Patrick Delfmann, Michael Hübers

Towards Supporting Business Process Compliance Checking with Compliance Pattern Catalogues

A Financial Industry Case Study

Business process compliance checking serves to discover legal or self-regulatory violations in business processes using process models. This way, companies can react to new regulations and avoid violations quickly, hence prevent negative monetary or even legal consequences. In the business process compliance management literature, we find an abundance of approaches supporting business process compliance checking. Although many of these approaches show promise to support business process compliance checking by providing model checking-like methodologies, hardly any of them provide a common list of relevant compliance rules or violations that should be checked for. With this paper, we aim at making a step towards comprehensive catalogues of compliance rules that can be used as input for business process compliance checking approaches. In particular, we analyse two legal documents providing a set of compliance rules for service processes in financial industries. We derive compliance patterns from them and apply them to a large business process model coming from a German IT service provider for banks, using a graph pattern-based compliance checking approach. As a result, we show that deriving business process compliance rules from legal texts leads to meaningful patterns matching several subsections of common process models of financial services. Hence, we can expect catalogues of such patterns to be promising for supporting business process analysts in compliance checking.

1 Introduction

In recent years, business process compliance management (BPCM) has become a common service in highly regulated industries. BPCM serves to assure the conformance of a company's business processes with law and internal regulations. Its goal is to prevent negative monetary or even legal consequences resulting from (mostly unconsciously committed) regulatory violations. Many BPCM approaches have been developed in the last few years (El Kharbili et al. 2008, Karagiannis 2008, Schwab 2012). One special kind of such approaches, which will be addressed in this paper, is business process compliance checking. The idea of business process compliance checking is to detect subsections of business processes that either indicate the fulfilment of a compliance requirement or its violation. Hence, through applying business process compliance checking, we can detect both compliant and non-compliant business process subsections. To detect such subsections, business process compliance checking approaches make use of business process models (and further, related models like, e.g., organizational charts and data models) and search them for subsections either fulfilling or violating a compliance rule (e.g., El Kharbili et al. 2008, Becker et al. 2012). The subsections that are searched for must correspond with a specific structure and contents – a model pattern – that specifies the compliance fulfilment/violation and serves as input for the compliance checking approach.

Figure 1 illustrates business process compliance checking with a simple rule called the 4-eye-principle (cf. Delfmann et al. 2015). It prescribes that a document has to be double-checked before it can be cleared. The double-check has to be performed by two different employees, where the employee who conducts the second check has to be superordinate to the first one.

In terms of a compliance pattern representing the *fulfilment* of a compliance rule, in a process model, there has to be an activity performing a check, which is followed by a further check activity across a process path. Both activities handle the same document, and the second activity is executed by an employee different from the first one. Furthermore, the second employee has to be superordinate to the first one. For a compliance check, this means that the structure described above must be contained in a process model, and, furthermore, the hierarchical order of employees must be contained in an organizational chart. A corresponding compliance checking approach takes a compliance pattern like the 4-eye-principle as input and returns all pattern occurrences in the models to be checked. In the exemplary case, the pattern describes a compliant model section. However, to find compliance *violations*, it is also possible to design anti-patterns describing model sections that violate compliance rules. For instance, to detect violations of the 4-eye-principle, we would formulate a pattern that either searched for double checks neglecting different responsible persons or for checks missing a second check.

A compliance checking process in a company using a compliance checking approach and patterns as outlined above typically proceeds as follows: The compliance responsible defines which compliance rules coming from law or internal regulations are relevant for the processes to be checked. These rules are then transformed into compliance patterns and applied to the process models of interest using a compliance checking approach. The compliance checking results in a set of pattern matches that may indicate either compliant parts of the process models of interest (in case the patterns depict compliant situations) or non-compliant parts (in case the patterns depict compliance violations). Usually, compliance pattern matches only indicate that there *may* be a compliant part or a compliance violation of the process model. Full automation is hardly possible due to the degrees of freedom of process modelling. Hence, every match has to be checked whether or not it is really a compliance fulfilment/violation. The benefit of compliance checking approaches, however, is still seen in time savings, as the pure search for compliance-relevant process parts can be heavily accelerated.

Many compliance checking approaches exist, which are based on graph pattern matching, linear temporal logic, computation tree logic, or similar concepts mostly coming from applied or theoretical computer science (cf. Becker et al. 2012, Delfmann et al. 2015). However, hardly any of these approaches provide comprehensive lists, catalogues or repositories of compliance patterns – neither general patterns relevant for most industries nor specialized ones related to particular industries or branches. Although no such list could ever be exhaustive in terms of providing every relevant compliance rule for a company, they can very likely support compliance managers in composing relevant patterns for the respective company.

With this paper, we strive to make a step towards a common catalogue of compliance patterns that can serve as input for business process compliance checking approaches. We

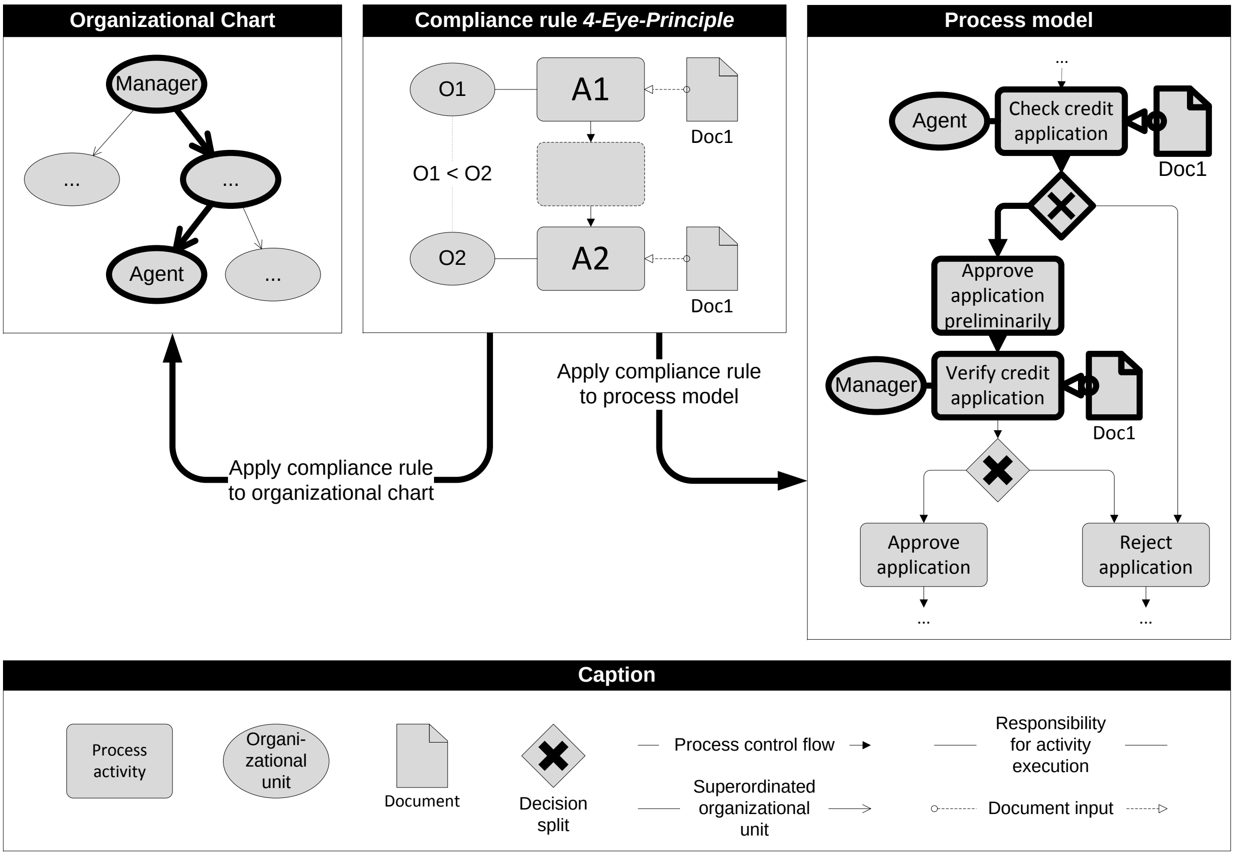

Figure 1: Business process compliance checking

argue that, with such a catalogue along with a compliance checking approach, compliance managers can be supported in their everyday work. Therefore, we analyse two particular legal texts – in this case relevant for the financial industry – and derive relevant compliance rules from them. We then transform these rules into patterns as outlined in Figure 1. For each rule, we consider that analysts might want to search both for compliant process sections and for non-compliant process sections. Hence, we formulate both patterns representing a process subsection complying with a rule and patterns representing a process subsection violating a rule (anti-patterns). To make the patterns reusable for different compliance checking approaches, we refrain from a special modelling language or a special compliance checking approach. We rather use a graph-like representation that makes it possible to extract the essence of the pattern easily, that is, to realize how a process model section has to look like in order to fulfil the pattern (i.e., either to be compliant or, in case of an anti-pattern, to violate a compliance rule). In order to test the patterns we identified, we apply them to a large process model coming from a German IT service provider for banks, which represents the business processes of wide-spread banking software. The checking is done using a graph pattern-based compliance checking approach, which was available from former research (Delfmann et al. 2015). Note that we do not claim to provide a comprehensive catalogue. We rather show how we can derive compliance rules from legal texts, how they can be represented as patterns and, most importantly, that such common patterns can indeed be found in typical financial services process models using (semi-)automatic compliance checking, hence that such catalogues very likely make sense. This paper is an extension of a previous study, which was extended by additional patterns and a discussion on anti-patterns (cf. Delfmann and Hübers 2014).

The remainder of this paper is structured as follows: In Section 2, we discuss the results of a literature survey taken from previous research, which suggests that up to now, related work does not provide catalogues of compliance patterns so far. Section 3 reports on how we analysed legal texts and how we transformed them into patterns. Furthermore, we elaborate on what (semi-formal, generic) format compliance patterns of a common catalogue should have and why. In Section 4, we apply the compliance patterns to a real-world process model. In particular, we demonstrate how we can transform the patterns from our preliminary catalogue into pattern queries of a particular model query language. Then, we execute the queries onto a large process model of a German IT service provider and provide statistics of how many of the patterns were actually found in the process model. In the discussion in Section 5, we estimate the reusability of the compliance patterns based on the experiences of the real-world application. Furthermore, we give an outlook to further research.

2 Related Work

To estimate the state-of-the-art of current research on compliance rule catalogues, we adopted findings from a literature survey on business process compliance checking (Becker et al. 2012). The analysis comprised compliance checking approaches that are all based on some kind of pattern matching, that is, they search for subsections in process models, the structure and contents of which indicate either a compliance violation or can be considered compliant. Hence, they can be considered to support compliance checking based on pattern catalogues.

The approaches are different in their underlying methodology for identifying model subsections. Hence, their expressive power differs. This means that some of them are able to identify model subsections of arbitrary complexity, some of them are restricted to express the order of process model activities only. Furthermore, they are different in the modelling languages they support. Some approaches are independent of a particular modelling language; some are specially tailored for particular modelling languages.

Moreover, some of the approaches solely exist as a concept, some come with a prototype, and only one was applied in a real-world scenario (for a detailed analysis of these characteristics, cf. Becker et al. 2012). Depending on the application scenario, that is, the business domain, the process models to be analysed, the compliance patterns to be checked and the technical set-up, it has to be decided which approach should be employed. The literature survey has shown that, so far, no compliance checking approach exists that comes with a comprehensive catalogue of compliance patterns. In the following, we list related approaches that could take compliance patterns as input. Furthermore, we outline which of them come with a list of compliance patterns. We distinguish three categories (cf. Table 1):

1. Approaches that come with only examples of compliance patterns
2. Approaches that come with examples of compliance patterns and apply them with help of an implementation of the approach
3. Approaches that come with real-world compliance patterns, that is, patterns that came from regulations in a real-world compliance checking scenario

The approaches listed here originate from a literature survey we conducted in a previous

#	Approach	Category
1	Accorsi, Lowis, and Sato (2011)	2
2	Arbab, Kokash, and Meng (2009), Elgammal, Turetken, van den Heuvel, and Papazoglou (2010), Kokash and Arbab (2009), Schumm, Turetken, Kokash, Elgammal, Leymann, and van den Heuvel (2010)	2
3	Awad (2007), Awad (2010), Awad, Decker, and Weske (2008), Awad, Smirnov, and Weske (2009a), Awad, Smirnov, and Weske (2009b), Awad and Weske (2009), Awad and Weske (2010)	2
4	Damaggio, Deutsch, Hull, and Vianu (2011)	2
5	Eshuis (2006)	2
6	Eshuis and Wieringa (2004)	2
7	Förster, Engels, and Schattkowsky (2005), Förster, Engels, Schattkowsky, and Van Der Straeten (2007)	2
8	Ghose and Koliadis (2007)	1
9	Goedertier and Vanthienen (2006)	1
10	Governatori and Milosevic (2006), Governatori, Milosevic, and Sadiq (2006), Governatori and Rotolo (2010), Hoffmann, Weber, and Governatori (2009), Lu, Sadiq, and Governatori (2008a), Lu, Sadiq, and Governatori (2008b)	1
11	Khaluf, Gerth, and Engels (2011)	2
12	Knuplesch, Ly, Rinderle-Ma, Pfeifer, and Dadam (2010), Ly, Göser, Rinderle-Ma, and Dadam (2008), Ly, Rinderle-Ma, and Dadam (2006), Ly, Rinderle-Ma, and Dadam (2010), Ly, Rinderle-Ma, Göser, and Dadam (2009), Ly, Rinderle, and Dadam (2008)	2
13	Küster, Ryndina, and Gall (2007)	2
14	Kumar and Liu (2008)	1
15	Liu, Müller, and Xu (2007)	2
16	Lohmann and Wolf (2010)	2
17	Monakova, Kopp, Leymann, Moser, and Schäfers (2009)	2
18	Müller (2010)	3
19	Sadiq, Governatori, and Namiri (2007)	1
20	Schleicher, Anstett, Leymann, and Schumm (2010)	2
21	Trčka, van der Aalst, and Sidorova (2009)	1
22	Wang and Zhao (2011)	2
23	Wörzberger, Kurpick, and Heer (2008a), Wörzberger, Kurpick, and Heer (2008b)	2
24	Wolter and Meinel (2010), Wolter, Miseldine, and Meinel (2009)	2
25	Xiangpeng, Cerone, and Krishnan (2006)	1
26	Becker, Bergener, Delfmann, and Weiß (2011)	2
27	Becker, Delfmann, Dietrich, Eggert, and Steinhorst (2014)	3

Table 1: Compliance checking approaches and their included compliance patterns

research project (Becker et al. 2012). Note that we do not elaborate on technical details of the underlying compliance checking technique of the approaches here. For this paper, it is particularly important that all of the approaches incorporate a compliance checking technique similar to that outlined in Section 1.

Virtually all of the analysed approaches present a compliance checking methodology, but abstain from providing particular compliance patterns. Most of the authors rely on example patterns to illustrate the functionality of their approach (cf. Table 1, Category 1); some even apply these example patterns in line with an implementation of their approach (cf. Table 1, Category 2). Only two articles report on an implementation and application of their approaches to a real-world industry scenario and provide a list of compliance patterns (cf. Table 1, Category 3). However, both articles provide a short list of exemplary patterns only and do not consider comprehensive catalogues. The reason is that the goal of both articles is to exemplarily apply and evaluate a compliance checking approach in a particular scenario, rather than providing a comprehensive catalogue of compliance patterns.

Hence, the survey suggests that, up to now, there is no general or at least industry branch-related list of reusable compliance patterns.

3 Business Process Compliance Patterns

To establish a first catalogue of business process compliance patterns, we chose two legal texts, which contain compliance rules relevant for one of the most highly regulated industries: the financial sector (Abdullah et al. 2010). In particular, we analysed the German Money Laundering Act (Geldwäschegesetz, GwG) and the Minimum Requirements for Risk Management (MaRisk) (Lorenz 2008).

As these legal texts are provided in text form only, we had to transform the text-based legal regulations into formal graph-based compliance patterns.

3.1 Representation of Compliance Patterns

To do so, we need to provide a way of pattern representation that takes into account the graph characteristic of business process models (and other kinds of models). So, for example, we must be able to represent process model nodes, edges, paths and captions. Figure 2 outlines the way we represent compliance patterns in the following. Alternative ways of representing compliance rules haven been proposed, for instance, by Awad et al. (2011), Knuplesch et al. (2013), and Siena et al. (2012). They obey a specified abstract syntax and are specifically related to a checking mechanism. We abstain from such a specialized formalization, as the patterns we aim at providing in the form of catalogues should be reusable by different compliance checking approaches (cf. Section 1).

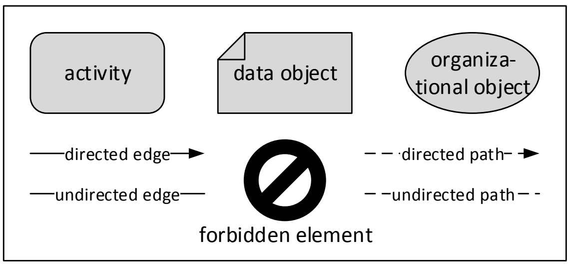

Figure 2: Representation of business process compliance patterns

The compliance patterns we derived from the legal texts consist of activity nodes, data object nodes, organizational object nodes, edges between these nodes, paths (i.e., edges of the pattern that represent paths over multiple nodes in the models to be checked; cf. Section 1), and captions. Furthermore, we introduce the “forbidden” element to display that in some situations, certain objects or relationships are not allowed to appear within a process model section to match a pattern. We decided to use these kinds of nodes and edges as they are the most common elements in process models. Furthermore, the regulations we identified from the legal texts were referring mostly to such parts of a business process. Captions, that is, the names nodes and/or edges should contain to match a pattern, are placed within the boundaries of the nodes or onto the edges/paths.

Using this way of representation, the compliance rule example outlined in Section 1 would be represented like illustrated in Figure 3. In particular, the pattern prescribes that a model section matching the pattern should contain an activity labelled with a phrase containing the term “check”, which is followed by another activity over a process path of previously unknown length. That second activity should be labelled with a phrase containing the term “verify”. The first activity must be annotated with an input data object, the second one with an output data object. Both data objects should be the same (indicated through the variables A and B contained in the captions and the equation A=B). Furthermore, the activities have to be annotated with organizational objects, where a directed path leads from the second organizational object to the first one, indicating that the second organizational object is superordinate to the first one.

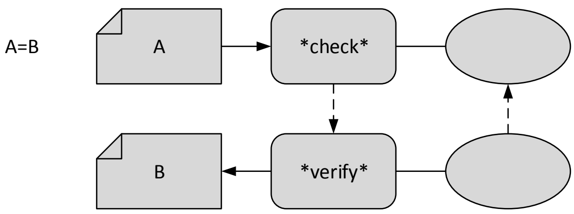

Figure 3: An exemplary compliance rule “4-eye-principle”, represented as a graph pattern

As already outlined in Section 1, in some situations it makes sense to search not only for compliant parts of a process model but also for such parts violating compliance rules. A process model part violating the above mentioned rule would violate the 4-eye-principle, hence the check and verify activity would be performed by the same person. A corresponding anti-pattern that could be used for identifying compliance violations could be represented as follows (cf. Figure 4). As opposed to the pattern matching to process model sections complying with the 4-eye-principle, the anti-pattern requires the annotated organizational units to carry the same label and does not require them to be related via a directed disciplinary path.

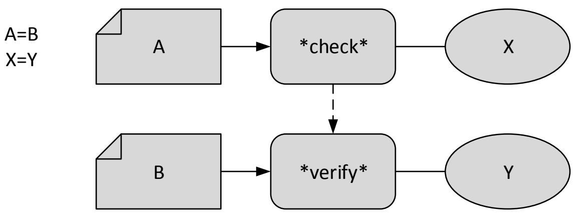

Figure 4: An exemplary compliance rule "4-eye-principle", representing a violation

We argue that deriving both patterns and anti-patterns from legal regulations makes sense as this way, we can search both for compliant process parts and non-compliant ones, so we can increase the chance to find compliance violations, which sometimes is hindered by the fact that the same process can be modelled in different ways.

To particularize compliance patterns, they can be further annotated with explanations. Certainly, as these patterns do not adhere to a special syntax, they have to be transformed into a representation fitting to the compliance checking approach they are to be used with.

3.2 Transformation of Legal Texts into Compliance Patterns

To obtain a pattern as described above, we need to transform a legal text containing a compliance rule into a compliance pattern. Such a transformation requires an analysis of the legal text concerning the nodes and edges contained in the target pattern. In particular, we analysed the legal texts with regard to statements describing tasks, involved persons and/or institutions, documents and/or data/information, execution order and relationships between tasks, persons/institutions, and documents/data/information. Based on these informations, we could create compliance patterns by transforming text describing a task into a process pattern activity, text describing a person/institution into an organizational object, text describing a document/data/information into a data object and text describing execution order and other relationships into edges or paths (related work on extracting models automatically from natural language texts has been done, e.g., by Hassan and Logrippo (2009) and Maxwell and Anton (2009). A survey of related approaches is provided by Ghanavati et al.(2011)).

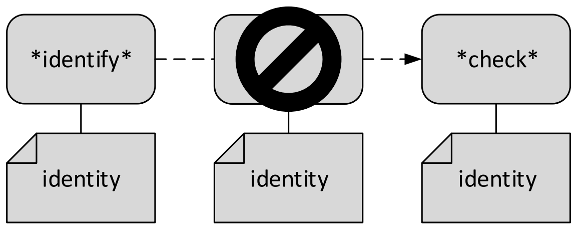

Figure 5: Pattern "Identification" of GwG

Consider the following example taken from §1 of the GwG [translated from German]: "*Identification in the context of this law consists of (1) the determination of the identity by data collection and (2) the check of the identity.*" This paragraph prescribes for every identification of a person (e.g., a customer of a bank) that, as soon as the identification data of a person have been gathered, these data have to be verified. For a compliance pattern, this means that an activity "*identify*" has to be followed by another activity "*check*", before the identity data can be reused in the further course of the process. Hence, between the identification and the check, no further activity is allowed that uses the identity data. Both activities have to be connected to a data object "identity". The names of the activities are specified using wild-cards as we cannot forecast how the terms "identify" and "check" will be inflected in an activity caption and what further words will be used. A corresponding compliance pattern looks like the one outlined in Figure 5.

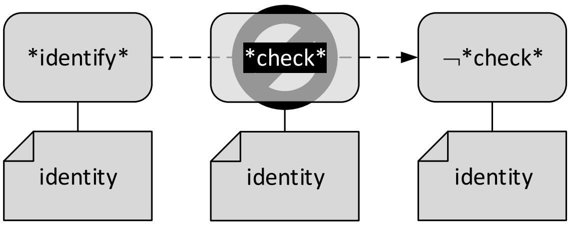

Figure 6: Anti-pattern "Identification" of GwG

The corresponding anti-pattern would require using personal data before they are verified. Hence, for the anti-pattern, the "*check*" activity would be the forbidden one, whereas the activity using the personal data and representing no check would be located at the end of the path (cf. Figure 6).

Note that that every process model to be checked with an automatic or semi-automatic compliance checking approach has to be unambiguous, that is, terminologically standardized. Otherwise, any attempt to extract information automatically (like it is done with compliance checking approaches) might result in ambiguous or even useless results.

Terminological standardization can be achieved, for instance, by enforcing a process modeller to use predefined correct terms and phrases coming, for instance, from an enterprise glossary (like it is proposed, e.g., by Delfmann et al. (2009)). Another way of assuring unambiguity of process model element labels is to annotate formal process ontologies defining the exact meaning of every used concept (cf., e.g., Thomas and Fellmann (2009)). To assure unambiguous models, both ontology annotation or term and phrase enforcement have to take place already during the modelling process. Otherwise, once we try to disambiguate existing process models, we cannot guess what the person who modelled the process exactly meant with a label given to a process element (e.g., a "bill" could be a banknote or an invoice). Hence, to use the compliance patterns we introduce in the following, it is necessary to align the terms used in their labels to the glossary of terms used in the respective company the patterns should be applied to.

A very restrictive way to ensure terminological unambiguousness in conceptual model is to not only predefine allowed terms but also the way how they can be combined. For instance, the *PICTURE* modelling language (Becker et al. 2007) provides 24 process activity types with a predefined semantics specially tailored for public administration processes. This way, the modeller has no chance to produce ambiguous labels. *icebricks* is another process modelling language (Becker et al. 2013) that enforces the modeller to use a combination of one business object term (a noun predefined through an enterprise glossary) and one business procedure term (a verb predefined through an enterprise glossary) to denote a process model activity.

As a precondition to use any automatic compliance checking approach at all is to have terminologically unambiguous models, we also employed one in our study. In particular, we analysed process models that were built with the icebricks language (cf. Section 4.2).

3.3 Identified Compliance Patterns

In total, we could identify 21 business process compliance patterns out of the GwG and 82 patterns out of MaRisk. In the following, we will outline three rules out of GwG and five rules out of MaRisk and present corresponding patterns and anti-patterns. Please note that we not claim to present a comprehensive catalogue here. We rather outline how such a catalogue can be designed and how corresponding patterns look like. The first pattern of GwG already served as an example in Section 3.2 (see above). Further patterns are presented below.

For instance, one rule explicated in GwG prescribes that before every financial transaction, the involved client has to be identified according to the definition in Section 3.2. A corresponding pattern is illustrated in Figure 7. Here, a process activity containing either the term “transfer”, “deposit” or “pay” has to be preceded by another activity named like “identify”, either directly or across a path of activities.

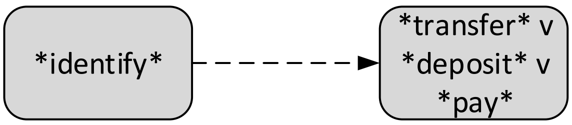

Figure 7: Pattern “identification – transaction” of GwG

The corresponding anti-pattern is depicted in Figure 8. In order to violate the compliance rule, a process model would have to contain a transaction activity (containing the terms “transfer”, “deposit”, or “pay”), which is preceded by a path that originates from a start activity of the process. The path is not allowed to contain an “identify” activity. Actually, this means that nowhere before the transaction, an identification takes place, so the compliance rule is violated. The start activity is characterized through the “forbidden” incoming edge.

Another rule out of GwG requires recurring checks of data or of documents. Such a rule can be represented as a compliance pattern as follows: an activity holding a label that contains the term “check” is connected to itself via a directed path of arbitrary length. This means that a corresponding process model must contain a loop of sequence flows containing the activity. Furthermore, the activity has to be connected to a data object (cf. Figure 9).

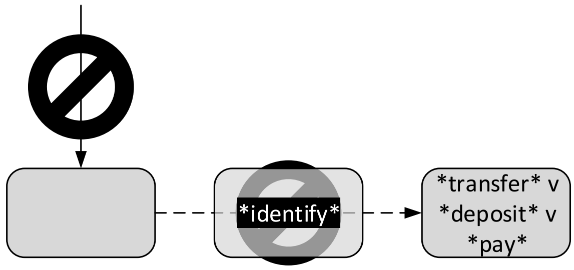

Figure 8: Anti-pattern “identification – transaction” of GwG

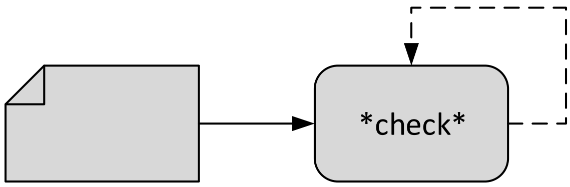

Figure 9: Pattern “recurring check” of GwG

In Figure 10, we provide two possible anti-patterns for the compliance rule “recurring checks”. Either the check is not recurring or there is no check at all. The former would require a process model to contain a “check” activity that is not connected to itself via a path, the latter would require the process model to contain no “check” activity at all.

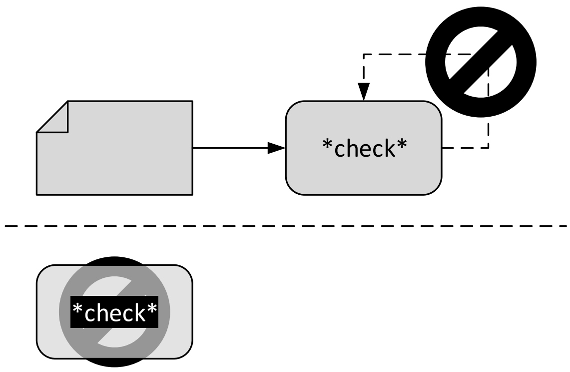

Figure 10: Two possible anti-patterns “recurring check” of GwG

A rule contained in MaRisk dictates that before every trade, the involved parties have to explicitly agree on the conditions of the trade. A corresponding pattern is shown in Figure 11. An activity named with a phrase containing the term "trade" has to be preceded by an activity named with a phrase containing "agree", which is, in turn, connected to a data object named "conditions".

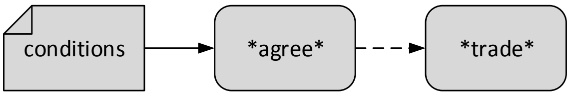

Figure 11: Pattern "agreement" of MaRisk

The anti-pattern representing a violation of the "agreement" rule is shown in Figure 12. The structure of this anti-pattern is similar to that of the anti-pattern "identification – transaction" of GwG. The pattern requires that, in order to violate the "agreement" rule, a process model containing a "trade" activity must not contain an "agree" activity connected to a "conditions" data object anywhere on a path before it. In other words, a path coming from any process start element and ending in the "trade" activity is not allowed to carry such an "agree activity" to match the pattern.

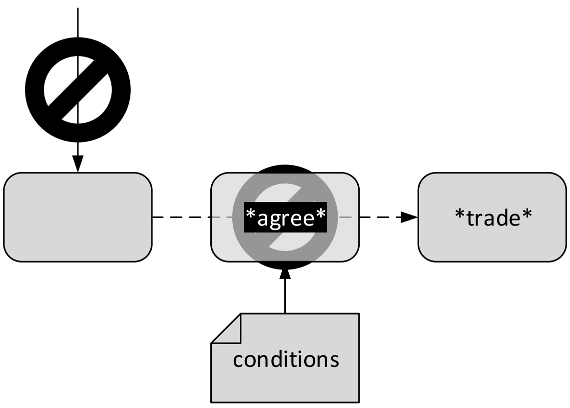

Figure 12: Anti-pattern "agreement" of MaRisk

A further MaRisk rule requires that any data produced during a trade transaction have to be forwarded to a settlement department. The corresponding pattern consists of a path of arbitrary length starting in an activity named like "trade", which has an output data object. The path ends in an activity that has an input data object, which has to be the same one as that of the first activity (A=B). Furthermore, it is connected to an organizational object named "settlement" (cf. Figure 13).

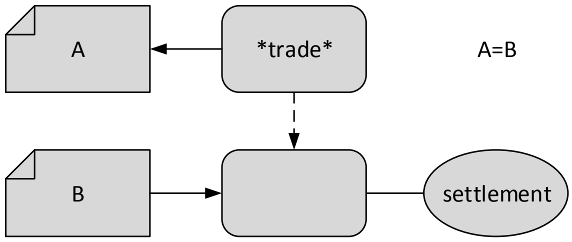

Figure 13: Pattern "settlement" of MaRisk

The corresponding anti-pattern of the "settlement" rule can be designed as follows (cf. Figure 14): Any "trade" activity found in the process model is not allowed to be followed anywhere on a path succeeding it by an activity sending the data produced by the "trade" activity to a settlement department. Note that this pattern would even match a process model containing a sending activity, but with inappropriate data.

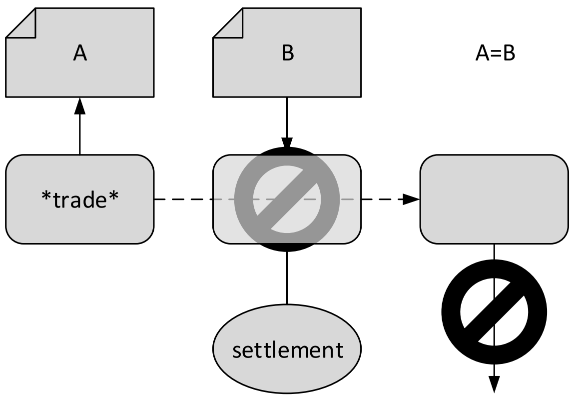

Figure 14: Anti-pattern "settlement" of MaRisk

The "confirmation" pattern example extracted from MaRisk is based on a rule prescribing that every trade transaction has to be confirmed. For the pattern, this means that a "trade" activity must be followed by a "confirm" activity over a path of arbitrary length. To assure that the confirmation is related to the trade, both activities must act upon the same data. Thus, the first activity produces output data that are consumed by the second one. To assure identity of both data objects, their labels have to be the same (A=B, cf. Figure 15).

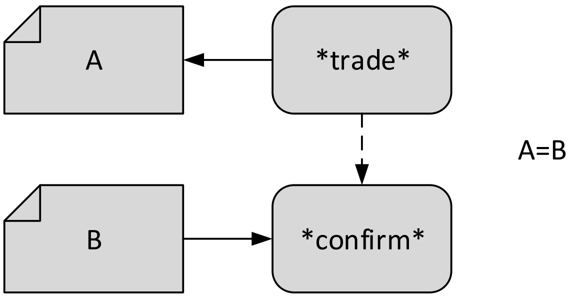

Figure 15: Pattern "confirmation" of MaRisk

The structure of the corresponding anti-pattern is similar to that of the "settlement" anti-pattern. It highlights every process model section that contains a "trade" activity not followed by a confirmation at all (cf. Figure 16).

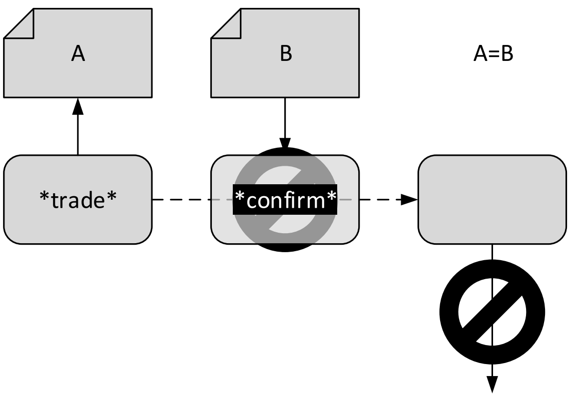

Figure 16: Anti-pattern "confirmation" of MaRisk

MaRisk prescribes a further rule that is related to dunning processes. It requires that, for every transaction related to a credit (e.g., redemption or transfer of documents), the financial institute has to establish a dunning process. A corresponding compliance pattern requires that, in every process related to a credit, an activity containing the term "credit" must be related to an activity across a sequence flow path of arbitrary length containing the term "dun" (cf. Figure 17).

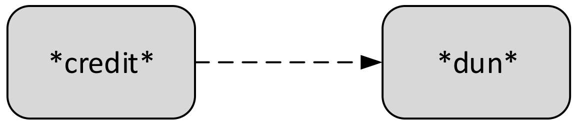

Figure 17: Pattern "dunning" of MaRisk

A corresponding anti-pattern can be designed as depicted in Figure 18. The pattern requires that if a process model contains an activity containing the term "credit", it must not be followed by another activity containing the term "dun" until the end of the process model.

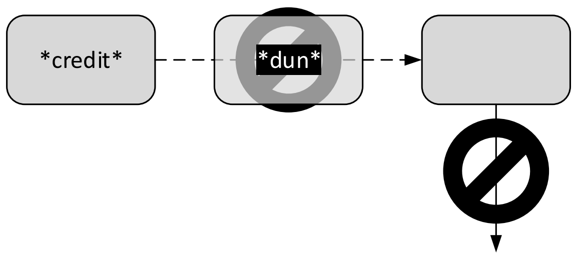

Figure 18: Anti-pattern "dunning" of MaRisk

The last pattern we present here also originates from MaRisk and is based on a rule prescribing that before a credit transaction can be executed, an explicit decision has to be made before. The corresponding pattern (cf. Figure 19) describes a process model section that contains a "transaction" activity that is preceded by a "decision" activity.

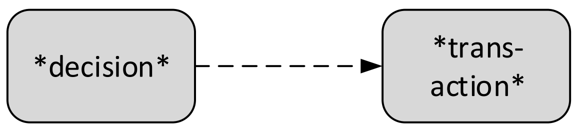

Figure 19: Pattern "credit decision" of MaRisk

The corresponding anti-pattern hence marks a process model section as non-compliant if it contains a “transaction” activity that is not preceded by a “decision” activity over a path of arbitrary length starting at the beginning of the process model (cf. Figure 20).

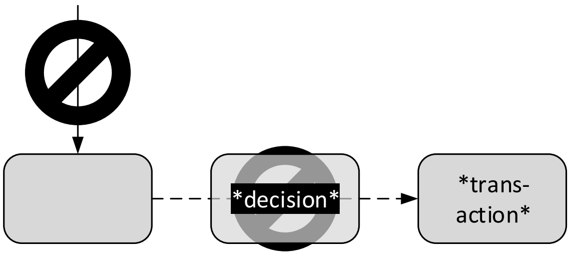

Figure 20: Anti-pattern “credit decision” of MaRisk

4 Application

In order to estimate the usefulness of the compliance patterns we identified, we applied them to a large business process model of a German IT service provider for banks using a graph pattern-based business process compliance checking approach named GMQL we developed in the past (Delfmann et al. 2015). Note that this approach was chosen arbitrarily. We could have chosen any other approach outlined in the related work section, as all these approaches have in common that they are searching for process model sections obeying some kind of pattern. However, this approach, alongside with a modelling tool implementation was available to us, as well as the process model of the IT service provider for banks, which was also accessible by the tool. Furthermore, GMQL can handle models of any graph-like modelling language, hence using GMQL, we were flexible in matters of the types of process models to analyse.

Most importantly, GMQL is flexible in the way how properties of process model nodes and edges (such as labels) can be compared and matched. The possibilities reach from simple string comparison over linguistic matching (synonyms, homonyms, composites, derivations, hyponyms, hypernyms, etc.) to ontological concept matching. This is necessary to keep unambiguousness not only in the process models but also within the compliance checking process.

4.1 The Compliance Checking Approach

GMQL consists of nested statements that build up a pattern successively. In particular, the statements available return nodes of a distinct type (e.g., process model activities or organizational units), nodes with defined labels (e.g., “*check*” or “*transfer*”), edges and paths as well as required or forbidden nodes on paths. As the purpose of this paper is not to introduce this approach, we refer to the literature for details on its syntax and semantics (Delfmann et al. 2015). We rather outline its mode of operation. Consider a rule prescribing that a “check” activity should be followed by a “verify” activity further on in the process. A corresponding pattern would be specified as follows using the compliance checking approach (note that for reasons of simplicity, we use a simplified syntax here; furthermore, we assume that the models to be analysed are terminologically standardised, hence the terms “check” and “verify” are part of the enterprise glossary that was used to create the process models):

```
DirectedPaths (
    ElementsOfType (
        ObjectsWithValues (“*check*”),
        Activity),
    ElementsOfType (
        ObjectsWithValues (“*verify*”),
        Activity)
)
```

When a process model is checked against this pattern, the inner statements *ObjectsWithValues* are evaluated first, and the checking approach returns all model nodes that are labelled with terms containing the word “check” (or “verify”, respectively). Then, these sets of nodes are restricted to nodes of the type

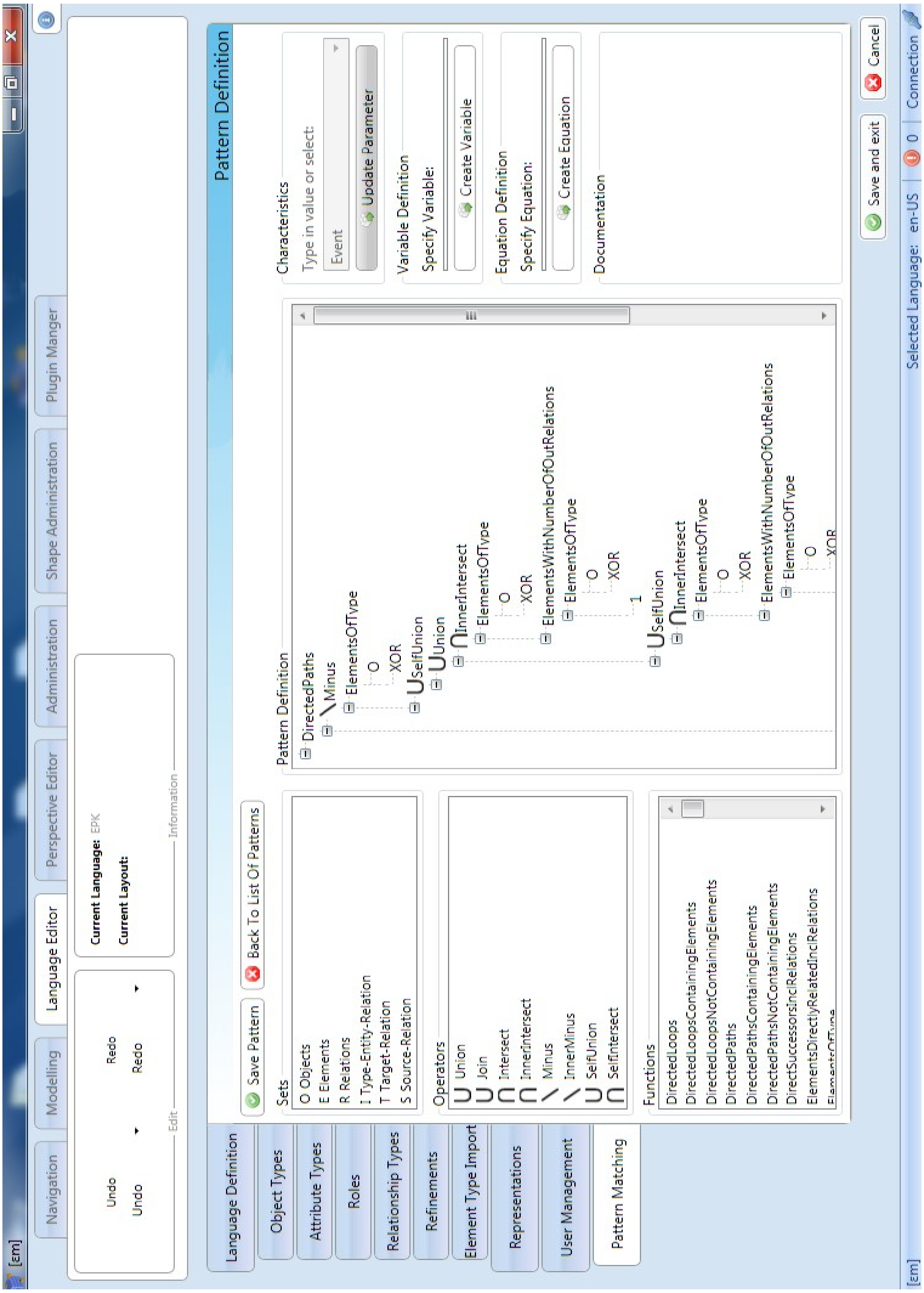

Figure 21: Pattern editor (cf. Delfmann et al. 2015)

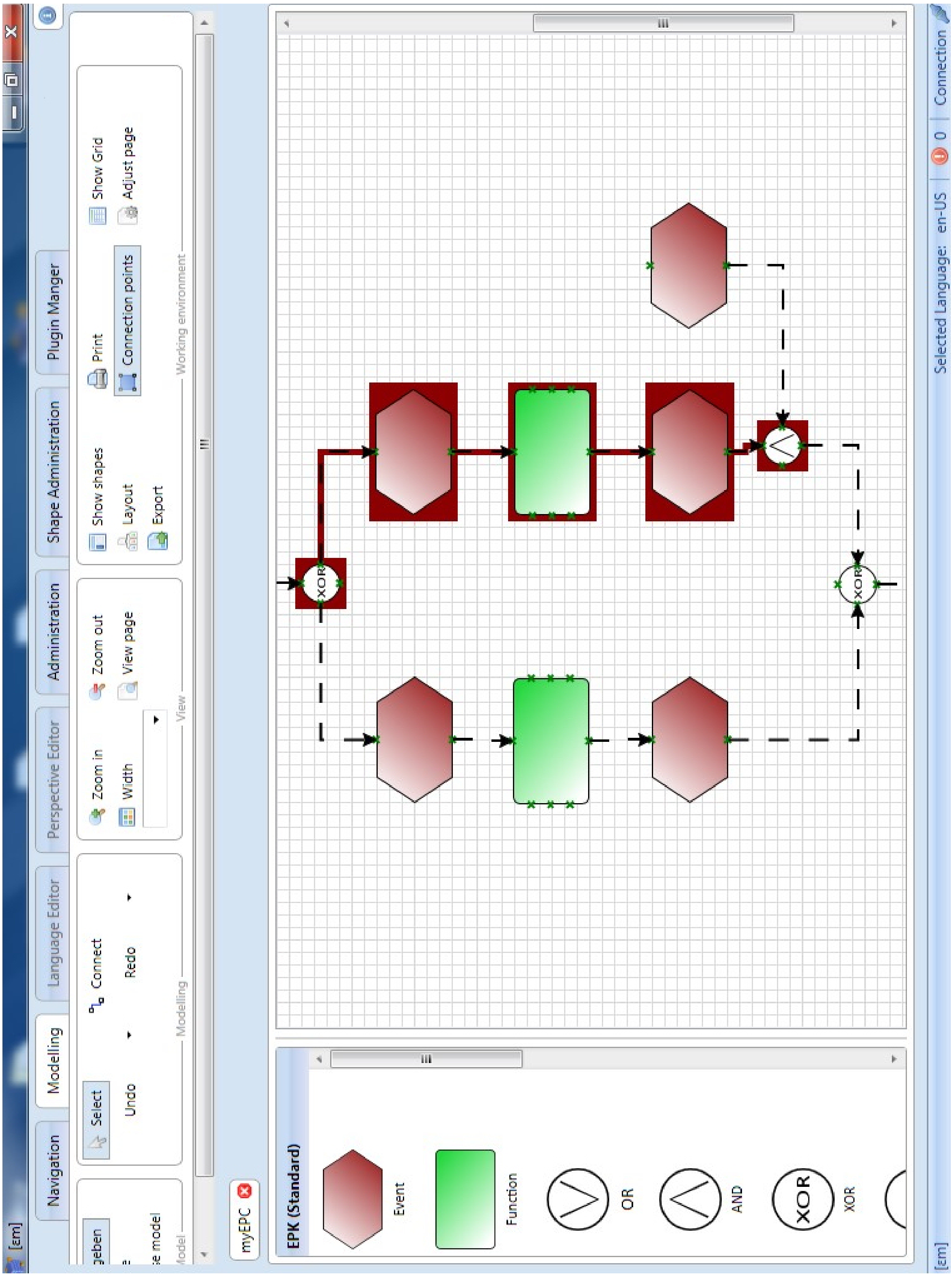

Figure 22: Highlighting a pattern match (cf. Delfmann et al. 2015)

"activity" by the statements *ElementsOfType*. Finally, the checking approach searches for all directed paths from all "check" activities to all "verify" activities using the statement *DirectedPaths*.

In the corresponding modelling tool the user can specify a pattern using the statements outlined above in a pattern editor (cf. Figure 21). The pattern statements, the basic sets containing the nodes and edges of the models, and the element types of the currently used modelling language are provided as select lists. Variables, variable equations (e.g., "A=B", cf. Figures 3-4 and 13-16) and labels can be defined and combined with pattern statements. As soon as a pattern is applied to a model, every pattern occurrence is highlighted (cf. Figure 22). The user can browse the different matches.

4.2 Results of Compliance Checking

The process model we checked against the patterns represents core business processes of banks and includes sub-processes dealing with consulting of customers, selling of financial products, archiving, and payment transactions. It was modelled using the process modelling language *icebricks* (cf. Section 3.2) and contains more than 2000 nodes. The fact that the process model was modelled with a non-standard language was no problem as the compliance checking approach we applied was independent of modelling languages. The modelling language of the process model comes with a terminological standardisation of labels, so we could assume that the model was unambiguous (cf. again Section 3.2). The catalogue of valid terms (including composite terms!) the modelling language was based upon was also known, so we could use the provided terms for the specification of the patterns. This way, we could expect that the terms used in the patterns were also used in

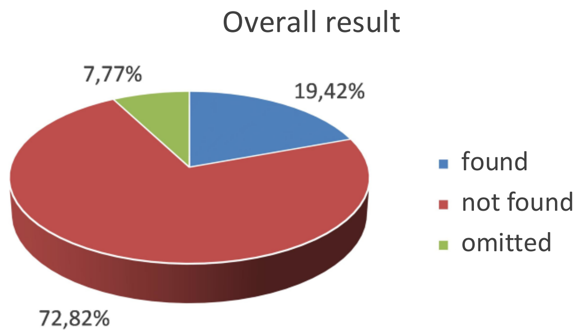

Figure 23: Overall compliance checking results

the models, so we would not encounter any problems related to term mismatches.

We checked the process model against all the compliance patterns indicating a compliant model section we could identify from the legal texts. The patterns were adjusted to the standardized terminology used by the modelling language. The valid terms were taken from the glossary the modelling language provided. We transformed the patterns into pattern queries of the pattern matching approach shortly introduced above. The results are shown in Figures 23-25. For some patterns, we could find occurrences in the model, some were not found, and, furthermore, some patterns had to be omitted. The omitted patterns represented structures purely occurring in organizational charts (not structures in process models accessing organizational units!). As we could only analyse a process model, it did not make sense to search for organizational chart structures. In the figures, we provide the percentage of patterns of which we could find occurrences in the models, the percentage of patterns of which we could not find any occurrences, and percentage of patterns we excluded from the search.

The results show that the percentage of patterns for which we could find occurrences in the process model is quite low. However, we did expect a low hit rate, due to the following reasons: First, we did not analyse the detailed context of the process model beforehand, so

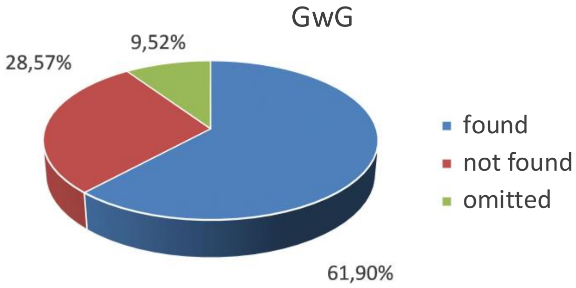

Figure 24: GwG compliance checking results

we did not know which of the compliance rules were relevant for the process at all. Second, we did not know the "degree of compliance" of the process. Hence, against the backdrop of these "tough" test conditions, the number of found pattern occurrences appears rather high.

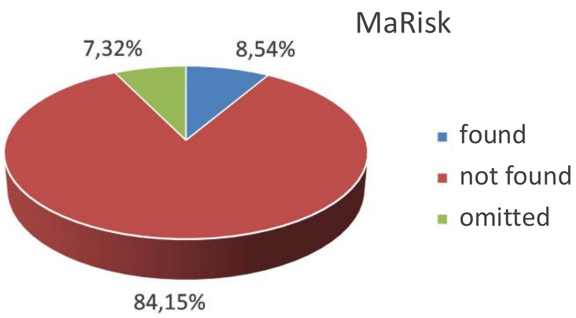

Figure 25: MaRisk compliance checking results

Regarding the relevance of the results, we have to distinguish two different views on precision and recall of GMQL and the patterns. As GMQL is based on the graph theoretical concepts of subgraph isomorphism and subgraph homeomorphism, it returns – by definition – all subsections of an analysed model that match the given pattern in terms of structure and label contents. Hence, regarding structure and label contents only, both precision and recall are always 100

Concerning the question whether a returned subsection is really a compliance fulfilment/ violation strongly depends on the compliance checking situation, hence the particular company, the process context, and the compliance rule depicted in the pattern. Hence, a returned result must always be reviewed by a compliance expert who decides whether or not the result really is a compliance fulfilment/ violation. At a first glance, inevitable manual review seems to question the automatic compliance checking approach. However, we argue that the pure process of searching for potential compliance fulfilments/violations can be considerably streamlined. Furthermore, it must be questioned if it is possible at all to fully automate compliance checking. Hence, although we did not perform a precision/recall evaluation concerning the question whether or not a found pattern match is really a compliance fulfilment/violation or vice versa, we argue that even if GMQL and the patterns would provide only a very low precision/recall, we would still benefit. However, former studies have already shown that using compliance checking and compliance patterns exhibit very satisfactory precision and recall (cf. Becker et al. 2014).

The results of the pattern matching experiment suggest that extracting regulations from legal texts, transforming them into compliance patterns and searching for them in conceptual models (even in models of which we do not know the contents and their compliance context in detail) can be useful, as occurrences of common compliance patterns indeed exist in such models and can be found through pattern matching. We can assume that these pattern occurrences indicate model sections that indeed comply with a legal regulation or – in the case of anti-patterns – contain a compliance violation. Identifying such model sections automatically may reduce the sheer effort of searching for them, and hence support compliance officers in their everyday work. If we could provide a common catalogue of compliance patterns, we could further reduce the effort of arranging a list of regulations valid for a company and their subsequent transformation into patterns, because we could reuse existing knowledge.

5 Discussion and Outlook

In this paper, we have identified several graph-like compliance patterns to identify sections in conceptual models representing either compliant parts of a company's information system or part violating compliance rules. The patterns were derived from regulations we extracted from legal texts related to the financial industries. In particular, we analysed the German Money Laundering Act (Geldwäschegesetz, GwG) and the Minimum Requirements for Risk Management (MaRisk) (Lorenz 2008). The patterns we identified are supposed to be provided to compliance officers of financial institutions in the form of a common catalogue, so they can be reused for compliance checking. We see the benefit of such a catalogue in the fact that compliance officers using business process compliance checking to inspect their business processes can reuse the patterns as input for compliance checking. Building such a catalogue each time from scratch instead may be cumbersome, as compliance patterns have to be derived from legal or other regulatory texts first. We expect that a compliance pattern catalogue that was derived from common regulations can contribute to decrease such effort, as we can reuse existing knowledge.

With our pattern matching experiment, we could show that patterns derived from common legal texts indeed can be found in process models (in this case, in an arbitrarily chosen process model form the financial industry!). We could observe several pattern occurrences, although we did not align the compliance context of the process model and the context of the compliance patterns. This result shows that there exist sections in process models that correspond with rules derived from common legal texts. For the construction of a company-specific compliance pattern catalogue, such common patterns can be used as a starting point.

In this paper, we developed a first beta-version of a common compliance pattern catalogue for the financial industry. Although we can expect that a catalogue can provide value for compliance management, the results of this paper are subject to limitations: First, in fact we derived compliance patterns to identify both compliant parts of a conceptual model and those parts violating a compliance rule. However, in our pattern matching experiment, we applied only the former ones. Hence, in further experiments, we will also apply the anti-patterns to estimate their usefulness in compliance checking. Second, we restricted our analysis to the financial sector. Future analyses will extend pattern construction to further business domains. Third, we restricted our analysis to regulations coming from the German speaking area. It is very likely that regulations of other countries can be transformed into patterns in a similar way. Hence, future work will also address the internationalisation of the catalogue to make companies from outside of middle Europe benefit from compliance pattern catalogues, too. Fourth, we do not yet know if the way how we represent compliance patterns is sufficient for all thinkable compliance patterns. This is to be checked in the context of further compliance rule constructions. However, as the way how we represent compliance patterns is not bound to any formal syntax, we expect that it can be extended easily. Fifth, we did not analyse for what kinds of business processes what kind of compliance pattern was relevant. Sixth, we did not include compliance experts from practice in our analysis. These both aspects will also be subject of future research and will, seventh, help us estimating precision and recall of compliance patterns.

Acknowledgements

This article was partly developed in the course of the research project "Business Process Compliance Management in the Financial Sector", which was funded by the German Research Association from September 2012 to August 2014 (reference number: DE1983/1-1).

References

Abdullah S N, Sadiq S, Indulska M (2010) Emerging Challenges in Information Systems Research for Regulatory Compliance Management. In: Proceedings of the CAISE. Hammamet, pp. 251-265

Accorsi R, Lowis D I L, Sato Y (2011) Automated certification for compliant cloud-based business processes. Business & Information Systems Engineering, 3(3), pp. 145-154

Arbab F, Kokash N, Meng S (2009) Towards Using Reo for Compliance-Aware Business Process Modeling. In: Tiziana M, S. Bernhard S. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Springer, Berlin, pp. 108-123

Awad A (2007) BPMN-Q: A language to query business processes. In: Reichert M, Strecker S, Turowski K (eds.) Proceedings of the EMISA Conference. St. Goar, pp. 115-128

Awad A (2010) A compliance management framework for business process models. Dissertation. University of Potsdam

Awad A, Decker G, Weske M (2008) Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In: Dumas M, Reichert M, Shan M-C (eds.) Business Process Management. Springer, Berlin, pp. 326-341

Awad A, Smirnov S, Weske M (2009a) Resolution of Compliance Violation in Business Process Models: A Planning-Based Approach. In: Meersman R, Dillon T, Herrero P (eds.) On the Move to Meaningful Internet Systems: OTM 2009. Springer, Berlin, pp. 6-23

Awad A, Smirnov S, Weske M (2009b) Towards Resolving Compliance Violations in Business Process Models. In: Proceedings of the International Workshop on Governance, Risk and Compliance, pp. 18-33

Awad A, Weske M (2009) Visualization of Compliance Violation Using Antipattern. In: Proceedings of the 5th Workshop on Business Process Intelligence (BPI). Ulm

Awad A, Weske M (2010) Visualization of Compliance Violation in Business Process Models. In: zur Mühlen M, Su J (eds.) Business Process Management Workshops, Hoboken, pp. 182-193

Awad A., Weidlich M,. Weske M (2011) Visually specifying compliance rules and explaining their violations for business processes. Journal of Visual Languages & Computing 22(1), pp. 30-55

Becker J, Algermissen L, Pfeiffer D, Räckers M (2007) Bausteinbasierte Modellierung von Prozesslandschaften mit der PICTURE-Methode am Beispiel der Universitätsverwaltung Münster. Wirtschaftsinformatik 49(4), pp. 267-279

Becker J, Bergener P, Delfmann P, WeißB (2011) Modeling and Checking Business Process Compliance Rules in the Financial Sector. In: Proceedings of the 32nd International Conference on Information Systems (ICIS 2011). Shanghai

Becker J, Clever N, Holler J, Shitkova M (2013) Icebricks. In: vom Brocke J, Ram S, Rossi M (eds.) Design Science at the Intersection of Physical and Virtual Design. 8th International Conference, DESRIST 2013, Helsinki, Finland, June 11-12,2013, Proceedings. Berlin, pp. 394-399

Becker J, Delfmann P, Dietrich H-A, Eggert M, Steinhorst M (2014) Model-based Business Process Compliance Checking in Financial Industries - Conceptualization, Implementation, and Evaluation. Information Systems Frontiers. Accepted for publication on 2014-08-15. DOI: 10.1007/s10796-014-9529-y.

Becker J, Delfmann P, Eggert M, Schwittay S (2012) Generalizability and Applicability of Model-based Business Process Compliance Checking Approaches – A State-of-the-Art Analysis and Research Roadmap. Business Research 5(2), pp. 221-247

Damaggio E, Deutsch A, Hull R, Vianu V (2011) Automatic Verification of Data-Centric Business Processes. In: Rinderle-Ma S, Toumani F, Wolf K (eds.) BPM 2011. Springer, Berlin, pp. 3-16

Delfmann P, Herwig S, Lis L (2009) Unified Enterprise Knowledge Representation with Conceptual Models – Capturing Corporate Language in Naming Conventions. In: Proceedings of the 30th International Conference on Information Systems (ICIS 2009). Phoenix

Delfmann P, Hübers M (2014) Entwicklung eines Katalogs von Regulationsmustern zur Unterstützung der Compliance-Überprüfung von Geschäftsprozessen im Finanzsektor. Proceedings des Workshops Dienstleistungsmodellierung (DLM) im Rahmen der Tagung Modellierung. Wien

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), pp. 129-177.

El Kharbili M, de Medeiros A, Stein S, van der Aalst WMP (2008) Business Process Compliance Checking: Current State and Future Challenges. Lecture Notes in Informatics (141), pp. 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: Maglio P, Weske M, Yang J, Fantinato M (eds.) Service-Oriented Computing. Springer, Berlin, pp. 17-31

Eshuis R (2006) Symbolic Model Checking of UML Activity Diagrams. ACM Transactions on Software Engineering and Methodology, 15(1), pp. 1-38

Eshuis R, Wieringa R (2004) Tool Support for Verifying UML Activity Diagrams. IEEE Transactions on Software Engineering, 30(7), pp. 437-447

Förster A, Engels G, Schattkowsky T (2005) Activity Diagram Patterns for Modeling Quality Constraints in Business Processes. In: Briand L, Williams C (eds.) Model Driven Engineering Languages and Systems. Springer, Berlin, pp. 2-16

Förster A, Engels G, Schattkowsky T, Van Der Straeten R (2007) Verification of Business Process Quality Constraints Based on Visual Process Patterns. In: First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE '07), pp. 197-208

Ghanavati S, Amyot D, Peyton L (2011) A systematic review of goal-oriented requirements management frameworks for business process compliance. In: 2011 Fourth International Workshop on Requirements Engineering and Law (RELAW), pp. 25-34

Ghose A, Koliadis G (2007) Auditing Business Process Compliance. In: Krämer BJ, Lin K-J, Narasimhan P (eds.) Service-Oriented Computing (ICSOC 2007). Springer, Vienna, pp. 169-180

Goedertier S, Vanthienen J (2006) Designing Compliant Business Processes with Obligations and Permissions. In: Eder J, Dustdar

S (eds.) Business Process Management Workshops, LNCS 4103. Springer, Berlin, pp. 5-14

Governatori G, Milosevic Z (2006) A Formal Analysis of a Business Contract Language. International Journal of Cooperative Information Systems, 15(4), pp. 659-685

Governatori G, Milosevic Z, Sadiq S (2006) Compliance checking between business processes and business contracts. In: Proceedings of the 10th IEEE International Enterprise Distributed Object Computer Conference, pp. 221-232

Governatori G, Rotolo A (2010) A conceptually rich model of business process compliance. In: Link S, Ghose AK (eds.) Proceedings of the 7th Asia-Pacific Conference on Conceptual Modelling (APCCM '10), Australian Computer Society. Darlinghurst, pp. 3-12

Hassan W, Logrippo L (2009) Governance requirements extraction model for legal compliance validation. In: 2nd International Workshop on Requirements Engineering and Law (RELAW'09), pp. 7-12

Hoffmann J, Weber I, Governatori G (2009) On compliance checking for clausal constraints in annotated process models. Information Systems Frontiers, 11(5), pp. 1-23

Karagiannis D (2008) A Business Process-Based Modelling Extension for Regulatory Compliance, In: Bichler M et al. (eds.): Multikonferenz Wirtschaftsinformatik, MKWI 2008. Berlin, pp. 1159-1173

Khaluf L, Gerth C, Engels G (2011) Pattern-Based Modeling and Formalizing of Business Process Quality Constraints. In: Mouratidis H, Rolland C (eds.) 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011). London, pp. 521-535

Knuplesch D, Ly LT, Rinderle-Ma S, Pfeifer H, Dadam P (2010) On Enabling Data-Aware Compliance Checking of Business Process Models. In: Parsons J, Saeki M, Shoval P, Woo C, Wand Y (eds.) Proceedings of the 29th International Conference on Conceptual Modeling – ER 2010. Springer, Berlin, pp. 332-346

Knuplesch D, Reichert M, Ly LT, Kumar A, Rinderle-Ma S (2013) Visual Modeling of Business Process Compliance Rules with the Support of Multiple Perspectives. In: Conceptual Modeling. Lecture Notes in Computer Science. Springer, Berlin, pp. 106-120

Kokash N, Arbab F (2009) Formal Behavioral Modeling and Compliance Analysis for Service-Oriented Systems. In: Boer FS, Bonsangue MM, Madelaine E (eds.) Formal Methods for Components and Objects. Springer: Berlin, pp. 21-41

Küster JM, Ryndina K, Gall H (2007) Generation of Business Process Models for Object Life Cycle Compliance. In: Alonso G, Dadam P, Rosemann M (eds.) Business Process Management. Springer, Berlin, pp. 165-181

Kumar A, Liu R (2008) A Rule-Based Framework Using Role Patterns for Business Process Compliance. In: Bassiliades N, Governatori G, Paschke A (eds.) Rule Representation, Interchange and Reasoning on the Web. Springer, Berlin, pp. 58-72

Liu Y, Müller S, Xu K (2007) A static compliance-checking framework for business process models. IBM Systems Journal, 46(2), pp. 335-361

Lohmann N, Wolf K (2010) How to Implement a Theory of Correctness in the Area of Business Processes and Services. In: Hull R, Mendling J, Tai S (eds.) Proceedings of the 8th International Conference on Business Process Management (BPM 2010). Hoboken, pp. 61-77

Lorenz M (2008) Einführung in die rechtlichen Grundlagen des Risikomanagements. In: Romeike F (ed.) Rechtliche Grundlagen des Risikomanagements. Haftungs- und Strafvermeidung für Corporate Compliance. Erich Schmidt Verlag, Berlin, pp. 3-30

Lu R, Sadiq S, Governatori G (2008a) Compliance Aware Business Process Design. In: ter Hofstede A, Benatallah B, Paik H-Y (eds.) Business Process Management Workshops. Springer, Berlin, pp. 120-131

Lu R, Sadiq S, Governatori G (2008b) Measurement of Compliance Distance in Business Processes. Information Systems Management, 25(4), pp. 344-355

Ly LT, Göser K, Rinderle-Ma S, Dadam P (2006) Semantic Correctness in Adaptive Process Management Systems. In: Dustdar S, Fiadeiro JL, Sheth A (eds.) BPM 2006, LNCS 4102. Springer, Berlin, pp. 193-208

Ly LT, Göser K, Rinderle-Ma S, Dadam P (2008) Compliance of Semantic Constraints – A Requirements Analysis for Process Management Systems. In: Sadiq S, Indulska M, zur Mühlen M, Franch X, Hunt E, Coletta R (eds.) Proceedings of the International Workshop on Governance, Risk and Compliance-Applications in Information Systems. Montpellier, pp. 31-45

Ly LT, Göser K, Rinderle-Ma S, Dadam P (2008) Integration and verification of semantic constraints in adaptive process management systems. Data & Knowledge Engineering, 64(1), pp. 3-23

Ly LT, Göser K, Rinderle-Ma S, Dadam P (2010) Design and Verification of Instantiable Compliance Rule Graphs in Process-Aware Information Systems. In: Pernici B (ed.) Advanced Information Systems Engineering, LNCS. Springer, Berlin, pp. 9-23

Ly LT, Rinderle-Ma S, Göser K, Dadam P (2009) On enabling integrated process compliance with semantic constraints in process management systems Requirements, challenges, solutions, Information Systems Frontiers 14(2), pp. 195-219

Maxwell JC, Anton AI (2009) Checking existing requirements for compliance with law using a production rule model. In: 2nd International Workshop on Requirements Engineering and Law (RELAW'09), pp. 1-6

Monakova G, Kopp O, Leymann F, Moser S, Schäfers K (2009) Verifying Business Rules Using an SMT Solver for BPEL Processes. In: Abramowicz W, Maciaszek LA, Kowalczyk R, Speck A (eds.) Proceedings of the Business Process and Services Computing Conference: BPSC'09. Leipzig, pp. 81-94

Müller J (2010) Strukturbasierte Verifikation von BPMN-Modellen. Dissertation. University of Tübingen

Sadiq S, Governatori G, Namiri K (2007) Modeling Control Objectives for Business Process Compliance. In: Alonso G, Dadam P, Rosemann M (eds.) Business Process Management. Springer, Berlin, pp. 149-164

Schleicher D, Anstett T, Leymann F, Schumm D (2010) Compliant Business Process Design Using Refinement Layers. In: Meersmann R, Dillon T, Herrero P (eds.) On the Move to Meaningful Internet Systems: OTM 2010. Springer, Berlin, pp. 114-131

Schumm D, Turetken O, Kokash N, Elgammal A, Leymann F, van den Heuvel W-J (2010) Business Process Compliance through Reusable Units of Compliant Processes. In:

Daniel F, Facca FM (eds.) Current Trends in Web Engineering. Springer, Berlin, pp. 325-237

Schwab, M (2012) Process-based compliance: Probabilities. In: Proceedings of the 6th International Conference on Research Challenges in Information Science (RCIS).

Siena A, Jureta I, Ingolfo S, Susi A, Perini A, Mylopoulos J (2012) Capturing Variability of Law with Nómos 2. In: Conceptual Modeling. Lecture Notes in Computer Science. Springer, Berlin, pp. 383-396

Thomas O, Fellmann M (2009) Semantic Process Modeling – Design and Implementation of an Ontology-Based Representation of Business Processes. Business & Information Systems Engineering 1(6), pp. 438-451

Trčka N, van der Aalst WMP, Sidorova N (2009) Data-Flow Anti-patterns: Discovering Data-Flow Errors in Workflows. In: van Eck P, Gordijn J, Wieringa R (eds.) Proceedings of the International Conference on Advanced Information Systems Engineering (CAISE 2009). Amsterdam, pp. 425-439

Wang HJ, Zhao JL (2011) Constraint-centric workflow change analytics. Decision Support Systems 51, pp. 562-575

Wörzberger R, Kurpick T, Heer T (2008a) Checking Correctness and Compliance of Integrated Process Models. In: Negru V, Jebelean T, Petcu D, Zaharie D (eds.) 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Timisoara, pp. 576-583

Wörzberger R, Kurpick T, Heer T (2008b) On Correctness, Compliance and Consistency of Process Models. In: 17th IEEE International Workshops on Enabling Technologies (WETICE '08). Rome, pp. 251-252

Wolter C, Meinel C (2010) An approach to capture authorisation requirements in business processes. Requirements Engineering, 15(4), pp. 359-373

Wolter C, Miseldine P, Meinel C (2009) Verification of Business Process Entailment Constraints Using SPIN. In: Massacci F, Redwine ST, Zannone N (eds.) Engineering Secure Software and Systems. Springer, Berlin, pp. 1-15

Xiangpeng Z, Cerone A, Krishnan P (2006) Verifying BPEL Workflows Under Authorisation Constraints. In: Dustdar S, Fiadeiro JL, Sheth AP (eds.) Proceedings of the International Conference on Business Process Management (BPM 2006). Vienna, pp. 439-444

Patrick Delfmann
University of Münster, Department of Information Systems,Leonardo-Campus 3, 48149/Münster
patrick.delfmann@ercis.uni-muenster.de

Michael Hübers
University of Münster, Department of Information Systems,Leonardo-Campus 3, 48149/Münster