Separating Compliance Management and Business Process Management

Elham Ramezani[1], Dirk Fahland*[2], Jan Martijn van der Werf**[2], and Peter Mattheis[1]

[1] Hochschule Furtwangen, Germany
(ramezani|Peter.Mattheis)@hs-furtwangen.de
[2] Eindhoven University of Technology, The Netherlands
(j.m.e.m.v.d.werf|d.fahland)@tue.nl

Abstract. The ever growing set of regulations and laws organizations have to comply to, introduces many new challenges. Current approaches that check for compliance by implementing controls in an existing information system (IS) decrease the maintainability of both the set of compliance rules and the IS. In this position paper, we advocate the separation of the compliance process from the organization's business processes. We introduce a life cycle for the management of compliance rules. A separate compliance engine is used to define and check compliance rules independent from the existing IS within an organization.

Key words: compliance management life cycle, compliance requirements, compliance rule, compliance checking

1 Introduction

Organizations are confronted with more and more regulations and laws to comply to. At a first glance, this seems rather as a burden for organizations. However, organizations see this as an opportunity to streamline their business and operations [10].

Compliance management (CM) within an organization comprises the design, implementation, maintenance, verification and reporting of compliance requirements originating in regulations and law enforcements. CM is closely related to risk management. Violating a compliance requirement introduces potential risks like consequences on management level, lost contracts with customers, service level agreements not been made, or non-identified security flaws [9, 13]. Therefore CM requires constant monitoring within organizations.

In the ideal situation, we would have a continuous auditing process that gives us real time insights into violation of business rules. Clearly this cannot be done manually. Therefore we need better techniques and software tools that make it possible to check arbitrary business rules automatically and in real time. Information systems (IS) play a major role in executing business processes either in cooperation with employees or

* supported by the European Community's Seventh Framework Programme FP7/2007- 2013 under grant agreement no 257593 (ACSI)

** supported by the PoSecCo project (project no. 257129), partially co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D (FP7)

autonomously. One of the approaches to enforce compliance in business operation is to embed *controls* in information systems, i.e., integration of compliance with BPM [3, 8]. However, implementing controls as tasks within an existing IS decreases the maintainability of both the IS as well as the set of compliance rules.

In this paper, we advocate to adapt the business process management (BPM) life cycle to manage compliance in a similar way. We propose to use a common business vocabulary based on BPM to specify compliance rules, and to separate the business operation from the process of compliance checking. Rather than inserting controls in the business process directly, we propose a specialized engine for CM that communicates with existing IS.

In the remainder of this position paper, we introduce the idea and the main concepts of a compliance management life cycle in Sect. 2 and discuss various aspects and open challenges in Sect. 3.

2 Compliance Management

The challenges posed by the need to implement compliance requirements in an organization call for a structured methodology. In this section, we make a step towards Compliance Management (CM), that is, a methodology to elicit, specify and formalize, implement, check and analyze, and optimize compliance requirements in organizations. We suppose the management of compliance requirements to follow a life cycle as sketched in Fig. 1.

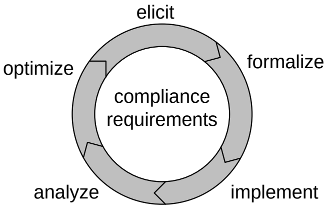

Fig. 1: Compliance Management Life Cycle

An initial life cycle for compliance has been proposed in [8]. In this paper, we take this idea one step further and *separate* CM from BPM. The key idea for separation is to introduce a separate *compliance engine* that is coupled with an existing information system (IS) to check its compliance, as sketched in Fig. 2.

In the following, we discuss each of the phases and at the same time introduce:

1. a business vocabulary for compliance rules similar to the basic notions of business process models in BPM,
2. a generic architecture of a compliance engine to implement compliance rules for checking compliance at a given IS, and
3. discuss techniques to check compliance.

Eliciting Compliance Requirements. In a rapidly evolving regulatory and compliance environment organizations are exposed constantly to different compliance sources [12]. The elicitation phase of the CM life cycle identifies the *compliance requirements* relevant for an organization by analyzing the profile of the organization including information such as company size, industry, region, and products or services.

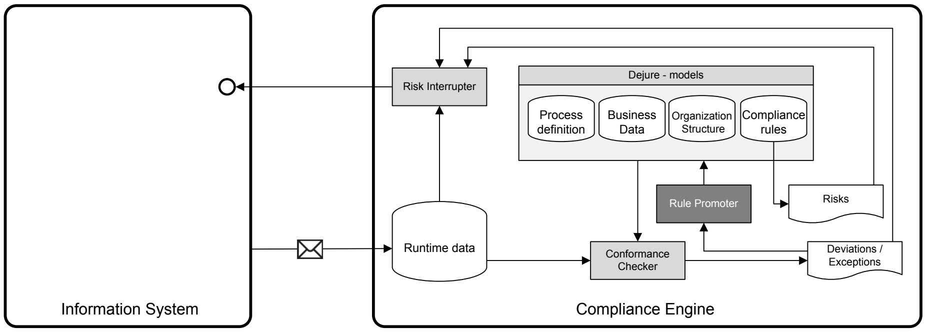

Fig. 2: Architecture of Compliance Engine

Specifying and Formalizing Compliance Rules The compliance requirements selected in the elicitation phase often originate in legal texts and have a very informal and abstract character. In order to be able to have tool support for compliance checking, these requirements should be represented in a formal and structured notation called *compliance rules*. The rules should focus on the business aspects of the requirements, rather than the technical aspects of the IS.

To guide this step and for maintainability, we propose to capture each aspect of a compliance requirement in a separate compliance rule based on a *business vocabulary*. This vocabulary builds on an abstract conceptual model of processes [3] which contains all primitive notions of business processes that are required to formulate compliance rules for processes in a precise manner. These primitives will then form the base vocabulary for writing compliance rules. Compliance requires distinguishing at least the following four primitives.

Process definition. A possibly hierarchical process consists of a set of *tasks* and subprocesses, which are usually *ordered* in some way.

Business data definition. Process data is represented by a *data model* consisting of a set of *entity types* and *relationships* between these entity types; each entity type defines a number of *attributes*. The actual data of the process is given by a number of *entities* for each type. Each entity assigns a *value* to its attributes, and is *associated* to other entities according to the relationships. Tasks are associated with entity types defining which attributes the task is allowed to read, write, or update.

Organizational definition. Each task is associated to a set of *roles*, e.g. a clerk or a manager, restricting who is allowed to execute that task. Roles may be ordered in a hierarchy. *Agents*, e.g., users or other systems, have a role assigned, which determines the tasks they are allowed to execute.

Runtime. A process is run by creating a new *case* (process instance); agents then execute tasks for a particular case. To execute a task in a case, an agent first gets assigned a role and a *permission* to execute the task, if permitted by its role. When the permission is granted, executing the task creates an *event* that records for which task and case the event was raised and by whom. Furthermore, it records which entities have been created or updated. The events are ordered by the moment in time they occurred.

These primitives pinpoint to the key elements of process definition and execution. Compliance rules are usually first stated in semi-formal sentences over these primitives; the sentences are then formalized, for instance in an appropriate logic, like in [8], in which the authors formalize compliance rules in a temporal object logic, or, as proposed in [3], using predicate logic.

Implementing Compliance Rules To ensure that an IS complies with a given requirement, its formalization (the formalized compliance rules) has to be implemented in a way that allows detecting if an execution violates some compliance rule. For this, the executions of the IS have to be observed and checked for violation of a rule.

At this point, the chosen business vocabulary of the compliance rules turns out crucial to CM. Each term in a compliance rule refers to information in the IS that needs to be observed and checked. The aforementioned business vocabulary allows for a generic architecture of a *compliance engine* that *extends* a given IS for checking compliance. For this, we propose to adapt the idea of an *online auditing tool* (OLAT) and a corresponding architecture [3] as shown in Fig. 2 to CM. In this way, the formalized compliance rules can be checked independently of the IS.

The engine assumes a *De Jure model* to be given, consisting of process models, data models, organizational models, and compliance rules, all formulated in the business vocabulary. The existing IS should send a message for every action it performs. This message is then recorded by the compliance engine in its *runtime data*. Hence, the runtime data comprises all information on process executions, that is, the current status and the history of the runtime primitives presented above. For instance, the events that have occurred, their order and duration, the values that were written by a particular event, the authorizations to access data granted to specific roles, or the role assignments given to specific agents.

In the *external* compliance checking setting, where the engine is separated from the IS, a *compliance checker* compares the De Jure models to the observed executions, i.e., the runtime data, and signals deviations or exceptions.

In the *internal* compliance checking setting, the engine is additionally allowed to control the IS by a *risk interrupter*. The risk interrupter takes as input the discovered deviations and assesses based on its information how severe the violation of the compliance rule would turn out in the future. In case of a severe risk, it can interrupt the process execution in the IS.

Checking and Analyzing Compliance Rules Having implemented compliance requirements as formalized compliance rules in a compliance engine allows checking compliance in an automated way. Thereby neither the proposed CM method, nor our architecture is tied to a particular compliance checker or a particular formalization of compliance rules (which are fed to the checker). For external compliance checking existing techniques like replay [4], temporal logic checking [2], or general database queries [3] can be used. In case of internal compliance checking operational support [11, 14] can be used to prevent compliance violations, for instance by revoking or granting data access, or by blocking or enforcing tasks.

Other techniques check compliance by incorporating the compliance rules already in the design [5, 6, 15], or the model is checked after design [7, 10]. However, in these cases runtime monitoring and automatic detection [5, 1] is essential, as a correct model does not necessarily imply a correct execution.

Optimization Each of the compliance checkers indicates if an execution of the IS violated a compliance rule. Depending on the setting, different steps for improving compliance then can be taken.

As mentioned above, internal compliance checking allows to prevent or mitigate compliance violations using operational support [11, 14]. If a violation cannot be prevented, particulary in case of external compliance checking, two cases may arise. (1) Either the deviation indicates a problem in the De Jure model, e.g., a wrong compliance rule. In this case, a *rule promoter* can be used to update the De Jure model to eliminate false positives in the future, see Fig. 2. (2) Or the IS or the business process are non-compliant and the process designer has to plan how to *optimize* IS and process to achieve compliance. The violated rule precisely tells which aspect of the process (e.g., which task and role) was non-compliant and where the process has to be improved.

3 Conclusions and Future Work

In this position paper we advocate the idea of managing compliance separated from BPM. In order to support this idea we introduce an engine that allows for the formal definition and auditing of compliance rules. The engine supports both detective and preventive approaches to check compliance rules. For external auditing, only the detective approach is allowed. However, the engine can be used to interrupt a business process if the next action would lead to a violation.

The crucial aspect of this approach is to identify a business vocabulary that allows to express all compliance requirements in the basic notions of business processes. By instrumenting the IS to report to the compliance engine all state changes of the primitives in the business vocabulary, compliance of the processes can be checked in a generic way and separately from the IS implementation itself. This approach requires to synchronize CM and BPM only in their formalization phases (to create consistent process models and compliance rules) and their optimization phases (to plan changes to models and rules). In all other phases, CM and BPM are separated, allowing to develop dedicated techniques particulary for CM.

Our proposed approach still faces many challenges. The elicitation phase in the life cycle still requires intense human work and knowledge for interpreting compliance sources and defining compliance requirements and compliance rules. However, tool support may assist in identifying affected processes and eliciting compliance requirements and rules. Moreover, laws usually require organizations to document how regulatory goals are achieved [13]. In particular, a compliance solution has to allow to explicitly trace the enforcement of compliance requirements in business operation. While this is not addressed in this paper, we believe that our structured approach supports traceability of compliance.

Finally, the proposed CM life cycle and our architecture address the technical side of compliance in organizations. It is meant to complement and facilitate governance programmes, such as the Unified Governance Framework [13], which define legal strategies and their enforcement on all organizational layers.

Each of the mentioned challenges is subject to further research. However the most urgent activity is experimentation of the separated compliance engine with a prototype which covers the CM life cycle.

References

1. chapter A Methodological Framework for Aligning Business Processes and Regulatory Compliance.
2. W.M.P. van der Aalst, H.T. de Beer, and B.F. van Dongen. Process Mining and Verification of Properties: An Approach Based on Temporal Logic. In *CoopIS 2005*, volume 3760 of *LNCS*, pages 130–147. Springer, 2005.
3. W.M.P. van der Aalst, K.M. van Hee, J.M.E.M. van der Werf, A. Kumar, and M. Verdonk. Conceptual Model for Online Auditing. *Decision Support Systems*, 50(3):636 – 647, 2011.
4. A. Adriansyah, N. Sidorova, and B.F. van Dongen. Cost-based Fitness in Conformance Checking. In *ACSD 2011*. IEEE, 2011.
5. M. El Kharbili, A.K. Alves de Medeiros, S. Stein, and W.M.P. van der Aalst. Business Process Compliance Checking: Current State and Future Challenges. In *Modellierung betrieblicher Informationssysteme (MobIS)*, pages 107–113. LNI, 2008.
6. D. Fötsch, E. Pulvermüller, and W. Rossak. Modeling and Verifying Workflow-based Regulations. In *Regulations Modelling and their Validation/ Verification*, pages 825 – 830, 2006.
7. A. Ghose and G. Koliadis. Auditing Business Process Compliance. In *Service-Oriented Computing ICSOC 2007*, volume 4749 of *LNCS*, pages 169–180. Springer, 2007.
8. C. Giblin, A.Y. Liu, S. Müller, B. Pfitzmann, and X. Zhou. Regulations Expressed As Logical Models (REALM). In *JURIX*, pages 37–48, 2005.
9. M. Kharbili, S. Stein, I. Markovic, and E. Pulvermller. Towards a Framework for Semantic Business Process Compliance Management. In *GRCIS*, volume 339 of *CEUR Workshop Proceedings*, pages 1 – 15, 2008.
10. R. Lu, S.W. Sadiq, and G. Governatori. Compliance Aware Business Process Design. In *Business Process Management Workshops*, volume 4928 of *LNCS*, pages 120–131. Springer, 2008.
11. F.M. Maggi, M. Montali, M. Westergaard, and W.M.P. van der Aalst. Monitoring business constraints with linear temporal logic: An approach based on colored automata. In *BPM 2011*, LNBIP. Springer, 2011. to appear.
12. C. Menzies. *Sarbanes-Oxley und Corporate Compliance: Nachhaltigkeit, Optimierung, Integration*. Schffer Poeschel, Stuttgart, 2006.
13. Waidner M. Pfitzmann B., Powers C. IBMs Unified Governance Framework (UGF). Technical report, IBM Research Division, Zurich, IBM Research Report RZ 3699 (99709) 10/12/2007.
14. A. Rozinat, M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C.J. Fridge. Workflow Simulation for Operational Decision Support. *Data & Knowledge Engineering*, 68:834–850, 2009.
15. S.W. Sadiq, G. Governatori, and K. Namiri. Modeling Control Objectives for Business Process Compliance. In *Business Process Management*, volume 4714 of *LNCS*, pages 149–164. Springer, 2007.