der Wirtschaftsprüfer in Deutschland e.V. [Institute of Public Auditors in Germany, Incorporated Association], 2002a). For simplification purposes, the compliance requirement 'logical access' only places demands to the IT component 'ERP MM'. Furthermore, the compliance process 'check invoice' satisfies the requirement 'internal payments policy', which specifies additional requirements that are necessary for the payment of invoices.

2.4. Formal definitions

Formal graph theory is widely used to model interrelations between elements (Heckel & Taentzer, 2020). This is also applied to business process models (Weske, 2019), IT architecture models (Dreyfus & Iyer, 2006), and compliance requirements (Sillaber & Breu, 2012). Following the compliance meta-model presented in Section 2.2 and (Seyffarth et al., 2018), we introduce several definitions to formally define relationships between compliance requirements, business processes, compliance processes and IT components.

Definition 1 (Compliance Requirement Graph). A compliance requirement graph is a directed graph $CRG = (N^{CRG}, E^{CRG})$, where: N^{CRG} is a nonempty finite set of nodes representing compliance requirements and $E^{CRG} \subseteq N^{CRG} \times N^{CRG}$ is a set of directed edges between nodes. An edge $e_{i,j}^{CRG}$ is considered to be directed from n_i^{CRG} to n_j^{CRG}.

Definition 2 (Business Process Graph). A business process graph PG is a 3-tupel $PG = (N^{PG}, E^{PG}, c\_type)$, where: $N^{PG} = BA \cup CA \cup C$ is a set of nodes in PG that follow common execution semantics. BA is a set of business activities and CA is a set of compliance activities, where: $BA \cap CA = \emptyset$. C is a set of coordinating nodes and $E^{PG} \subseteq N^{PG} \times N^{PG}$ is a set of directed edges between nodes representing a control flow such that (N^{PG}, E^{PG}) is a connected process graph. An edge $e_{i,j}^{PG}$ is considered to be directed from n_i^{PG} to n_j^{PG}. The function $c\_type : C \rightarrow \{start, end, intermediate, split, synchronize, choice, merge\}$ assigns a coordinator type to each coordinating node of PG.

Definition 3 (Compliance Process Graph). A compliance process graph is a subgraph of PG and a 3-tupel $CP = (N^{CP}, E^{CP}, c\_type)$ if $N^{CP} \subseteq CA \cup C$ and $E^{CP} \subseteq E^{PG}$.

Definition 4 (IT Architecture Graph). We define an IT architecture as a directed graph $ITG = (N^{ITG}, E^{ITG})$ where N^{ITG} is a nonempty finite set of nodes representing IT components and $E^{ITG} \subseteq N^{ITG} \times N^{ITG}$ is a set of directed edges between nodes. An edge $e_{i,j}^{ITG}$ is considered to be directed from n_i^{ITG} to n_j^{ITG}.

Definition 5 (Integrated Compliance Graph). In order to integrate the elements of the aforementioned models into one model, we define an integrated graph, which includes compliance requirements, a business process with integrated compliance processes and IT components (Seyffarth et al., 2018). The connection of elements of the different graph types is based on the set of directed edges E^{con}. An edge $e_{i,j}^{con}$ is considered to be directed from n_i^{PG} to n_j^{ITG}, n_i^{PG} to n_j^{CRG}, n_i^{ITG} to n_j^{PG}, n_i^{CRG} to n_j^{PG}, or n_i^{CRG} to n_j^{ITG}. In consequence, a directed integrated compliance graph G is a 4-tupel $G = (N^G, E^G, n\_identify, n\_type)$, where: $N^G = N^{CRG} \cup N^{PG} \cup N^{ITG}$ is a nonempty finite set of nodes, $E^G = E^{CRG} \cup E^{PG} \cup E^{ITG} \cup E^{con}$ is a set of directed edges between nodes with $E^G \subseteq N^G \times N^G$. The function $n\_identify(n_i^G)$ assigns an unique identificator (id) to each node n_i^G. The node type of n_i^G is specified by the function

$$n\_type(n_i^G) = \begin{cases} BusinessActivity : n_i^G \in BA \\ ComplianceActivity : n_i^G \in CA \\ CoordinatingNode : n_i^G \in C \\ ITComponent : n_i^G \in ITG \\ ComplianceRequirement : n_i^G \in CRG \end{cases}$$

For a practical implementation of the generation of the integrated compliance graph *G*, the process model, the IT architecture model and the compliance model must be available in machine-readable form (Seyffarth et al., 2018; Seyffarth & Raschke, 2018). Given that processes are available, e.g. in BPMN 2.0, IT architectures, e.g. in TOGAF Open Exchange, and compliance requirements, e.g. XML-based legal texts, the models are automatically parsed and converted into corresponding nodes and edges in a first step. To realise an instance of the integrated graph *G*, connections between the different model types are defined manually in a second step.

3. Related work

3.1. Method

We conducted two structured literature reviews (Sackmann et al., 2018; Seyffarth et al., 2017b) in order to analyse related work. We followed the method proposed by (Vom Brocke et al., 2009). Our first review focused on a broad search for approaches from the field of business process management dealing with compliance in order to get an overall impression of the research field. Our second review focused on an in-depth search for approaches from the field of business process compliance dealing with changes in processes, compliance requirements and IT components. In the following, we highlight the main ideas of relevant related approaches, which we categorised according to our research questions.

3.2. Analysing the interactions

There are approaches that queries the relations between compliance requirements and business processes, compliance requirements and IT components and interrelated requirements. We also consider query languages against the process, as they can, for example, also consider documents in the process model. However, none of these distinguishes different change patterns.

Many approaches query the relations within business process models based on pattern matching. (Awad, 2007) presents an approach for querying specific patterns on process graphs that are modelled using BPMN. Both (Delfmann et al., 2015) and (Gacitua-Decar & Pahl, 2009) use graph searching techniques to realise pattern matching in in business process models. In addition, the approach of (Delfmann et al., 2015) is a multi-model approach, which is able to consider different model types, such as documents, organisational units, or IT components. (Gacitua-Decar & Pahl, 2009) also consider semantical aspects. (Fellmann et al., 2011) transform business process models into ontologies and enrich them with structural and domain representation information. However, approaches that are based on pattern matching always require an apriori known pattern query (Delfmann et al., 2015). Although these approaches are in principle applicable in the given application context, they are limited to a specific set of known patterns. However, the practice is characterised by new and constantly changing compliance requirements. Therefore, it is of particular importance that approaches to ensure compliance in business processes are not exclusively limited to known patterns.

Several authors, such as (Ghanavati et al., 2009) and (Corea & Delfmann, 2017), propose approaches for linking business processes and compliance requirements in formal models. Beyond that, (Rudzajs & Buksa, 2011) developed an approach that identifies business activities which are affected by changes in compliance requirements. This approach is based on a version control system for the compliance requirements. Both (Knuplesch et al., 2015) and (Fdhila et al., 2015) discuss the impacts of changed compliance requirements on business processes in a cross-organisational context.

Moreover, a few authors discuss the interrelations between compliance requirements and IT components on a conceptual level (Becker et al., 2011; Knackstedt et al., 2013). (Becker et al., 2014) present an approach for formally modelling and querying interrelations between IT components and single compliance requirements.

Only a few authors address the link between different compliance requirements. Independently of each other, both (Elgammal et al., 2010) and (Halle, 2011) present approaches for detecting root cause violations in rules that are formalised in linear temporal logic.

(Koetter et al., 2014) propose a so called 'compliance descriptor' to model interrelations between compliance requirements, business processes and IT components; however, their approach cannot link compliance requirements. This work is extended in (Koetter et al., 2016) with a graphical modelling notation for the compliance descriptor. The approach of Koetter et al. shows a conceptual relationship to our work. However, due to the inability to map connections between interdependent requirements and the focus of modelling dependencies rather than graph search algorithms, we could not use their approach as a starting point for developing our decision support system for maintaining compliance despite changes.

3.3. Business process adaption

The literature discusses two different types of approaches for adapting business processes automatically to ensure compliance in case changes are made to process activities or process flows.

On the one hand, there are approaches aimed at ensuring compliance within business processes by removing or reordering flow elements, such as gateways or activities, e.g. (Awad et al., 2009; Elgammal et al., 2012). On the other hand, there are approaches aimed at modelling and storing compliant process fragments separately from the business process model. Each compliant process fragment serves to fulfil a compliance requirement. The fragments can be integrated into the business process either at design time or during run time (Sackmann & Kittel, 2015; Schumm et al., 2010a). The separate modelling and storage of fragments offers the possibility for reuse in different business process models. However, none of these approaches considers explicitly the modelling of alternative compliance processes in consideration of IT components.

3.4. Research gap

As depicted in Table 1, we summarised the related work in an author centric concept in accordance with (Webster & Watson, 2002). On the one hand, there is a lack of a method that allows for automatically determining compliance requirements and compliance

Table 1. Author centric concept matrix.

	Determine compliance requirements and compliance violations					Generate compliant business processes	
	BP	BP-CR	CR-IT	CR-CR	BP-IT-CR	Reorder	Separate
(Awad, 2007)	X						
(Gacitua-Decar & Pahl, 2009)	X	(X)					
(Delfmann et al., 2015)	X	(X)			(X)		
(Fellmann et al., 2011)	X	(X)					
(Ghanavati et al., 2009)		X					
(Corea & Delfmann, 2017)		X					
(Rudzajs & Buksa, 2011)		X					
(Fdhila et al., 2015)		X					
(Knuplesch et al., 2015)		X					
(Becker et al., 2011)			X				
(Becker et al., 2014)			X				
(Knackstedt et al., 2013)			X				
(Elgammal et al., 2010)				X			
(Halle, 2011)				X			
(Koetter et al., 2014)					X		
(Koetter et al., 2016)					X		
(Awad et al., 2009)						X	
(Elgammal et al., 2012)						X	
(Schumm et al., 2010a)							X
(Sackmann & Kittel, 2015)							X
Our proposed method	X	X	X	X	X	(X)	X
Legend	• BP: Query within Business Process • BP-CR: Query relations between Business Process and Compliance Requirement • CR-IT: Query relations between compliance requirement and IT component • CR-CR: Query relations between different compliance requirements • BP-IT-CR: Business Process, Compliance Requirement and IT component • Reorder: reorder process elements to ensure compliance • Separate: separate modelling and storing of business processes and compliant business fragments • (X): Can be done implicitly						

violations when considering the change patterns of delete and replace. This is particularly true if the underlying model includes business processes, IT components, and compliance requirements, which in turn can be connected to each other. On the other hand, existing methods for a business process adaption do not make a process adjustment based on the previous analysis, and they do not consider prerequisite IT components to execute compliant process fragments.

4. Research method

We applied a design science research approach inspired by the method described in (Peffers et al., 2006) to structure our procedure and ensure scientific rigour. Accordingly, we conducted six process steps to implement our research project (Peffers et al., 2006) as shown in Figure 3: (1) identify problem and motivate; (2) define objectives of a solution; (3) design and development; (4) demonstration; (5) evaluation; and (6) communication.

We previously addressed the problem of identification and motivation (step 1) in the introduction (c.f. Section 1) and explained it using an illustrative example (cf. Section 2.3). To the best of our knowledge, there are currently no solutions available both for the automatic identification of compliance requirements and compliance violations after the
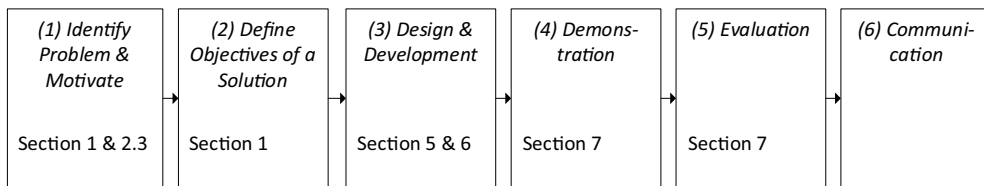
(1) Identify Problem & Motivate	(2) Define Objectives of a Solution	(3) Design & Development	(4) Demons-tration	(5) Evaluation	(6) Communi-cation
Section 1 & 2.3	Section 1	Section 5 & 6	Section 7	Section 7	

Figure 3. Our design science research approach based on (Peffers et al., 2006).

replacement or deletion of business activities, IT components, and/or compliance require-ments, and for the automatic generation of proposals of compliant business processes for curing compliance violations. Therefore, as already outlined in the introduction (cf. Section 1), our research objective (step 2) is a decision-support approach to recommend compliant business processes. Consequently, we developed two artefacts for problem solving (step 3). First, we developed a method to analyse the impact of both replacements and deletions of model elements on compliance (cf. Section 5). Second, we developed a method to recommend compliant business processes in response to compliance viola-tions resulting from replacements or deletions of model elements (cf. Section 6).

To verify the feasibility of our artefacts, we demonstrated them with the software prototype called 'BCIT' (step 4; c.f. Section 7). The demonstration of artefacts by means of a prototype is already considered as an evaluation in DSR (Hevner et al., 2004). However, an essential objective of evaluating design studies is to investigate the usability of artefacts (Hevner et al., 2004). Accordingly, we evaluated the perceived usefulness of our solution artefacts in Section 7 by means of an expert survey (step 5). Finally, the communication of our completed research project (step 6) occurs in this research paper.

5. Analyse the interactions between compliance and change

In this section, we address RQ1 by presenting a method to identify compliance require-ments when replacing an element as well as compliance violations when removing an element (Seyffarth et al., 2018). In the following, any arbitrary nodes of the integrated compliance graph will be referred to as 'elements'.

5.1. Business process change patterns

Various business process change patterns have been discussed in the literature. For example, (Weber et al., 2008) distinguished 18 change patterns and split them into 'adaption patterns' and 'patterns of change in predefined regions of business processes'. Further authors have combined the patterns into four change patterns: 'insert element', 'delete element', 'replace element', and 'update element' (Fdhila et al., 2012; Rinderle-Ma et al., 2008). The 'insert pattern' inserts a new element into a business process at a defined place. The 'delete pattern' removes an existing element. The 'replace pattern' replaces an existing element with a new one. The 'update pattern' modifies an attribute of an existing

element (Fdhila et al., 2012). These change patterns can be applied to all kinds of elements in the integrated compliance graph.

As outlined in the introduction section, we focus this study on the impact of the change patterns 'replace element' and 'delete element', since their resulting influences on compliance can be derived directly from information of the integrated compliance graph *G*. Of course, the mentioned change patterns can also have an impact on business activities and/or IT components. However, we do not take these into account, as they are not part of a focused compliance investigation, but rather of a business impact analysis e.g. (Radeschütz et al., 2015).

The analyses differ depending on the change pattern applied. In the case of a replacement, existing compliance requirements have an effect on the changed element. In the case of a deletion, the change to process-/IT-elements has an effect on the relevance of existing compliance requirements. Consequently, the change patterns 'replace' and 'delete' lead to different cause and effect relationships. Figure 4 refers to
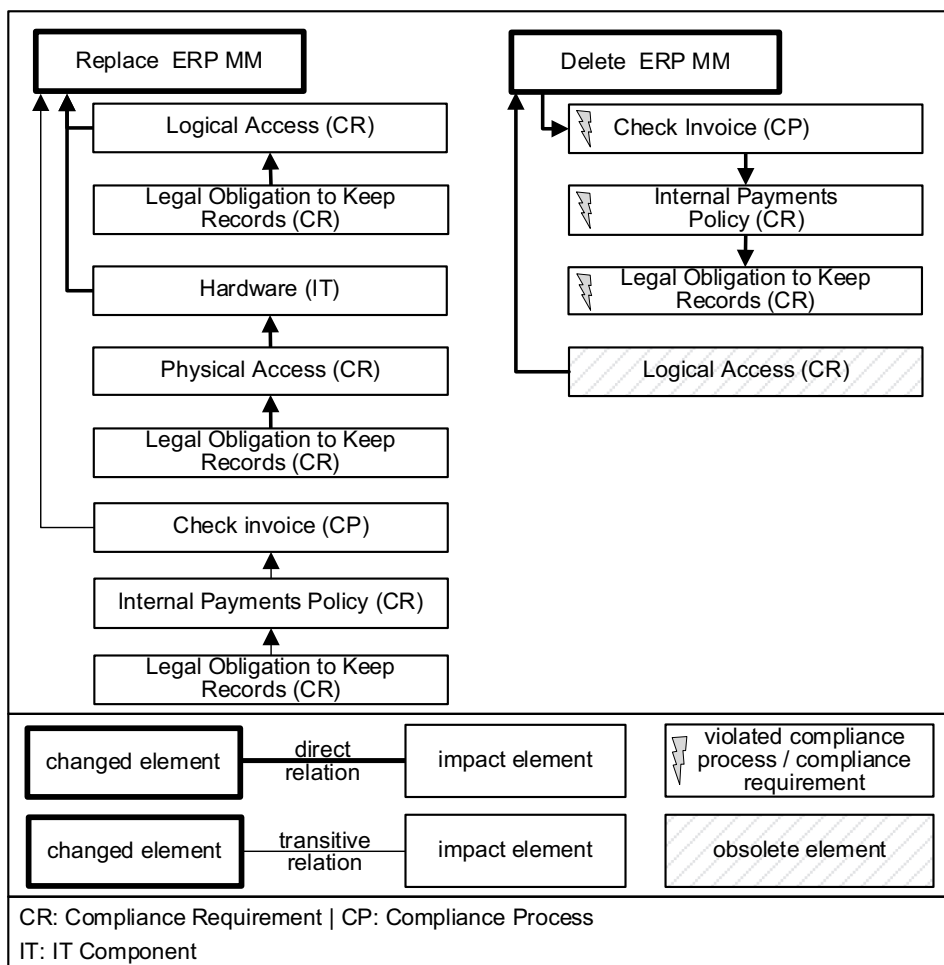

Figure 4. Compliance requirements when replacing andcompliance violations when deleting 'ERP MM'.

our example model and shows the compliance demands when replacing the IT component 'ERP MM'. Further, it shows compliance violations and obsolete compliance requirements when removing 'ERP MM'.

5.2. Replacing an element

In order to satisfy related compliance requirements, the replaced element must fulfil the related compliance requirements and if necessary be able to execute a related compliance process. In this case, compliance requirements as well as compliance processes may place conditions on elements, which in turn must be taken into consideration by the element replacing the old one. There are direct and transitive relations between a compliance requirement and the replaced element.

A direct relation between compliance requirements and replaced IT components occurs in two different ways. First, it is necessary to consider any compliance requirement that is directly related to the replaced element. Within our example, model the compliance requirement 'logical access' is directly related to the replaced IT component 'ERP MM'. Consequently, the compliance requirement 'legal obligation to keep records' is also directly related, since it is a prerequisite of 'logical access'. Second, all compliance requirements of the replaced element prerequisites must also be taken into consideration. Within the motivation scenario, the compliance requirement 'physical access' is directly related to the replaced element 'ERP MM' because it is a prerequisite for the IT component 'hardware', which is, in turn a prerequisite for 'ERP MM'.

A transitive relation between an updated IT component and a compliance requirement (or respectively a compliance process) also occurs in two different ways. First, the changed IT component can be a prerequisite of a business activity that is affected by at least one compliance requirement. Within the example model, this is the case at the compliance process 'check invoice' whose prerequisites are the compliance requirements 'internal payments policy' and 'legal obligation to keep records'. Second, the replaced IT component is a prerequisite of a compliance process that satisfies at least one compliance requirement.

Basically, the result when replacing an IT component (as shown in Figure 4) is automatically generated by a forward and backward search. The search is performed on the integrated compliance graph G and takes the different node types into account. Since we defined all possible interrelations between the different node types, the identification of demanding compliance requirements is based on fixed search strategies, which depend on the type of changed node. Figure 5 shows the corresponding algorithm to determine compliance requirements when replacing an IT component.

The algorithm in Figure 5 can be described as follows. First, all compliance requirements directly related to the replacement of an IT component are identified and all previous IT components are determined. Second, all compliance requirements that are directly related to the previous IT components are identified as well. Third, to identify all transitively related compliance requirements, we define all business activities and compliance processes supported by the replaced IT component. This also includes IT components that are a prerequisite for IT components that in turn directly support business activities and compliance processes.

```
Input: Graph g, element that shall be replaced {n ∈ g | n_type(n) = IT component}
1  // get all direct related compliance requirements and compliance processes to n
2  preds_n = get_all_predecessor(n, type(pred_n)=compliance requirement OR compliance process)
3  Foreach i in (preds_n) do
4  └ mark i as direct AND add i including all vertices between i und n to result
5  // get all transitive related compliance processes and compliance requirements to n
6  Foreach it in (get_all_leafs(n, type(leaf_n)=it component)) do
7   Foreach activity in (get_all_direct_successor(it, type(suc_it)= business process)do
8    Foreach cr in (get_all_direct_predecessor(activity, type(pred_activity)=compliance requirement) do
9     i = get_all_predecessor(cr)
10    └ mark it, activity, cr and i as transitiv AND add to result
11   Foreach complianceprocess in (get_all_direct_successor(it, type(dir_suc)=compliance process) do
12    Foreach cr in (get_all_direct_successor(complianceprocess, type(cp)=compliance requirement) do
13     i = get_all_predecessor(cr)
14    └ └ mark it, complianceprocess, cr and i as transitiv AND add to result
   generate result_graph based on g and result
Output: Graph result_graph
```

Figure 5. Algorithm to automatically determine compliance requirements when replacing an IT component.

5.3. Deleting an element

The removal of an element can affect compliance in two different ways. On the one hand, any deletion can impact compliance requirements and/or compliance processes that are related to the deleted element. On the other hand, the deletion of an element can result in an affected compliance requirement becoming obsolete.

In our motivating example, the removal of the IT component 'ERP MM' leads to the non-executability of the compliance process 'check invoice'. Consequently, the compliance requirements 'internal payments policy' and 'legal obligation to keep records' are violated. In addition, the removal makes the compliance requirement 'logical access' obsolete because it only affects the d IT component 'ERP MM'. However, the superior compliance requirement 'legal obligation to keep records' does not become obsolete, as it is still the prerequisite of the compliance requirements 'physical access' and 'internal policy'.

As with the replacement method, the method for automatically identifying the effects of the element deletion on compliance is based on a backward and forward search in graph *G*. The search strategy depends on the deleted element type. Figure 6 shows the corresponding algorithm.

The algorithm in Figure 6 can be described as follows: First, to determine compliance violations when removing an IT component, we identify all compliance processes supported by the component. This includes both an implicit and explicit support. Second, we determine all compliance requirements that are satisfied by the supported compliance process. Third, to obtain all obsolete compliance requirements when an IT component is deleted, three major steps are done.

In step 1, we determine all succeeding IT components of the deleted IT component. In step 2, we determine all previous IT components of the deleted IT component until none of these predecessors support a component that is not an implicit predecessor of the deleted one. In step 3, for each IT component identified and the deleted IT component, we determine all related compliance requirements including all of their associated child elements. We check systematically whether each of the identified compliance requirements also affect an element other than the deleted IT component and the IT components identified in steps 1 and 2. If this is not the case, we mark them as obsolete.

```
Input: Graph g, element that shall be removed {n ∈ g | n_type(n) = IT component}
1  // get all violated compliance requirements
2  Foreach it in (get_all_leafs(n, type(leaf(n))=it component) do
3    Foreach cp in (get_all_direct_successor(it, type(suc(it))=compliance process) do
4      Foreach cr in (get_all_direct_successor(cp, type(suc(cp))=compliance requirement) do
5        i = get_all_predecessor(cr)
6        mark cr and i as violated AND add to result
7  // get all obsolete compliance requirements
8  add n to list_it
9  Foreach it in (get_all_successor(n, type(suc(n))=it component) do
10   If (it only requires n)
11     add it to list_it
12 Foreach it in (get_all_predecessor(n, type(pred(n))=it component) do
13   If (it only supports n)
14     add it to list_it
15 Foreach it in (list_it) do
16   Foreach cr in (get_all_direct_predecessor(it, type(pred(it))=compliance requirement) do
17     If (get_all_direct_successor(cr)==it)
18       mark cr as obsolte AND add to result
19       Foreach cr2 in (get _all_predecessor(cr), type(pred(cr))=compliance requirement) do
20         If (cr2 hasn't other direct successor than get_all_predecessor(cr, type(cr)=comp.req.)
21           mark cr2 as obsolte AND add to result
22 generate result_graph based on g and result
Output: Graph result_graph
```

Figure 6. Algorithm to automatically determine compliance violations and obsolete compliance requirements when deleting an IT component.

6. Propose compliant business process models

In this section, we present a method that recommends compliant business process models at design time in response to compliance violations (Seyffarth et al., 2019). The compliance violations are caused by replacement or deletion at design time. The method automatically generates adaptation proposals for non-compliant business processes and thus addresses RQ2. First, we present a running scenario. Second, we present a method to recommend compliant business processes and thus operationalise the running scenario.

6.1. Running scenario

As a continuation of our example model, Figure 7 shows different variants of a purchase to pay process that complies with the requirement 'internal payments policy'. The compliance processes shown in Figure 7 are, of course, not complete, but can easily be extended if new ideas or opportunities arise. The alternative compliance processes shown on the left side lead to different compliant business processes on the right side. The compliance processes included in the purchase to pay processes differ according to their triggers and execution requirements.

6.2. Modelling alternative compliance processes

In general, the modelling of alternative compliance processes during design time of the business process model (e.g. (Kittel et al., 2013; Sackmann & Kittel, 2015; Schumm et al., 2010b)) is based on two ideas:

- The differentiation of alternative compliance processes based on their properties, and
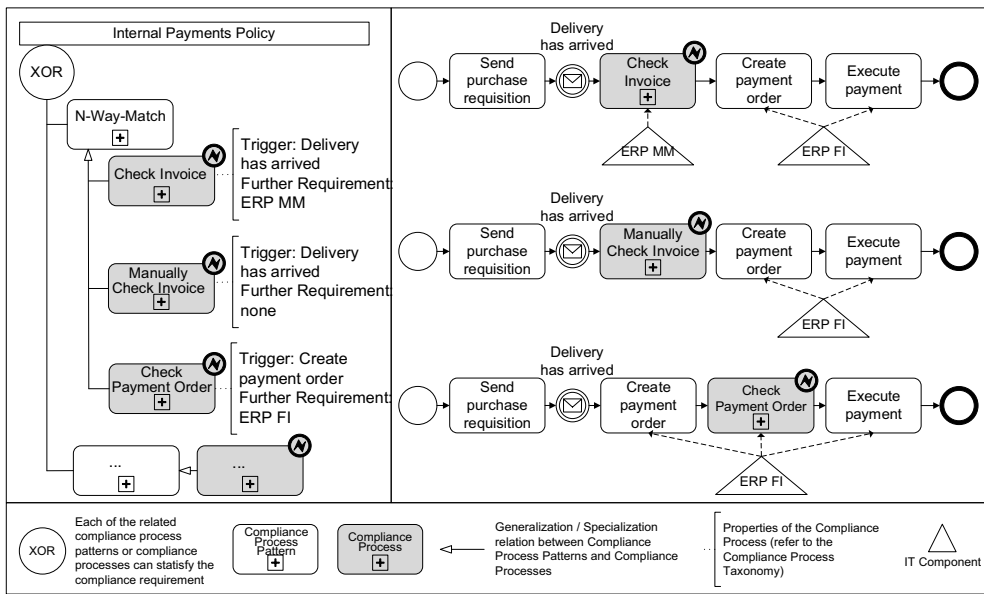
Figure 7. Alternative compliance processes and proposed compliantpurchase to pay processes (based on (Seyffarth et al., 2019)).

- The generalisation of alternative compliance processes in one so-called compliance process pattern.

6.2.1. Properties of a compliance process

Each compliance process may have different properties, such as different triggers, execution requirements or execution types. In (Seyffarth et al., 2017a) we proposed a compliance process taxonomy that enables the categorisation of compliance processes based on their characteristics. Thus, it is possible to differentiate (alternative) compliance processes by means of 3 meta-characteristics and 37 sub-characteristics in 9 dimensions. Even though the taxonomy allows for a theoretical differentiation of compliance processes, their specific modelling and instantiation always depend on the conditions of the respective company. For example, the content and number of compliance requirements, or the degree of automation determine the type and number of available alternative compliance processes.

In the following, we present several characteristics and dimensions whose understanding is necessary for the further course of argumentation. A complete discussion of all characteristics and dimensions can be found in (Seyffarth et al., 2017a). The meta-characteristic 'integration constraint' specifies requirements for the integration of a compliance process into a business process. One dimension within these meta-characteristic is the dimension 'trigger'. A trigger initiates the integration of a compliance process into a business process to meet a compliance requirement.

Another dimension of 'integration constraint' is 'further requirements for execution', which specifies the dependency of process execution on various characteristics, such as the existence of an IT component. Besides other features, the meta-characteristic

'modelling' includes patterns for modelling compliance processes. The meta-characteristic 'property' allows for differentiation compliance processes based on their timing, degree of automation or process type. These properties may depend on other dimensions and characteristics of the compliance process taxonomy, such as the type of execution, which can be either fully automated, fully manual or manual, but IT dependent.

6.2.2. *Compliance process patterns as a generalisation of compliance processes*

As a basis for modelling alternative compliance processes we use the compliance process patterns proposed in (Namiri & Stojanovic, 2007; Schultz, 2013). In general, patterns are high-level domain-specific templates used to represent desired properties and constraints (Turetken et al., 2011). Consequently, we define a compliance process pattern as a process template that contains process elements (e.g. activities, gateways and connectors) that are necessary to satisfy at least one compliance requirement. Examples of such compliance process patterns includes the 'second set of eyes' pattern, the 'separation of duties' pattern or document patterns such as an 'N-way-match' that compares different text values.

On a general level, the compliance requirement 'internal payments policy' from Figure 7 can be satisfied by a compliance process pattern of type 'N-way-match'. In our example, the level of detail of this pattern is increased by specifying three corresponding compliance processes, which differ in their characteristics. Consequently, the three alternative compliance processes 'check invoice', 'manually check invoice' and 'check payment order' in Figure 7 can each be used to ensure the requirement 'internal payments policy' and represent specialisations of the compliance process pattern 'N-way-match'.

6.3. *Query compliance processes and integrate them into the business process model*

For further explanations, we first define a data structure that contains compliance requirements, compliance process patterns and associated compliance processes. The data structure can be used to represent the alternative compliance processes shown in Figure 7. Second, we describe a method that (1) queries suitable compliance processes from these data structures and (2) integrates them into the business process model.

Definition 6 (Alternative Compliance Process Graph). We define a directed graph $AG = \left(N^{AG}, E^{AG}, n^{AG}_{identify}, n^{AG}_{type}, n^{AG}_{rule}, \right)$. In addition, N^{AG} is a nonempty finite set of nodes and an edge $e^{AG}_{i,j}$ is considered to be directed from n^{AG}_i to n^{AG}_j.

The node type of n^{AG}_i is specified by the function $n^{AG}_{type}\left(n^{AG}_i \right)$ and represents the node types that can occur in the alternative compliance process graph.

$$n^{AG}_{type}\left(n^{AG}_i \right) = \begin{cases} Compliancerequirement(CR) \\ Complainceprocesspattern(CPP) \\ Complianceprocess(CP) \end{cases}$$

In accordance with Definitions 1–5, the nodes of AG can be of the type of compliance requirement (CR), compliance process pattern (CPP), or compliance process (CP). To

ensure adherence to the data structure of the *AG* (as shown in Figure 7), we restrict the allowed edges between the defined node types as follows

$$E^{AG} = \left\{ \left(\left(n_i^{AG} | n_{type}^{AG} (n_i^{AG}) = CPP \right), \left(n_j^{AG} | n_{type}^{AG} (n_i^{AG}) = CR \right) \right); \right.$$
$$\left(\left(n_k^{AG} | n_{type}^{AG} (n_k^{AG}) = CP \right), \left(n_l^{AG} | n_{type}^{AG} e (n_l^{AG}) = CR \right) \right);$$
$$\left(\left(n_m^{AG} | n_{type}^{AG} (n_m^{AG}) = CPP \right), \left(n_n^{AG} | n_{type}^{AG} (n_n^{AG}) = CPP \right) \right);$$
$$\left. \left(\left(n_o^{AG} | n_{type}^{AG} (n_o^{AG}) = CP \right), \left(n_p^{AG} | n_{type}^{AG} \left(n_p^{AG} \right) = CPP \right) \right) \right\}$$

Accordingly, only edges between CR and CPP, CR and CP, CP and CPP, as well as CPP and CPP are allowed.

The function $n_{identify}^{AG} (n_i^{AG})$ assigns a unique identification (ID) to each node n_i^{AG}, the function $n_{rule}^{AG} (n_i^{AG})$ assigns a rule to each node n_i^{AG}. Each rule includes a trigger, a business activity prior to which the compliance process was executed and required IT components. Each element of the rule must be named after the name of the corresponding business activities and IT components. We are aware that this can lead to naming ambiguities. In the literature there are already approaches to tackle this challenge in business processes (e.g. (Klinkmüller & Weber, 2017; Leopold et al., 2015)). However, a first simple solution to this challenge is a tool support, which specifies the appropriate elements based on existing process models and IT architecture models.

As Figure 8 shows, the query of alternative compliance processes starts at n_i^{AG}, which represents the compliance process that is no longer executable. Within *AG*, we first search for alternative compliance processes. These compliance processes are modelled as sibling nodes of n_i^{AG} where $n_{type}^{AG} = CP$. Next, the rule or integration constraints of each sibling node must be checked against the business process to investigate if the sibling nodes found can be integrated into the business process *PG*. In our case, the integration constraints include the presence or absence of both business activities and IT components. Hence, we consider the control flow and resource perspective of a process.

The method for querying alternative compliance processes is based on three steps. In order to check the executability of each alternative compliance processes n_i^{AG} in *AG*, we

```
Input: alternative graph ag, result graph rg, process graph pg
1  // get all violated compliance processes
2  Foreach cp in (get all violated compliance processes in rg) do
3    // get all alternative compliance processes
4    Foreach sibling_cp in (get sibling compliance processes of cp) do
5      // check the executability of alternative compliance process
6      req_it = silbing_cp.getRequiredIT
7      trigger = silbing_cp.getTrigger
8      If (req_it OR req_it.predecessor not in rg) AND (trigger not in rg)
9        add sibling_cp to alternatives
10     // search for compliance process patterns
11     If (alternatives is empty)
12       add get_all_succesor(cp, type(suc_cp)=compliance process pattern) to alternatives
13   // integrate alternative cp into the business process graph pg
14   Foreach element in alternatives do
15     bp = pg
16     removeViolatedComplianceProcess from bp
17     integrate element in bp
18     add bp to compliantBusinessProcesses
Output: list process_graph compliantBusinessProcesses
```

Figure 8. Algorithm to propose compliant business process models.

first check their integration constraints against the result graph. As discussed in Section 5.3 the result graph, is calculated when an element is deleted. It contains the deleted element, the violated compliance requirements or compliance processes and obsolete compliance requirements. In the second step, a compliance process is classified as executable if the triggering business activity is not deleted. It is further executable if the required IT component is not deleted or if the required IT component is not a predecessor of the deleted IT component. To check these conditions, in the third step we look at whether the trigger or the required IT component are part of the result graph. If this is the case for at least one of the conditions, the compliance process cannot be integrated into the business process.

If there is no appropriate compliance process, we perform a second search, one for appropriate compliance process patterns. Accordingly, we search for successors of n_i^{AG} that are of the type 'compliance process pattern'. Finally, each query can lead to three possible results: one alternative compliance process, more than one alternative compliance process or no alternative compliance process.

We propose a solution for each case. In the case of one alternative compliance process, exactly one adapted business process is proposed. If there are numerous alternative compliance processes, our method queries all alternative compliance processes and thus proposes more than one adapted business process, as is the case in our example. Subsequently, the user decides which process adaption should be made. If no suitable compliance process is available, our method proposes a generic compliance process pattern. In this case, the compliance pattern is integrated in place of the former compliance process. Subsequently, the compliance process pattern must be specified syntactically and semantically for practical use.

7. Evaluation

The goal of DSR is to solve identified organisational problems. Since the evaluation of design-oriented studies serves to assess the usefulness of solution artefacts (March & Smith, 1995), it is essential to evaluate the usefulness of our solutions (Hevner et al., 2004). Consequently, we evaluated the perceived usefulness of our decision support approach with domain experts. For such an expert evaluation, knowledge of the requirements that are placed on a method to assess the perceived usefulness is very important. We demonstrated the application of our solution artefacts with a software prototype. In addition to our algorithms introduced in Section 6, the demonstration also takes into account the necessary requirements, such as the modelling of alternative compliance processes.

Section 7 is divided into two logical parts. First, we apply the Framework for Evaluation in Design Science Research (FEDS) (Venable et al., 2016) to specify appropriate evaluation episodes of our DSR project. Second, we present a case study based on which domain experts evaluate the perceived usefulness of our approaches.

7.1. Evaluation strategy based on FEDS

Basically, FEDS (Venable et al., 2016) consists of three dimensions: (1) functional purpose of evaluation, (2) paradigm of the evaluation study and (3) evaluation strategies. (Venable

et al., 2016) recommend the evaluation strategy 'technical risk and efficacy', if an evaluation with real users and real systems in a real setting is only possible with tremendous effort. Based on this strategy, we conducted four evaluation episodes in line with our research project, with the first three episodes being artificially formative. They are also preparative for the final evaluation in episode 4, which is naturalistically summative (see Table 2). Within the artificial evaluation paradigm, we applied the evaluations methods 'literature review' and 'prototyping'. Within the naturalistic evaluation we performed a case study based on a proof-of-concept implementation called 'BCIT'.

In evaluation episode 1 we derived both the elements of the compliance meta-model and the dimensions and characteristics of the compliance process taxonomy from the literature. Thus, both artefacts were artificially evaluated ex ante (Sonnenberg & Vom Brocke, 2012) and meet the ending conditions according to (Nickerson et al., 2013).

In evaluation episode 2, we demonstrated the feasibility of an integrated graph *G* which includes compliance requirements, business activities, compliance activities and IT components. Further, we demonstrated the identification of compliance demands and violations through a prototypical implementation of the corresponding algorithms. (March & Storey, 2008) referred to this as prototyping suitable evaluation methods for demonstrating the feasibility of artefacts. The result of this evaluation episode is the tentative proof-of-concept implementation 'BCIT' (Seyffarth & Raschke, 2018), which we presented at an international conference and discussed with peers of the BPM community.

In evaluation episode 3, we extended the tentative proof-of-concept of the previous evaluation episode. We added the functionality to provide adaption proposals for business processes following the identification of compliance violations (Seyffarth & Raschke, 2020). The prototype uses the adapted graph search algorithms introduced in Section 6. In general, graph search techniques are well known in computer science and their operation is widely proven (Aleliunas et al., 1979; Rosenkrantz et al., 1977). Thus, the aim is not to evaluate the formal correctness of the established graph search methods we use, but rather their usefulness for identifying compliance violations and alternative compliance processes in the context of the given problem.

Figure 9 shows the identified compliance violations when removing an IT component from the advanced prototype 'BCIT. A proposed compliant business process by 'BCIT' is shown in Figure 10. A tutorial document, a screencast and two demo projects for 'BCIT' are available on the following GitHub repository: https://github.com/tobiasseyffarth/bcit/.

In evaluation episode 4, we performed a naturalistic summative evaluation. We used the goal question metric approach (Basili, 1992) for formulating the evaluation goal: The aim of the evaluation is to determine the perceived usefulness and relevance (purpose) of our solution artefacts by means of the advanced proof-of-concept implementation BCIT from the perspective of domain experts. In order to reach this goal, we followed the recommendations of (Hevner et al., 2004; Sonnenberg & Vom Brocke, 2012) and conducted a case study with a subsequent expert survey.

7.2. Case study design and results

In this section, we briefly present the case study structure, the test design, our questionnaire and our sample. In order to document the data collection and the analysis in a comprehensible way (Foster & Deardorff, 2017), the questionnaire and

Table 2. Evaluation episodes in the research project.

| #	Functional Purpose	Paradigm of Evaluation Study	Focus of Evaluation	Applied Evaluation Method	Evaluation Result
1	Formative	Artificial	Compliance meta model, compliance process taxonomy	Ex ante evaluation (Literature review)	Meet ending conditions according to (Nickerson et al., 2013)
2	Formative	Artificial	Methods to create the analyse graph and compliance requirements/compliance violations	Prototyping and discussion with peers	Tentative proof-of-concept BCIT
3	Formative	Artificial/Naturalistic	Method to recommend compliant business processes	Prototyping and discussion with peers	Advanced proof-of-concept BCIT
4	Summative	Naturalistic	Extension of Proof-of-concept BCIT	Case study with domain experts	Perceived usefulness of BCIT
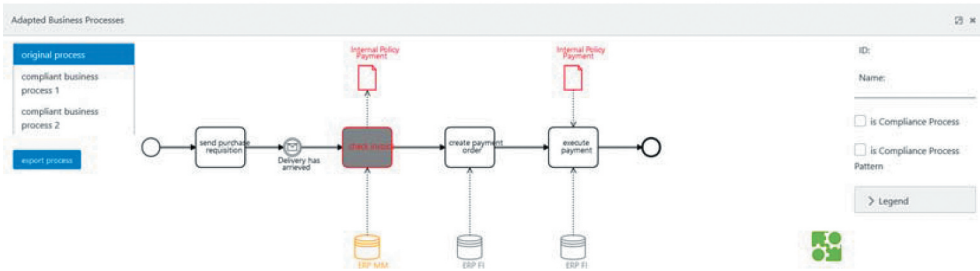
Figure 9. Violated compliance requirement when removing an IT component.
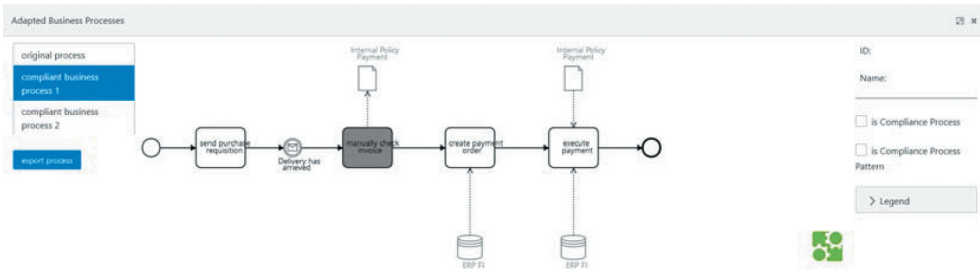

Figure 10. Compliant business process including thealternative compliance process 'manually check invoice'.

all answers are also available on our GitHub project page: https://github.com/tobias seyffarth/bcit/tree/master/resources/3-evaluation. Subsequently, we discuss the evaluation results and threats to validity. Inspired by (Runeson & Höst, 2009), we structured the case study in four parts:

(1) *Objective of the case study*. As previously stated, the goal of our case study is to evaluate the perceived usefulness and relevance of our proof-of-concept implementation of 'BCIT'.
(2) *Frame the knowledge and present the cases*. Based on a case study, we explained the (research) problem to the test persons and presented our basic ideas for its solution.
(3) *Demonstration of BCIT*. We exemplarily demonstrated BCIT's graph search algorithms and process adaptation procedures for solving the problem presented in the case study.
(4) *Collect data*. Following (Kriglstein et al., 2016; Runeson & Höst, 2009), we used questionnaires to obtain expert estimations of the perceived usefulness of BCIT.

7.2.1. Questionnaire and respondents

The construct of perceived usefulness is widely used for evaluating software artefacts and is part of various models, such as the Technology Acceptance Model (TAM), integrated models of TAM and the Theory of Planned Behaviour (TPB), and numerous extensions of these model types (Cheng, 2019; Hess et al., 2014). Science has already expressed strong

criticism of TAM. In the literature, e.g. the lack of contextual factors and external predictors is complained about (Marangunić & Granić, 2015), the importance of the construct Ease of Use is questioned (Yousafzai et al., 2007), and the general conception of the model is scrutinised (Cheng, 2019; Yousafzai et al., 2007). However, to the best of our knowledge and belief, the construct of perceived usefulness itself is not subject to criticism and is still frequently used for evaluating design-oriented research in general (see, e.g. (Sturm & Sunyaev, 2019), (Coenen et al., 2018; Santos & Alves, 2017)) and for decision support systems in particular (see, e.g. (Buchert et al., 2019; Kramer et al., 2017; Mican et al., 2020)). With regard to the objective of our case study, external predictors, contextual factors, and the construct Ease of Use are not relevant to our evaluation. Therefore, in accordance with our evaluation strategy according to (Venable et al., 2016) and their requirements for summative evaluations, we do not focus on TAM, TPB, or their extensions in our summative evaluation, but rather on specifically measuring the perceived usefulness of BCIT. The standard questionnaire for evaluating the perceived usefulness consists of ten statements (Davis, 1985). These statements are assigned to specific clusters which are related to the following usefulness aspects (Seeliger et al., 2019): (A) job effectiveness, (B) productivity and time savings, (C) importance of the system to the users' job, (D) control over the job and (E) overall assessment.

The statements were evaluated by experts on a Likert scale ranging from 'strongly disagree' (−3) to 'strongly agree' (3). It was also possible to comment on each statement or to reply with 'not specified' (abstention). In addition, the participants had the opportunity to make final comments on the questionnaire, the case study or BCIT in a blank comment field.

Before the actual case study took place, we pre-tested the questionnaire with four people having profound knowledge of research methods and the research field (three PhD students and one postdoctoral researcher). The pre-test gave an indication about the required time frame for the case study and the time to complete the questionnaire. Moreover, we corrected several ambiguities and minor mistakes in the questionnaire and created its final version.

We performed the case study 8 times with a total of 41 participants. The completion of the questionnaire was voluntary which led to 24 completed questionnaires (response rate: 58%). Table 3 contains the statements and the aggregated results including the arithmetic mean, standard deviation, median and the number of responses.

Additionally, we collected socio-demographic data on a voluntary basis, such as the job role or work experience. The majority of the participants worked in large companies and mainly in the financial and IT service sectors. A few worked in a large German research organisation or in small consulting companies. The participants worked in the field either of compliance (e.g. head of internal audit, chief compliance officer, IT risk manager and auditor), business process management (e.g. head of process management, process manager) or IT architecture management (e.g. IT architect, IT project manager). The working experience ranged from 4 to 32 years.

7.2.2. Discussion of the evaluation results

Figure 11 illustrates the distribution of the experts' voting results for the statements S1 to S10 using box plots. In addition, the box plots were categorised into the clusters A to E. We interpreted S10 and S6; all other box plots shown in Figure 11 are

Table 3. Statements and aggregated results of the perceived usefulness evaluation.

Statement (S)	Cluster	Mean	Standard Deviation	Median	Number of responses
S1: Using BCIT improves the quality of the work I do.	A	1.043	1.574	2	23
S6. Using BCIT improves my job performance.	A	0.455	1.437	1	22
S8. Using BCIT enhances my effectiveness on the job.	A	0.409	1.586	1	22
S3. BCIT enables me to accomplish tasks more quickly.	B	0.727	1.420	1	22
S5. Using BCIT increases my productivity.	B	0.429	1.466	0	21
S7. Using BCIT allows me to accomplish more work than would otherwise be possible.	B	0.609	1.635	1	23
S4. BCIT supports critical aspects of my job.	C	1.318	1.394	2	22
S9. Using BCIT makes it easier to do my job.	C	0.522	1.559	1	23
S2. Using BCIT gives me greater control over my work.	D	1.000	1.826	2	24
S10. Overall, I find BCIT useful in my job.	E	1.083	1.441	1.5	24

A: job effectiveness | B: productivity and time savings | C: Importance of the system to the users' job | D: control over the job | E: overall assessment

to be interpreted analogously. A look at the box plot of S10 (overall usefulness) shows that BCIT was generally evaluated positively. Even though the assessment of S10 covers the entire spectrum of possible answers, the interquartile range is still in the positive area of the Likert scale. Further, 75% of the participants agreed with statement S10. Looking at the results of S6 (job performance), outliers appear at both ends of the scale.

The majority of the participants liked the idea of modelling the relations between compliance requirements, business processes and IT components, because it provides the basis for an integrated model. They stated that the integrated model increases the transparency of their workflow as well as the associated technical and legal dependencies. This opens up new potentials for detailed root cause analyses, which can result in a competitive advantage. These comments also correspond to the results for S4 (support of critical aspects) and S2 (control over the job).

However, some participants pointed out that the effort for both modelling the business/compliance processes and the IT architecture was too high in comparison to the expected benefit. This is also a possible explanation for the poorer ratings on productivity (S5), performance (S6), and effectiveness (S8) compared to increasing quality (S1) and greater control (S2).

7.2.3. Threads to validity

Following (Runeson & Höst, 2009; Wohlin et al., 2012), we discuss several aspects of validity to acknowledge the limitations of our study. Construct validity denotes that the variables of interest are measured correctly. The construct we were interested in was the perceived usefulness. Since this construct has been used for several decades and has proven its worth in numerous studies, construct validity can be taken for granted.

The internal validity is of concern when causal relations are examined. Since we did not examine any causal relationships in our evaluation, the internal validity is of rather minor importance. Nevertheless, there is a risk that the variable being investigated may be affected by other neglected factors. To address this problem, we explained the case study in detail to all participants and made sure that there was a consistent knowledge base.
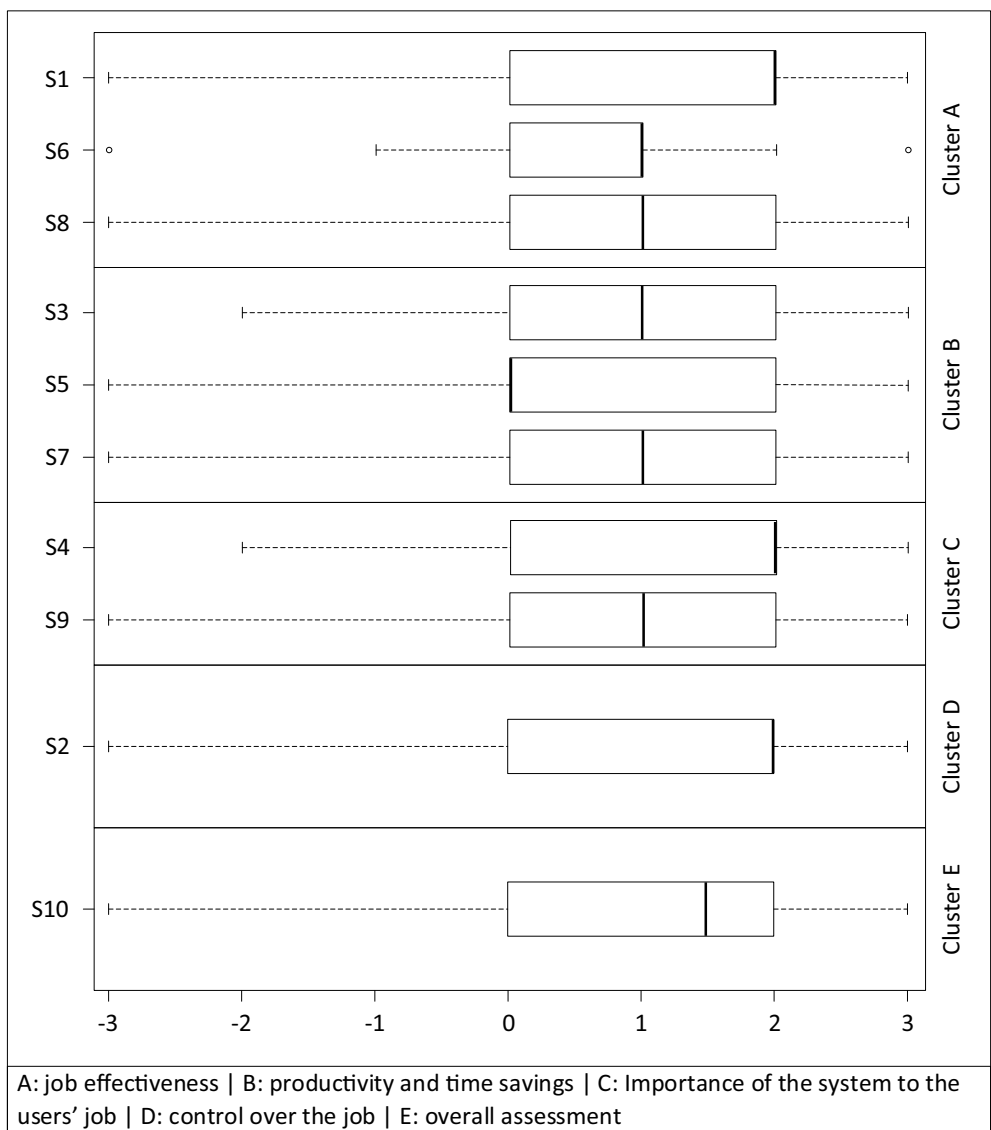
Figure 11. Overview of the perceived usefulness assessment of BCIT.

Furthermore, the skills of the participants involved in the experiments can be assumed to be appropriate due to their professional roles.

The external validity concerns the generalisability of our evaluation results. We interviewed participants with different positions and experiences from different companies and industries to address generalisability. Of course, the generalisability of the results could have been further improved by choosing a larger sample with additional participants from other companies and industries.

The conclusion validity is concerned with the relationship between the treatment and the outcome. We used the well-known questionnaire of (Davis, 1985) to measure

perceived usefulness with Likert scales in our evaluation for ensuring this relationship.

8. Implication for research and practice

In this paper, we presented methods for the identification of compliance requirements and compliance violations in case of replacing or removing an element. Further, we presented a method that recommends adaptions of business process models following compliance violations. We demonstrated the feasibility of our methods in the proof-of-concept implementation called BCIT. From these contributions, implications for research and practice can be derived.

For research, there are implications to the descriptive and prescriptive knowledge base (Gregor & Hevner, 2013). Contributions to the descriptive knowledge base are the identified research gap, and the compliance meta-model in order to conceptualise our domain. The contribution to the prescriptive knowledge base includes the methods to identify compliance violations and propose compliant business process models.

Despite the fact that compliance processes have to be modelled and instantiated individually in each organisation due to specific conditions, our solution artefacts have a general impact on practice. First, the explicit modelling of compliance requirements, business processes and IT components in an integrated model enhances the transparency of an enterprise architecture. The identified effects on compliance when changing business processes and IT components are therefore supported by factual evidence and can be used to support decisions. This support can also be further improved by allowing different scenarios to be tested quickly and cost-effectively. Second, a separate compliance view, which includes compliance processes, allows for thinking in alternatives. Just as business processes can be stored in a repository, alternative compliance processes can also be stored centrally and connected to their respective compliance requirements. Consequently, the repository can be used to identify a kind of best-practice compliance processes.

9. Conclusion

Compliance requirements can place demands on both business activities and IT components. In case of replacing and removing one of these elements, the demanding and violated compliance requirements have to be identified. If a business process violates a compliance requirement, an adaption needs to be initiated to re-establish compliance conformity. In order to solve this challenge, we presented three contributions. First, we presented methods for determining the interactions between requirements and the consequences on compliance when replacing or removing a business activity, an IT component or another compliance requirement. Second, we developed the proof-of-concept BCIT, which shows the feasibility of our methods. Third, we presented the results of a case study followed by an expert survey that confirmed the perceived usefulness of our methods and BCIT.

A prerequisite for the applicability of our method is the availability of corresponding information from existing design-time models, whose level of detail is company-specific. The availability of models, their levels of detail and the amount of information influences

how many solutions for compliance violations can be offered by our algorithm. The amount of information and, thus, the amount of available alternative compliance processes naturally increases over time. It is foreseeable that additional information will be gained through a long-term use of our approach/tool, thus extending its applicability. A central compliance process repository, similar to existing process fragment repositories, e.g. (Schumm et al., 2011), is a promising approach for the further development and joint use of compliance processes.

Further, naming ambiguities between the business process and the rule of the compliance process is currently an open topic. Currently, we bypass this challenge with the tool-support of BCIT.

We focused on the compliance change patterns 'replace' and 'delete', as their impact on compliance can be determined based on previously modelled dependencies between compliance requirements, business processes and IT components. An extension of the presented approach for further patterns is a research desideratum, as the impact of patterns such as 'insert a (possibly prohibited) new element' or 'update attributes of an element' on compliance cannot be derived exclusively by means of information from the underlying design-time models. Accordingly, further sources of information would have to be consulted to broaden our approach.

Besides considering further views on business processes, such as a data and organisational views, it is also possible to take into account additional parameters, such as effectiveness and costs, as a starting point for further research. Additional parameters allow for, e.g. the choice of alternative compliance processes regarding economic principles. This also enables making economic decisions on the migration to alternative compliant business process models. Beyond that, the idea of modelling alternative compliance processes can also be adopted to a modelling of alternative IT components, which may expand to a part of an alternative IT architecture. An alternative IT architecture can be used to react quickly to changes in the existing IT architecture.

Disclosure statement

References

Aleliunas, R., Karp, R.M., Lipton, R.J., Lovasz, L., & Rackoff, C. (1979). Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)* (pp. 218–223), San Juan, Puerto Rico, USA. https://doi.org/10.1109/SFCS.1979.34

Awad, A. (2007). BPMN-Q: A language to query business processes. In Reichert, M., Strecker, S. & Turowski, K. (Hrsg.), (Eds.), *Enterprise modelling and information systems architectures – concepts and applications* (pp. 115–128). Bonn: Gesellschaft für Informatik e. V.

Awad, A., Smirnov, S., & Weske, M. (2009). Resolution of compliance violation in business process models: A planning-based approach. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, R. Meersman, T. Dillon, & P. Herrero (Eds.), *On

the Move to Meaningful Internet Systems: OTM 2009 (Vol. 5870, pp. 6–23). Springer Berlin Heidelberg.

Basili, V. (1992). Software modeling and measurement: The Goal/Question/Metric Paradigm. University of Maryland, CS-TR-2956, UMIACS-TR-92-96, September 1992.

Becker, J., Delfmann, P., Dietrich, H.-A., Steinhorst, M., & Eggert, M. (2016). Business process compliance checking – Applying and evaluating a generic pattern matching approach for conceptual models in the financial sector. Information Systems Frontiers, 18(2), 359–405. https://doi.org/10.1007/s10796-014-9529-y

Becker, J., Eggert, M., Winkelmann, A., & Knackstedt, R. (2011). Towards a contingency theory based model of the influence of regulation on MIS. In Americas conference on information systems, Detroit, Michigan, USA.

Becker, J., Heddier, M., Braeuer, S., & Knackstedt, R. (2014). Integrating regulatory requirements into information systems design and implementation. In ICIS 2014 proceedings, Auckland, New Zealand.

Buchert, T., Ko, N., Graf, R., Vollmer, T., Alkhayat, M., Brandenburg, E., Stark, R., Klocke, F., Leistner, P., & Schleifenbaum, J.H. (2019). Increasing resource efficiency with an engineering decision support system for comparison of product design variants. Journal of Cleaner Production, 210, 1051–1062. https://doi.org/10.1016/j.jclepro.2018.11.104

Cheng, E.W.L. (2019). Choosing between the theory of planned behavior (TPB) and the technology acceptance model (TAM). Educational Technology Research and Development, 67(1), 21–37. https://doi.org/10.1007/s11423-018-9598-6

Coenen, T., Coertjens, L., Vlerick, P., Lesterhuis, M., Mortier, A.V., Donche, V., & Ballon, P. (2018). An information system design theory for the comparative judgement of competences. European Journal of Information Systems, 27(2), 248–261. https://doi.org/10.1080/0960085X.2018.1445461

Committee of Sponsoring Organizations of the Treadway Commission. (2012). Internal control - integrated framework: framework and appendices.

Corea, C., & Delfmann, P. (2017). Detecting Compliance with Business Rules in Ontology-Based Process Modeling, in Leimeister, J.M.; Brenner, W. (eds..): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, 226–240.

Davis, F. (1985). A technology acceptance model for empirically testing new end-user information systems: Theory and results. MIT.

Delfmann, P., Steinhorst, M., Dietrich, H.-A., & Becker, J. (2015). The generic model query language GMQL – Conceptual specification, implementation, and runtime evaluation. Information Systems, 47(1), 129–177. https://doi.org/10.1016/j.is.2014.06.003

Dreyfus, D., & Iyer, B. (2006). Enterprise architecture: A social network perspective. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06) (pp. 178a–178a), Kauia, HI, USA. https://doi.org/10.1109/HICSS.2006.155

El Kharbili, M., Medeiros, A., Stein, S., & van der Aalst, W.M.P. (2008). Business process compliance checking: Current state and future challenges. Modellierung betrieblicher Informationssysteme (MobIS 2008), 107–113.

Elgammal, A., Turetken, O., van den Heuvel, W.-J., & Papazoglou, M. (2010). Root-cause analysis of design-time compliance violations on the basis of property patterns. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P. P. Maglio, M. Weske, J. Yang, & M. Fantinato (Eds.), Service-oriented computing (Vol. 6470, pp. 17–31). Springer Berlin Heidelberg.

Elgammal, A., Turetken, O., & van den Heuvel, W.-J. (2012). Using patterns for the analysis and resolution of compliance violations. International Journal of Cooperative Information Systems, 21 (1), 31–54. https://doi.org/10.1142/S0218843012400023

Fdhila, W., Indiono, C., Rinderle-Ma, S., & Reichert, M. (2015). Dealing with change in process choreographies: Design and implementation of propagation algorithms. Information Systems, 49(1), 1–24. https://doi.org/10.1016/j.is.2014.10.004

Fdhila, W., Rinderle-Ma, S., & Reichert, M. (2012). Change propagation in collaborative processes scenarios. In *8th international conference on collaborative computing: networking, applications and worksharing (CollaborateCom),*Pittsburgh, PA, USA.

Fellmann, M., Thomas, O., & Busch, B. (2011). A query-driven approach for checking the semantic correctness of ontology-based process representations. In: W. Abramowicz (ed) *Business Information Systems: 14th International Conference, BIS 2011* (pp. 62–73). Poznań, Poland, June 15–17, 2011. Proceedings. Springer-Verlag GmbH Berlin Heidelberg, Berlin, Heidelberg.

Foster, M.E.D., & Deardorff, M.A. (2017). Open Science Framework (OSF). *Journal of the Medical Library Association (JMLA),105*(2). https://doi.org/10.5195/JMLA.2017.88

Frank, U., Heise, D., Ulrich, K.H., Ferguson, D.F., Hadar, E., & Waschke, M.G. (2009). ITML: A domain-specific modeling language for supporting business driven IT management. In *Proceeding of the 9th OOPSLA workshop on domain-specific modeling,*Helsinki, Finland.

Gacitua-Decar, V., & Pahl, C. (2009). Automatic business process pattern matching for enterprise services design. In *2009 world conference on services – II* (pp. 111–118), Bangalore, India . https://doi.org/10.1109/SERVICES-2.2009.28

Ghanavati, S., Amyot, D., & Peyton, L. (2009). Compliance analysis based on a goal-oriented requirement language evaluation methodology. In *17th IEEE international requirements engineering conference* (pp. 133–142), Atlanta, GA, USA. https://doi.org/10.1109/RE.2009.42

Governatori, G., & Sadiq, S. (2009). The journey to business process compliance. *Handbook of Research on Business Process Modeling*, 426–454. https://doi.org/10.4018/978-1-60566-288-6.ch020

Gregor, S., & Hevner, A.R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337-355. https://doi.org/10.25300/MISQ/2013/37.2.01

Halle, S. (2011). Causality in message-based contract violations: A temporal logic "Whodunit". In *2011 IEEE 15th international enterprise distributed object computing conference, Helsinki, Finland* . https://doi.org/10.1109/EDOC.2011.21

Heckel, R., & Taentzer, G. (2020). *Graph transformation for software engineers: With applications to model-based development and domain-specific language engineering* (1st ed.). Springer eBook Collection. Springer International Publishing; Imprint Springer.

Hess, T.J., McNab, A.L., & Basoglu, K.A. (2014). Reliability generalization of perceived ease of use, perceived usefulness, and behavioral intentions. *MIS Quarterly*, *38*(1), 1–28. https://doi.org/10.25300/MISQ/2014/38.1.01

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

ISACA. (2013). *COBIT 5: Enabling information*.

Kittel, K., Sackmann, S., & Göser, K. (2013). Flexibility and compliance in workflow systems: The KitCom prototype. In *Proceedings of the CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering (CAiSE)* (pp. 154–160), Valencia, Spain.

Klinkmüller, C., & Weber, I. (2017). Analyzing control flow information to improve the effectiveness of process model matching techniques. *Decision Support Systems*, *100*(1), 6–14. https://doi.org/10.1016/j.dss.2017.06.002

Knackstedt, R., Eggert, M., Heddier, M., Chasin, F., & Becker, J. (2013). The relationship of is and law - The perspective of and implications for IS research. *ECIS 2013 Completed Research*. https://aisel.aisnet.org/ecis2013_cr/18/URL

Knuplesch, D., Fdhila, W., Reichert, M., & Rinderle-Ma, S. (2015). Detecting the effects of changes on the compliance of cross-organizational business processes. In P. Johannesson, M. Lee, S. Liddle, A. Opdahl, & Ó. Pastor López (eds) *Conceptual Modeling. Lecture Notes in Computer Science* (pp. 94–107), Springer, Cham. https://doi.org/10.1007/978-3-319-25264-3_7

Koetter, F., Kintz, M., Kochanowski, M., Wiriyarattanakul, T., Fehling, C., Gildein, P., Wagner, S., Leymann, F., & Weisbecker, A. (2016). An universal approach for compliance management using compliance descriptors. In M. Helfert, D. Ferguson, V. Méndez Muñoz, & J. Cardoso (eds) *Cloud Computing and Services Science: 6th International Conference, CLOSER 2016* (pp. 209–231). Rome, Italy, April 23–25, 2016, Springer International Publishing. Revised Selected Papers. Cham, s.l.

Koetter, F., Kochanowski, M., Weisbecker, A., Fehling, C., & Leymann, F. (2014). Integrating compliance requirements across business and IT. In *2014 IEEE 18th international enterprise distributed object computing conference,*Ulm, Germany. https://doi.org/10.1109/EDOC.2014.37

Kramer, T., Heinzl, A., & Neben, T. (2017). Cross-organizational software development: Design and evaluation of a decision support system for software component outsourcing. In *Hawaii international conference on system sciences* (pp. 343–352), Puako, Hawaii, United States. https://doi.org/10.24251/HICSS.2017.041

Kriglstein, S., Leitner, M., Kabicher-Fuchs, S., & Rinderle-Ma, S. (2016). Evaluation methods in process-aware information systems research with a perspective on human orientation. *Business & Information Systems Engineering*, *58*(6), 397–414. https://doi.org/10.1007/s12599-016-0427-3

Leopold, H., Meilicke, C., Fellmann, M., Pittke, F., Stuckenschmidt, H., & Mendling, J. (2015). Towards the automated annotation of process models. In *International Conference on Advanced Information Systems Engineering* (pp. 401–416), Stockholm, Sweden.

Marangunić, N., & Granić, A. (2015). Technology acceptance model: A literature review from 1986 to 2013. *Universal Access in the Information Society*, *14*(1), 81–95. https://doi.org/10.1007/s10209-014-0348-1

March, S.T., & Smith, G.F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

March, S.T., & Storey, V.C. (2008). Design science in the information systems discipline: An introduction to the special issue on design science research. *MIS Quarterly*, *32*(4), 725–730. https://doi.org/10.2307/25148869

Mican, D., Sitar-Tăut, D.-A., & Moisescu, O.-I. (2020). Perceived usefulness: A silver bullet to assure user data availability for online recommendation systems. *Decision Support Systems*, *139*(1), 113420. https://doi.org/10.1016/j.dss.2020.113420

Namiri, K., & Stojanovic, N. (2007). Using control patterns in business processes compliance. In *WISE 2007 Workshops* (pp. 178–190), Nancy, France.

Nickerson, R.C., Varshney, U., & Muntermann, J. (2013). A method for taxonomy development and its productservice in information systems. *European Journal of Information Systems*, *22*(3), 336–359. https://doi.org/10.1057/ejis.2012.26

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. In *1st International Conference on Design Science in Information Systems and Technology (DESRIST)* (pp. 83–106), Claremont, California, USA.

Radeschütz, S., Schwarz, H., & Niedermann, F. (2015). Business impact analysis—a framework for a comprehensive analysis and optimization of business processes. *Computer Science – Research and Development*, *30*(1), 69–86. https://doi.org/10.1007/s00450-013-0247-3

Rinderle, S., Reichert, M., & Dadam, P. (2004). Correctness criteria for dynamic changes in workflow systems: A survey. *Data & Knowledge Engineering*, *50*(1), 9–34. https://doi.org/10.1016/j.datak.2004.01.002

Rinderle-Ma, S., Reichert, M., & Weber, B. (2008). On the formal semantics of change patterns in process-aware information systems. In *Proc. 27th Int'l Conference on Conceptual Modeling (ER'08)* (pp. 279-293), Barcelona, Spain.

Rosenkrantz, D.J., Stearns, R.E., & Lewis, I.P.M.I. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, *6*(3), 563–581. https://doi.org/10.1137/0206041

Rudzajs, P., & Buksa, I. (2011). Business process and regulations: Approach to linkage and change management. In J. Grabis & M. Kirikova (Eds.), *Perspectives in business informatics research* (Vol. 90, pp. 96–109). Springer Berlin Heidelberg.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164. https://doi.org/10.1007/s10664-008-9102-8

Sackmann, S., & Kittel, K. (2015). Flexible workflows and compliance: A solvable contradiction? In J. Vom Brocke & T. Schmiedel (Eds.), *BPM - Driving innovation in a digital world* (pp. 247–258). Springer International Publishing.

Sackmann, S., Kühnel, S., & Seyffarth, T. (2018). Using business process compliance approaches for compliance management with regard to digitization: Evidence from a systematic literature review. In *International Conference on Business Process Management (BPM)* (pp. 409-425), Sydney, Australia.

Sadiq, S., Governatori, G., & Namiri, K. (2007). Modeling control objectives for business process compliance. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business process management* (Vol. 4714, pp. 149–164). Springer Berlin Heidelberg.

Santos, H., & Alves, C. (2017). Exploring the ambidextrous analysis of business processes: A design science research. In *International conference on enterprise information systems* (pp. 543–566), Porto, Portugal.

Schultz, M. (2013). Enriching process models for business process compliance checking in ERP environments. *International Conference on Design Science Research in Information Systems*, 120–135, Helsinki, Finland. https://doi.org/10.1007/978-3-642-38827-9_9

Schumm, D., Turetken, O., Kokash, N., Elgammal, A., Leymann, F., & van den Heuvel, W.-J. (2010a). Business process compliance through reusable units of compliant processes. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, F. Daniel, & F. M. Facca (Eds.), *Current trends in web engineering* (Vol. 6385, pp. 325–337). Springer Berlin Heidelberg.

Schumm, D., Karastoyanova, D., Leymann, F., & Strauch, S. (2011). Fragmento: Advanced process fragment library. *19th International Conference onInformation Systems Development*, Prague, Czech Republic.

Schumm, D., Leymann, F., Ma, Z., Scheibler, T., & Strauch, S. (2010b). Integrating compliance into business processes: process fragments as reusable compliance controls. *Proceedings of the Multikonferenz Wirtschaftsinformatik* (pp. 10), Göttingen,Germany.

Seeliger, A., Guinea, A.S., & Mühlhäuser, M. (2019). Process Explorer: Intelligent process mining guidance. *17th International Conference on Business Process Management (BPM)*, *2019*, (pp. 216–231), Vienna, Austria. https://doi.org/10.1007/978-3-030-26619-6_15

Seyffarth, T., Kühnel, S., & Sackmann, S. (2016). ConFlex: An ontology-based approach for the flexible integration of controls into business processes. *Proceedings of theMultikonferenz Wirtschaftsinformatik 2016 (MKWI'16)*, *2016*, (pp. 1341–1352), Ilmenau, Germany.

Seyffarth, T., Kühnel, S., & Sackmann, S. (2017a). A taxonomy of compliance processes for business process compliance. In *15th international conference on business process management, business process management forum. In: Lecture Notes in Business Information Processing (LNBIP)* (pp. 71–87), Barcelona, Spain.

Seyffarth, T., Kühnel, S., & Sackmann, S. (2017b). Welche Compliance-Anforderungen sind für Geschäftsprozessänderungen relevant?: Ein Ansatz zur Modellierung der Beziehungen. In *Proceedings of the Informatik 2017, Lecture Notes in Informatics (LNI)* (pp. 1641–1646), Chemnitz, Germany.

Seyffarth, T., Kühnel, S., & Sackmann, S. (2018). Business process compliance and business process change: An approach to analyze the interactions. In *Business Information Systems. BIS 2018. Lecture notes in business information processing* (pp. 176–189), Berlin, Germany.

Seyffarth, T., Kühnel, S., & Sackmann, S. (2019). Business process compliance despite change: Towards proposals for a business process adaption. *Information Systems Engineering in Responsible Information Systems. CAiSE 2019. Lecture Notes in Business Information Processing*, *350*, 227–239. https://doi.org/10.1007/978-3-030-21297-1_20

Seyffarth, T., & Raschke, K. (2018). BCIT: A tool for analyzing the interactions between business process compliance and business process change. *Proceedings of the Dissertation Award and Demonstration, Industrial Track at BPM*, *2018*, 81–85. http://ceur-ws.org/Vol-2196/BPM_2018_paper_17.pdf

Seyffarth, T., & Raschke, K. (2020). BCIT: A tool to recommend compliant business processes based on process adaption. In *Proceedings of the best dissertation award, doctoral consortium, and demonstration & resources track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020)* (pp. 107–111), Vienna, Austria.

Sillaber, C., & Breu, R. (2012). Managing legal compliance through security requirements across service provider chains: A case study on the German Federal Data Protection Act. *GI-Jahrestagung, Informatik 2012*, (pp. 1306–1318), Braunschweig, Germany.

Sonnenberg, C., & Vom Brocke, J. (2012). Evaluation patterns for design science research artefacts. In M. Helfert & B. Donnellan (Eds.), *Practical aspects of design science* (Vol. 286, pp. 71–83). Springer Berlin Heidelberg.

Sturm, B., & Sunyaev, A. (2019). Design principles for systematic search systems: A holistic synthesis of a rigorous multi-cycle design science research journey. *Business & Information Systems Engineering*, *61*(1), 91–111. https://doi.org/10.1007/s12599-018-0569-6

The Institut der Wirtschaftsprüfer in Deutschland e.V. [Institute of Public Auditors in Germany, Incorporated Association]. (2002a). *Principles of proper accounting when using information technology* (IDW AcP FAIT 1).

The Institut der Wirtschaftsprüfer in Deutschland e.V. [Institute of Public Auditors in Germany, Incorporated Association]. (2002b). *The audit of financial statements in an information technology environment* (IDW AuS 330).

Turetken, O., Elgammal, A., van den Heuvel, W.-J., & Papazoglou, M. (2011). Enforcing compliance on business processes through the use of patterns. *ECIS 2011 Proceedings*, 2011, Helsinki, Finland. https://aisel.aisnet.org/ecis2011/5/

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. In *17th European Conference on Information Systems* (pp. 2206–2217), Verona, Italy.

Weber, B., Reichert, M., & Rinderle-Ma, S. (2008). Change patterns and change support features: Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, *66*(3), 438–466. https://doi.org/10.1016/j.datak.2008.05.001

Webster, J., & Watson, R.T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii–xxiii.

Weske, M. (2019). *Business process management: Concepts, languages, architectures* (3rd ed.). Springer-Verlag Berlin Heidelberg.

Winter, R., & Fischer, R. (2006). Essential layers, artifacts, and dependencies of enterprise architecture. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)* (pp. 30–37), Hong Kong, China. https://doi.org/10.1109/EDOCW.2006.33

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Berlin Heidelberg.

Yousafzai, S.Y., Foxall, G.R., & Pallister, J.G. (2007). Technology acceptance: A meta-analysis of the TAM: Part 1. *Journal of Modelling in Management*, *2*(3), 251–280. https://doi.org/10.1108/17465660710834453