Institute of Architecture of Application Systems

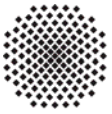
Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify Compliance Requirements

Daniel Schleicher, Frank Leymann, David Schumm, Monika Weidmann

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{schleicher, schumm, leymann}@iaas.uni-stuttgart.de
monika.weidmann@iao.fraunhofer.de

BibTeX:

```
@inproceedings {INPROC-2010-93,
   author = {Daniel Schleicher and Monika Weidmann and Frank Leymann and David
Schumm},
   title = {{Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify
Compliance Requirements}},
   booktitle = {Proceedings of SOCA 2010},
   publisher = {IEEE Computer Society},
   institution = {Universit{\"a}t Stuttgart, Fakult{\"a}t Informatik,
Elektrotechnik und Informationstechnik, Germany},
   pages = {1--18},
   month = {Dezember},
   year = {2010},
   keywords = {BPMN; Compliance; Workflow Management},
   language = {Englisch},
   cr-category = {H.4.1 Office Automation}
}
```

Universität Stuttgart
Germany

Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify Compliance Requirements

Daniel Schleicher, Frank Leymann,
David Schumm
Institute of Architecture of Application Systems
University of Stuttgart
Stuttgart, Germany
lastname@iaas.uni-stuttgart.de

Monika Weidmann
Fraunhofer Institute for Industrial Engineering IAO
Stuttgart, Germany
monika.weidmann@iao.fraunhofer.de

***Abstract*—Compliance of business processes is becoming increasingly important in the domain of business process design. Despite that, human process designers must be able to concentrate on the business goals which a business process needs to fulfil. Compliance aspects of the business process should not be in the main focus of the human process designer during the development phase. Therefore, tools must support human process designers in developing compliant business processes. In this paper we introduce the concept of *compliance scopes*. Compliance scopes are areas in a business process where certain compliance conditions must hold. These conditions are attached to the compliance scopes.**

Compliance scopes can be applied to existing business process models as well as to process templates. In this way compliance rules are applied to certain areas of a business process. During design time, compliance scopes can be used in graphical workbenches to evaluate modifications to business processes.

***Keywords*-compliance; business process design;**

I. Introduction

Compliance is becoming an important factor for the business operations of enterprises these days. Regulations are either imposed by governments or internal departments responsible to prevent economical or legal harm. There exist a number of bodies of regulations, for example SOX [1] or BASEL II [2]. As these tend to be huge in size it is rather impossible for a single human to keep all these regulations in mind, especially when designing a business process. Additionally, enterprises are facing considerable fines when they violate commercial regulations [3]. These findings led us to the conclusion that enterprises need to be sure that the business processes they are creating are compliant. As a consequence it follows that there is a need for theoretical concepts and tools to support human business process designers to create compliant business processes. Human business process designers should not have to deal with compliance during the development phase of a business process.

Based on our experience with insurance companies, we present solutions for compliant business process design in this paper. The goal of these insurance companies is to minimise costs of their business processes. One possibility to achieve this is to integrate the information systems of their partners into their business processes. Therefore, so called *process templates* can be handed to the partner enterprises in order to be completed by them. A process template is a business process skeleton that is not meant to be executed. This process template comprises the main activities that must be executed by a partner to interact with the insurance business process of the insurance company. It must be completed with partner-specific activities. Since it is important for the insurance company to run compliant business processes, the partner enterprises have to be guided in completing the business process templates to create compliant business processes. In this paper we provide a means to guide the partner enterprises in completing process templates to become compliant business processes.

The main contribution of this paper is a theoretical concept that is called *compliance scope*. Compliance scopes are used to restrict certain areas of a business process to prevent modifications that would cause non-compliance of the business process. Further contributions are a method explaining how compliance scopes can be used within enterprises, and scenarios showing where compliance scopes are useful. In order to show the practicality of the concepts presented in this paper we decided to show and apply them to business processes modelled in BPMN 2.0. However, the concept of a compliance scope can be applied to an arbitrary business process language because of its generic nature.

The remainder of this paper is structured as follows. In Section II we analyse related work that has been done in the direction of compliant business process design and show advantages and shortcomings that we address in this paper. Section III shows how the problems of compliant business process design became apparent to us. Here we describe a motivational example by means of an insurance case that we came across during our work in the *openXchange*[1]

[1]http://www.openxchange-project.de

project. Section IV provides a detailed specification of the concept of a compliance scope. In this section we extend the meta model of the Business Process Model and Notation (BPMN 2.0) [4] and enrich it with compliance scopes. We therefore use BPMN 2.0 in contrast to BPMN 1.1 because it has an extension mechanism that has been refined and enhanced. A method showing how compliance scopes are applied is presented in Section V. Future work is discussed in Section VI. Finally, Section VII concludes this paper.

II. Related Work

In this section we present related work in the field of compliant business process design. We discuss advantages and shortcomings with an eye to the goals of this paper. The related work is categorised in two aspects. The first part of the works presented here are coming from the domain of compliance engineering. The second part describes ways to build variants of business processes. Seeking a combination of these areas, we look at them regarding design-time aspects.

In [5] an approach is presented that uses process fragments to be added to a process. These fragments implement certain compliance requirements the process has to meet. After the insertion of a process fragment into a process, verification tools perform a check to ensure the process fragment was inserted at the right place. The ability to define compliance rules for certain areas of a process is missing in this work. From our experience we see that many compliance rules must only be applied to certain areas of a business process. For validation purposes only the business process activities have to be considered that are contained within a certain compliance scope. Here, we see potential to accelerate the verification of compliance rules during design time.

A different approach to design compliant business processes is presented in [6]. Here, compliance requirements are expressed with statements in deontic logic. These statements are automatically transformed into a graphical process model. This process model can then be used by a human process designer to verify the validity of the process being developed. In this approach the human process designer is supported by tools in developing a compliant process. These tools show a possible compliant control flow but do not automatically check the resulting process model at design time. This would be desirable since human process designers are already challenged by reaching the complex business goal of the new business process during design time. They should not have to bother with additional requirements like those coming from compliance considerations.

In [7] business processes are enriched with control tags describing compliance requirements. These control tags help human business process designers to design compliant business processes. The use of analysis tools to check compliance of a business process model is encouraged. Apart from this, no statements are made on concepts describing how to check the compliance of business process models during design time.

Compliance of a BPEL process is checked by using language patterns in [8]. These language patterns combine a number of statements in a given logical language. The authors argue that these patterns are more understandable and easier to use than plain LTL for example. The compliance rules specified in this pattern language can be automatically checked against a BPEL process. With the concept of compliance scopes we make the definition of compliance rules another step easier. With compliance scopes compliance rules can be applied to certain areas of a business process. This leads to shorter and less complex compliance rules. During the creation of a compliance rule for a compliance scope it is not necessary to take the whole business process into account.

In the area of modelling process variants several approaches have been presented. In [9] a method is described which allows configurable process models to be defined preserving soundness during the configuration process. Compliance issues need to be addressed implicitly during the creation of the configurable process model.

A different approach called Provop is described in [10]. Here a basic process model is extended with options, which include possible modifications of the basic model. Although some research has been done in the field of guaranteeing soundness, compliance requirements could only be modelled implicitly through general process constraints.

Another method is called PESOA, and uses features similar to UML-concepts like stereotypes in order to model variability [11]. Here, the concept of *variation points* is introduced as an abstract activity, which is then implemented using concrete activities. Compliance has not been in the focus of this work.

In contrast to the mentioned methods, the variability descriptors described in [12] allow the addition of variability to the complete application and not only process models. The separation of models and variability, which can be applied to any xml-based file or model, allows the creation of a valid model according to any specification. The creation of compliant applications was not considered in this work.

To sum up, in previous work we see approaches dealing with compliance at design time and approaches dealing with the creation of process variants. In this paper, we combine compliance considerations with creation of process variants at design time. We further want to emphasise that our goal is to support human business process designers during design time. We want to provide concepts that are user friendly.

III. Motivating Case Study

As mentioned in Section I, compliance is an important factor in the design phase of business processes.

In our work with customers in the openXchange project the need for means to deal with compliance in business

processes became clear. The openXchange project deals with the management of property damage through a service network of involved small and medium sized enterprises. In this project we worked with concepts to design variants of damage claim processes. Process templates play an important role in this business. Variants of these process templates can be created for every special use case. This fosters reuse of existing knowledge and thus helps to reduce process costs. The partner companies and customers also profit from using these reference processes because their damage claims can be processed automatically, resulting in faster and defined resolution.

Process templates implement a number of compliance rules. Hence, these process templates are compliant to these compliance rules. Building variants from these process templates raises the question if these variants are still compliant. Without concepts and tools to ensure the compliance of such process templates at design-time, we cannot be sure of the resulting business process meets the same compliance requirements as the process template.

Insurance companies and their partners need to conform to certain regulations. This also applies to the business processes they are using. During our work in the openXchange project, the following requirements became apparent.

- Enterprises want to reuse business processes
- Enterprises need to build compliant business processes
- Human process designers must be able to fully concentrate on the implementation of the business goals of a business process
- Tools must support human process designers in the design phase to prevent them of being distracted by compliance concerns
- Concepts, methods, and techniques are needed to enable tools like graphical process design workbenches supporting human process designers

In this section we show which concepts are needed for compliance management during design time in a real-world use case.

We assume an insurance company named *Business Insurance Group (BIG)* wants to build a number of process templates for different purposes which support the company in the creation of compliant business processes.

One process that is often used in the insurance industry is the claims management process. As the claims management process is slightly different for the different insurance products (like motor insurance, building insurance, or indemnity insurance) it is qualified to be reused by means of process templates. Process fragments are stored in a repository to be used to complete these process templates to executable business processes. All claims management processes basically consist of four steps (see Figure 1 for details):

- *Damage set-up* – the damage claim is filed and matched with contract data
- *Active claims management* – partners (service providers) are involved (e.g. surveyor, repair shop)
- *Claims Settlement* – calculation of the compensation and payments
- *Claims to damages* – claims to third parties are persecuted

For each of these steps different compliance rules apply, as all of them contain distinct sets of activities. In the current example we focus on the *damage set-up*.

In the mentioned process repository, besides process fragments, process templates for processing damage claims are stored. These process templates should be used by human process designers as a starting point for new business processes. Figure 2 shows a simplified process template. Only activities labelled *Region* can be filled by human process designers with additional logic, implementing the requirements for a particular domain like fire damage, for instance. No other modifications are allowed to be done by human process designers. The process template implicitly implements certain compliance rules. One rule is for example that the activity “check data and set-up damage report” must be executed before any other activity in the business process. These implicitly defined compliance rules must hold during the design phase and when the process is finished.

The purpose of a process template is to become a compliant business process in the end. To keep the process template compliant, modifications to the process template must be restricted. Such restrictions are realised as compliance rules. Only those modifications to a process template should be possible that do not introduce compliance violations into the process template.

We provide a formal definition of compliance scopes in the next section. This is done to facilitate the creation of tools to support human business process designers. These tools must be able to evaluate the compliance rules attached to compliance scopes at every design step.

IV. Compliance Scope

To accomplish the goal of supporting the human business process designer in designing compliant business processes, we introduce the concept of a *compliance scope*.

Figure 2 shows two possible compliance scopes as an example. These compliance scopes define areas in a process model where certain rules must hold. In compliance scope one such a rule could be: *The activity ”check data and set-up damage report” must be executed once before any other activity in the business process.* A possible compliance rule attached to compliance scope two could be: *Either activity ”check by compliance expert” or activity ”check by case officer” must be executed once before any activity that can be inserted in region three.*

These two rules imply that the mentioned activities cannot be used to complete the process template, because they are still present.

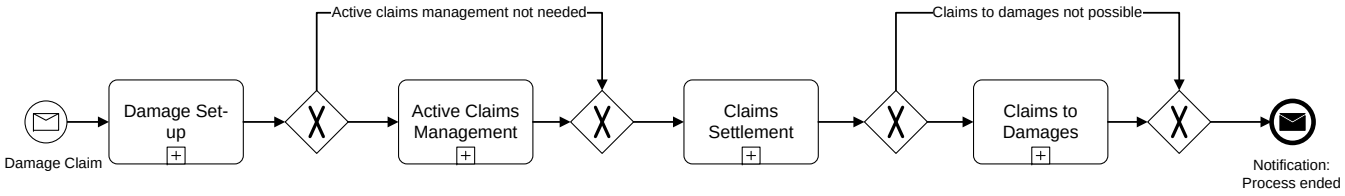


Figure 1. Claims management — top level process template in BPMN 2.0

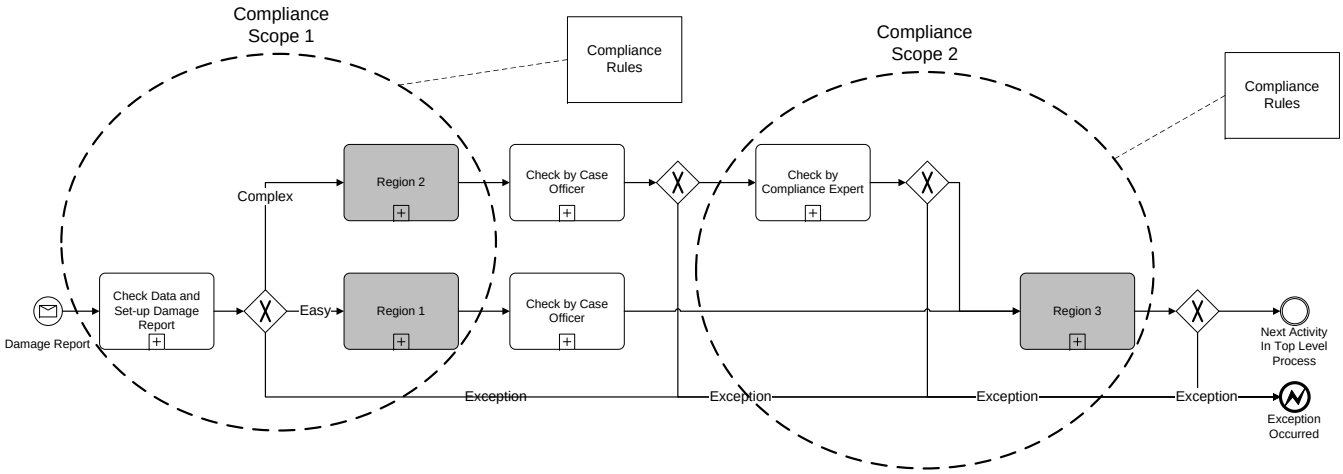


Figure 2. Damage set-up process template — simplified version in BPMN 2.0

Further implicit compliance rules implemented by the process template shown in Figure 2 are:

- *As the first step after the reception of a damage report, the damage report has to be set-up in the system once*
- *All damage reports need to be checked once for completeness by a case officer*
- *Complicated damage claims need to be examined once by a legal expert afterwards*
- *In the last region-activity an activity has to be inserted that does additional insurance and branch specific checks*

To prevent human business process designers from developing a non-compliant business process we need to enforce some rules. A compliance scope can contain an arbitrary number of activities of a business process, starting from one single activity up to all activities of a business process.

A distinct set of compliance rules is attached to each of these scopes. With this concept it is possible to define sets of rules over special parts of a process template. These sets of rules have to be valid within a certain compliance scope and can be validated at design time.

Compliance scopes can be applied to all four basic steps of claims management and to all different insurance branches. For example in the liability insurance, an additional liability check needs to be conducted in the damage set-up phase. Another example is the assignment of tasks to internal or external surveyors. If there are external partners involved, invoice management needs to be conducted. Another rule of that type would be that an immediate payment can only take place if the damage has been visited by an employee.

Finally, overall compliance rules can be applied depending on the insurance itself: an insurance A might have the general rule that payments over a certain limit need to be signed off by a first-line manager. Another rule for the complete scope would be that if other parties are involved, a claims to damage phase definitely needs to be run through once in the process. These are rules aiming at the presence of additional activities to be inserted during completion of a process template.

The concept of a compliance scope is independent of its visual appearance in a graphical business process modelling workbench. A compliance scope can be visualised as a circle around a number of tasks in a BPMN 2.0 business process model, for example.

A. Formal Definition

A compliance scope is a hyper edge in a graph-based business process model (like BPMN) comprising activities that are not necessarily connected. A set of compliance rules is attached to each compliance scope. This set of rules can be used by a graphical workbench to check if modifications to the current process model result in a non-compliant process model. Thus, this set of rules is not intended to be read by human process designers. The formal definition of a

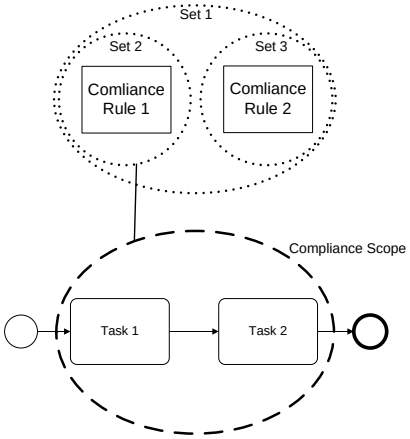


Figure 3. Illustration of equation 3

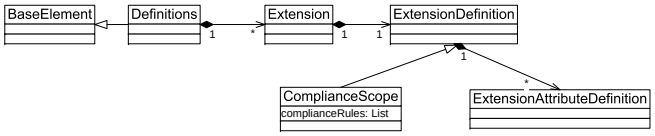


Figure 4. Part of BPMN 2.0 meta model designed in UML concerning extensibility (Source [4])

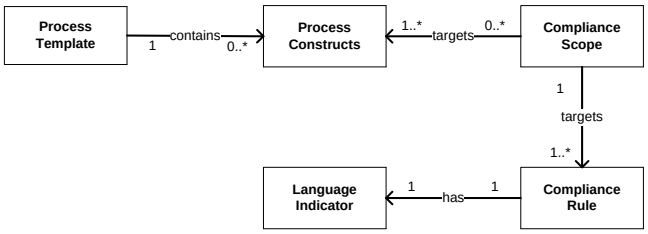


Figure 5. Relations of compliance scopes

compliance scope is based on the definition of a hyper graph [13].

Let $X = x_1, x_2, ..., x_n$ be a finite set of BPMN 2.0 tasks. Let E_i be a hyper edge in a hyper graph. A hyper graph on X is a family $H = E_1, E_2, ..., E_m$ of subsets of X such that

$$E_i \neq \varnothing \quad (i = 1, 2, ..., m) \tag{1}$$

and

$$\bigcup_{i=1}^{m} E_i = X. \tag{2}$$

Equation 1 defines that an edge in a hyper graph must connect more than zero elements in that graph. Equation 2 means that the union of all subsets of X must be equal to X.

Let C be a finite set of all compliance rules. Let CS be the finite set of compliance scopes applied to a business process, then

$$CS \subseteq E \times (2^C \setminus \{\ \}). \tag{3}$$

Equation 3 defines what a compliance scope is. It is a subset or equal to the cartesian product of all hyper edges and the power set of all compliance rules.

We illustrate that with an example, see Figure 3. We assume a BPMN 2.0 business process with two tasks. These two tasks are connected by a hyper edge. We assume that there are two compliance rules. So the power set of all compliance rules has four elements: the empty set, two sets with one compliance rule (Set 2 and Set 3), and one set with all compliance rules (Set 1). We subtract the empty set from that power set and get three resulting sets of rules. The empty set is discarded because a compliance scope without a compliance rule is useless. The hyper edge can now be connected to an element of the power set of the compliance rules making the hyper edge a compliance scope. In our example we connected Set 2 with the hyper edge.

B. BPMN 2.0 Extension Mechanism

BPMN 2.0 is the next incarnation of the Business Process Modelling Notation. The extension mechanism in BPMN 1.1 has been revised in BPMN 2.0. We use these mechanisms to integrate compliance scopes in BPMN 2.0.

Figure 4 shows a simplified excerpt of the BPMN 2.0 meta model, designed as an UML class diagram. The definitions class extends the *base element* class which provides the basis for all BPMN 2.0 elements. Each class with name *Definitions* is composed with one or more *Extension* classes.

To technically integrate compliance scopes in a BPMN 2.0 business process we extend the class *ExtensionDefinition*. We name this new class *ComplianceScope* (see Figure 4). Due to the fact that the class ComplianceScope inherits all attributes and methods from the class ExtensionDefinition it also inherits the extension attributes defined in the class ExtensionAttributeDefinition. To define which BPMN 2.0 tasks are contained within one compliance scope we use these extension attributes. Each extension attribute of the class ComplianceScope contains the identifier of a BPMN 2.0 task that is contained within this compliance scope. The class compliance scope further contains a property called *complianceRules*. This property holds the references to the compliance rules that are applied to the compliance scope.

C. Specifying Compliance Rules

There is a multitude of different languages and concepts that can be used to specify compliance rules [14]. Every language has its strengths and weaknesses. For example, ontologies are a good match to define causal dependencies. Others are better suited to model the temporal properties of a requirement, like Linear Temporal Logic (LTL).

We experienced in software projects, with industrial partners, that two classes of compliance problems exist within the domain of business processes. These two classes represent two logical fields that have to be covered by suitable

languages. The two classes are: *temporal execution order of activities (1)* and *presence or absence of activities (2)*.

To define rules that restrict the execution order of activities, Linear Temporal Logic (LTL) [15] is a suitable language. LTL has been used in a variety of works to model compliance requirements concerning the temporal order of activities. As an example compliance scope two in Figure 2 could have been annotated with the following rule: *If the activity "Check by Compliance Expert (Check)" is executed, then activity "Rate damage reporter (Rate)" must be inserted in region three*. A translation of this rule in LTL would look like this:

$$CheckFRate \tag{4}$$

In LTL F means *eventually (in the future)*. Thus, equation 4 means that a *rate* activity can only occur if a *check* activity has occurred before.

To define rules in field (2) we propose to use ontologies to restrict activities to be used in a process template. The idea is to match an ontology of a process fragment that is being inserted into a compliance scope to one or more ontologies being applied to that compliance scope.

V. Method

To show how the concept of compliance scopes is best used we provide a method. At first we introduce roles that must be present in this environment to enact the method. After that is done we describe the method by means of a BPMN diagram.

In the following list we provide a short description for every required role of the method:

- `Process Template Owner`: Coordinates the creation and modification of process templates equipped with compliance scopes.
- `Process Template Designer`:The process template designer designs process templates and uses compliance scopes along with compliance rules to restrict certain areas of a process template.
- `Compliance Expert`: The compliance expert has considerable knowledge in the field of compliance. Being responsible to provide all compliance rules that must be applied to a new business process, the compliance expert defines sets of patterns that implement certain compliance rules. The compliance expert also has the responsibility to keep up to date with changing and newly introduced legislations.
- `Process Designer (User)`: The human process designer who uses the process template is responsible to complete it to a executable business process.

Figure 6 shows the method by means of a BPMN diagram. The diagram shows the interactions of all participating roles and partners in this method. The partners involved are an enterprise acting as a *compliance as a services provider*, a consulting enterprise (*Compliance Consulting*) having knowledge in the field of compliance, and an enterprise acting as a *customer* for process templates. The goal of the compliance as a service provider is to sell process templates that are annotated with compliance scopes along with software tools to fill these process templates with business activities.

In Figure 6 we modelled the case when a new process template is needed. The BPMN process starts at the top left with the experience for the need for a new process template. After that the template designer begins to create a process template. In parallel the compliance expert begins to assess the compliance rules that must be applied to the new process template. After the process template is finished and the compliance assessment is done, the compliance rules are integrated into the process template by the compliance expert using compliance scopes. After the process template is complete, it is certified by the template owner. The certification is done to keep track of changes to the process template. An unauthorised modification to a process template can be discovered this way.

The process designer can now use the process template as a starting point to create a new business process. A number of activities are inserted into the process template to make it executable. After the process template is validated against the annotated compliance rules, it can be executed.

Another scenario in which compliance scopes can be applied to support compliant business process design is described in the following.

A. Imposing compliance constraints on an existing process

This scenario considers the existence of legacy processes which have to be made compliant to laws, regulations and other compliance sources that have been identified as being relevant for that specific process. There are two important stakeholders involved in this scenario. On the one hand, there is the compliance expert who is aware of the constraints the business process has to comply with. On the other hand there is the process owner, who designed the process in order to perform particular work, possibly not considering any compliance requirements yet. Figure 7 illustrates the different steps that have to be performed in this scenario.

We start with an existing process (a), which possibly does not yet consider compliance requirements. This existing process is fully specified and thus executable. In cooperation with the process owner, the compliance expert defines compliance scopes (b) around particular parts of the process which have to comply with particular compliance requirements. These compliance scopes help to prevent modifications to this process that would make it non-compliant. For example, one scope could be related to security (the upper branch in the process) and another one could be related to privacy (the lower branch). In the following step (c) the compliance scopes are being validated by corresponding tools (e.g. model checkers, custom analysis tools etc.). The result of this

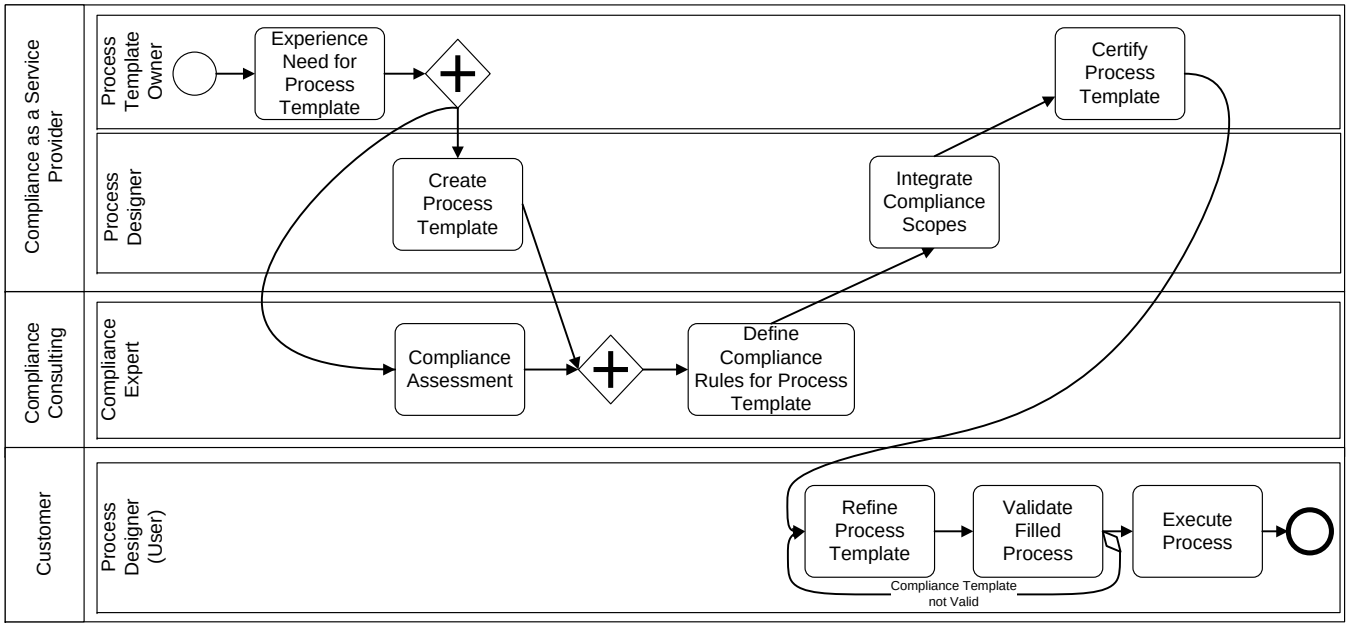


Figure 6. Method showing how to use compliance scopes

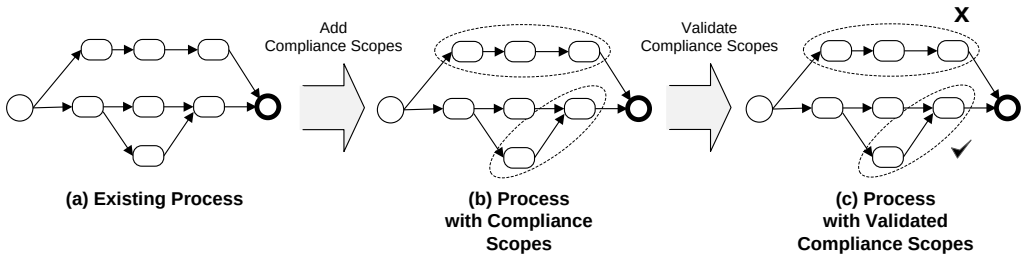


Figure 7. Method for Compliance Management in existing Processes

validation step is presented to both, the compliance expert and the process owner. In case of any violation the process owner has to redesign the process accordingly (in the example the security requirements are violated). After redesign, which has to be done by the process expert, not by the compliance expert, the validation step is executed again. When the validation results are positive, then the compliance expert has the certainty, that the constraints he imposed on the process are realised in an adequate manner.

VI. Future Work

In upcoming works we will analyse existing languages if they are suitable to be used with compliance scopes. Criteria for this analysis are the availability of tools to check constraints written in a certain language (i) and the expressiveness of a language (ii). Language that will be examined in this analysis are LTL, the object constraint language (OCL), or the formal contract language (FCL), besides others. We will also investigate which languages are best suited for different compliance problems.

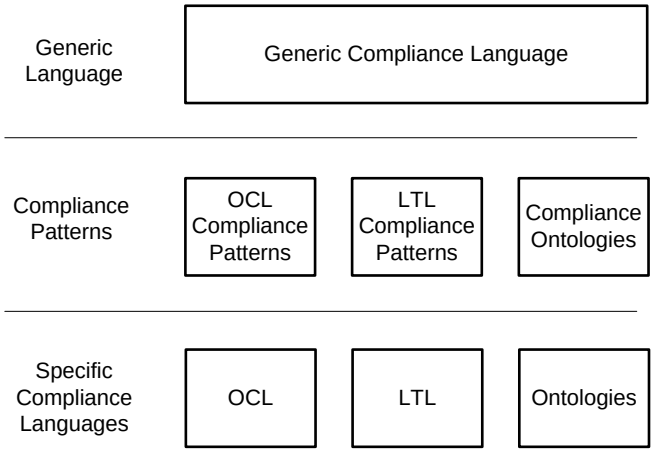


Figure 8. Compliance Patterns

We will further work on a solution to facilitate the definition of compliance constraints. To achieve this goal we will examine an approach that uses so called compliance

patterns [5]. A compliance pattern is a combination of statements in a specific logical language defining a compliance constraint. Figure 8 shows our pattern based approach. On the lowest level on the bottom of the figure possible languages are listed. One level higher in the middle of the figure you find patterns defined in each of these languages. On the highest layer we will define a generic compliance language. This generic compliance language contains descriptions of recurring compliance problems like the segregation of duties problem, for example. These descriptions are then linked to the compliance pattern written in a specific language that is most suitable for a specific compliance problem.

VII. Conclusion

In this paper we showed the concept of compliance scopes. Our claim is that human process designers should not bother with compliance during the design phase of a business process. Thus, this concept has been developed to support human process designers in creating compliant business processes. We showed that there is a need for such concepts by presenting a real-world use case. This use case has been designed using our experience with customers. In Section IV we formally defined the concept of a compliance scope and showed how the BPMN 2.0 meta model can be extended to use compliance scopes with BPMN 2.0. We further showed in this section how the BPMN 2.0 meta model can be extended to be able to create compliance scopes in BPMN 2.0 diagrams. We concluded this section with a discussion of logical languages to be used with compliance scopes. A method for compliance scopes was presented in Section V. This method describes which roles must be created and what infrastructure must be in place in order to work with compliance scopes in an enterprise. Here we also showed that the concept of compliance scopes can also be applied to existing business processes apart from process templates.

Acknowledgement

The work published in this article was partially funded by the MASTER project (http://www.master-fp7.eu; contract no. FP7-216917) and the COMPAS project (http://www.compas-ict.eu; contract no. FP7-215175) of the EU 7th Research Framework Programme Information and Communication Technologies Objective. It was further supported by the openXchange project of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ09011.

References

[1] United States Code, "Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745," Codified in Sections 11, 15, 18, 28, and 29 USC, July 2002.

[2] B. C. on Banking Supervision, "International Convergence of Capital Measurement and Capital Standards: A Revised Framework," 2006.

[3] P. Meller, "EU to beef up its cartel busters," http://www.nytimes.com/2005/03/10/business/worldbusiness/10iht-compete.html?_r=1&scp=7&sq=microsoft%20cartel&st=cse, 2005.

[4] Object Management Group (OMG), *Business Process Model and Notation (BPMN) Version 2.0*, OMG, 2010, oMG Document Number: dtc/2010-06-05.

[5] D. Schumm, O. Turetken, N. Kokash, A. Elgammal, F. Leymann, and W.-J. van den Heuvel, "Business Process Compliance through Reusable Units of Compliant Processes," in *Proceedings of the 1st Workshop on Engineering SOA and the Web (ESW'10)*. Springer, Juli 2010.

[6] S. Goedertier and J. Vanthienen, "Designing Compliant Business Processes from Obligations and Permissions," in *Business Process Management Workshops*, ser. Lecture Notes in Computer Science, J. Eder and S. Dustdar, Eds., vol. 4103. Springer Verlag, 2006, pp. 5–14.

[7] S. W. Sadiq, G. Governatori, and K. Namiri, "Modeling Control Objectives for Business Process Compliance," in *BPM*, 2007, pp. 149–164.

[8] J. Yu, T. P. Manh, J. Han, Y. Jin, Y. Han, and J. Wang, "Pattern based property specification and verification for service composition," in *In: Proceedings of 7th International Conference on Web Information Systems Engineering (WISE*, 2006, pp. 156–168.

[9] W. M. P. Aalst, M. Dumas, F. Gottschalk, A. H. M. Hofstede, M. L. Rosa, and J. Mendling, "Preserving correctness during business process model configuration," *Formal Aspects of Computing*, vol. 22, no. 3, pp. 459–482, Apr. 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/s00165-009-0112-0

[10] "Capturing variability in business process models: the provop approach," pp. n/a–n/a, 2009. [Online]. Available: http://www3.interscience.wiley.com/journal/122652902/abstract

[11] F. Puhlmann, A. Schnieders, J. Weiland, and M. Weske, "Variability Mechanisms for Process Models," DaimlerChrysler Research and Technology, Hasso-Plattner-Institut, PESOA-Report 17/2005, 2005.

[12] R. Mietzner and F. Leymann, "Generation of BPEL Customization Processes for SaaS Applications from Variability Descriptors," in *SCC '08: Proceedings of the 2008 IEEE International Conference on Services Computing*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 359–366.

[13] C. Berge, Ed., *Hypergraphs Combinatorics of Finite Sets*. Elsevier, 1989, vol. 45.

[14] European Project COMPAS, "State-of-the-art Report on the Existing Approaches to Improving Reusability of Processes and Service Compositions," 2008.

[15] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi, "On the temporal analysis of fairness," in *POPL '80: Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. New York, NY, USA: ACM, 1980, pp. 163–173.