Hints on how to face business process compliance [*]

Cristina Cabanillas, Manuel Resinas and Antonio Ruiz-Cortés

Departamento de Lenguajes y Sistemas Informáticos

ETS Ingeniería Informática

Universidad de Sevilla

41012 Sevilla

{cristinacabanillas, resinas, aruiz}@us.es

Abstract

The concept *business process compliance* refers to the degree of conformance between the business processes of an organization and the regulations and rules that govern it. This paper intends to be a starting point for people interested in business process compliance who have no knowledge about how to address compliance checking. We introduce the four most relevant points to be considered before facing the problem and present some hints for those points in the form of a state of the art based on the literature about business process compliance checking. We also state possible future work in the context of business process compliance derived from this study.

Keywords: compliance, business processes, regulations, modelling language, compliance checking.

1 Introduction

Nowadays there is a trend towards changing the traditional design and development of products and services offered by organizations, often focused on software engineering techniques, by methods and techniques that try to eliminate the need of software engineers on initial stages of the development cycle. In this context, business processes have emerged as an "easy" way to represent the work performed by an organization with the aim of guiding the development of products and the delivery of services. A business process is a sequence of activities that work together to reach a final goal, i.e. they produce a specific product or provide a specific service. They can be modelled with different workflow languages. Furthermore, companies must fulfill a set of rules, from high-level regulations and frameworks that can be applied to as many companies as desired, such as CMMI, ITIL, COBIT and ISO rules, to low-level business rules that emerge and are applied in the specific environment of a company. These rules usually consist of books written in natural language.

In this scenario, ensuring the compliance of business processes with regulations is becoming increasingly important to organizations, since fulfilling the rules gives them a higher level of quality and is an added value to the services they provide.

Dealing with automatic (or semiautomatic) compliance checking is not an easy task, regarding business processes because many elements are involved in their models (activities, data, ...), and with regard to rules because they cannot easily be represented in a process-oriented way, i.e. visually modelled like business processes. We have identified several issues one should consider before addressing compliance checking and we have classified

[*] This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and project P07-TIC-2533 funded by the Andalusian Local Government.

the current state-of-the-art literature according to those aspects in order to find out the most common behaviour and come up with important open challenges that can be tackled in the future in the context of business process compliance checking.

We believe the first thing to do when facing this problem is giving answers to the following questions:

- *When will we check the degree of compliance?*
- *What are we intending to check?*
- *How will we do it?*
- *What languages will we use to model business processes and rules?*

We are introducing only the four most relevant issues, but we could also think of questions related to the level of automation or the visualization of the degree of compliance. Due to space limitations those questions are out of the scope of this paper.

Section 2 answers the questions above from the literature on techniques to check business process compliance. Finally, section 3 draws a set of conclusions from the study and envisages some challenges that can be addressed in the future.

2 How to address business process compliance checking

In the following we present the study we have carried out to answer the questions planned above.

2.1 When will we check the degree of compliance?

The problem of ensuring compliance of business processes with regulations and business rules can be tackled from two main perspectives. At first, researchers opted for a retrospective detection of compliance, i.e. “after-the-fact” or reactive detection, also known as Backward Compliance Checking (BCC). The work in [1,4,18,21] is a good representative of this approach. BCC techniques' main flaw is that they can neither prevent the occurrence of non-compliant situations nor modify the behaviour of the process instance during its execution to solve problems, since they just compare the results of the execution with the expected behaviour once the process execution is over.

Forward Compliance Checking (FCC) emerged with a much more preventative focus, with the aim of avoiding the previous problems. FCC techniques target the verification of rules at design time or run time, resulting in two sub-approaches: Design-Time Compliance Checking (DTCC), commonly called “compliance by design” and Run-Time Compliance Checking (RTCC). Checking compliance at design time means that we must try to make the business process comply with the rules since its design, so we can prevent non-compliant situations while modelling both elements. Similarly, RTCC techniques try to check the rules at execution time, which has some advantages, e.g. finding a non-compliant circumstance while running the process may let us solve the problem on time to avoid ending in a non-compliant result. Furthermore, RTCC can check more aspects than DTCC, such as data or resources and performance information, in an easier way. We will further discuss these aspects in the next section.

Most of the current work focuses on FCC approaches, especially on DTCC. The work in [6–15, 20, 22] comprises techniques to check the compliance at design time. Awad et al. [2] use a language called BPMN-Q to check the compliance during the execution of business processes.

2.2 What are we intending to check?

Both business processes and rules range over many aspects we should consider when addressing compliance checking.

- **Control flow.** It is the order in which tasks must be run. Business process modelling languages usually show the control flow implicitly, since modelling business processes mainly consists of representing the execution order of their activities over time. Therefore, the control flow is inherent in these models.

- **Data.** The execution of business process tasks may involve managing a large amount of data, e.g. information stored in databases may change, new data may be produced and tasks may need specific pieces of data to complete. This information can flow along the process in the form of data objects, e.g. in the form of documents. Rules concerning data management can be defined, so data objects must be also represented in the model and could be the object of compliance checking.

- **Resources.** People interacting with the business process can be considered resources and can be modelled with the process in the form of external agents. Rules stating how they must interact with the process may be defined, so they constitute an aspect that should be taken into account.

- **Time.** Temporal constraints, such as a task expiration date, are handled by means of events in most of business process modelling languages. Rules may impose restrictions on task and process execution timing, e.g. we can indicate that a task must wait eight days to be launched.

Checking the correct work of control flow, data and resource management and the fulfillment of temporal requirements is a non-trivial task that must be carried out when running a business process instance. The fact that a rule can add constraints to business processes on one or more of these aspects makes this checking even more difficult.

The main and most straightforward aspect to check in business process compliance is the control flow. Most of the current techniques assume it and do not specify whether or not they deal with the other aspects.

2.3 How will we do it?

Planning the procedure we will follow is maybe the most important task we must do before tackling the problem. It is strongly related to the classification described in Section 2.1, since techniques may be different depending on when we want to do the compliance checking.

BCC techniques will only be able to decide whether a process instance complied with the rules or not and do nothing to solve non-compliant cases but letting the experts know, so the problems can be solved before the execution of the next business process instance. We can think about monitoring business processes to check the point at which they fail or just compare them with previously run instances. The technique introduced in [18] consists of quantifying how much the execution of a business process matches its expected behaviour by comparing it with previous instances registered in history logs. Its main shortcoming is its inability to deal with data fields, temporal aspects and run-time information, since the checking takes place when the execution is over. The approach described in [21] is based on Linear Temporal Logic (LTL) and its aim is to verify if a given LTL rule holds for a set of process instances, separating the result into two groups: one containing the compliant process instances and one with the non-compliant ones. Although it can handle data and temporal and performance aspects, the inexistence of a graphical notation hinders the work of analysts. In [4] this problem is solved by modelling the rules with a graphical language called GOSpeL. The models are then translated into the declarative language SCIFF and applied over process instances the same way as Alberti et al. [1] do. Also, the approach of Alberti et al. [1] allows both kinds of FCC by means of the language g-SCIFF, developed by the authors.

Regarding FCC, many authors propose to separately model the business process and the rules, do the transformations required to convert the two models into formal representations (see Section 2.4) and then apply model-checking algorithms to evaluate the degree of compliance [2, 6, 14, 19]. Some of these techniques include the presentation of counterexamples that show where the non-compliance problem has arisen. Sometimes the authors use patterns, either to simplify the modelling of the rules or to facilitate the task of checking compliance in scenarios that often recur. The procedure described in [8] is based on the annotation of business process models with effects derived from the contracts. It requires a process for pair-wise effect acummulation and the semantics necessary to transform the annotated model into a Semantic Process Network (SPNet), which is used to check the degree of compliance. Some structural and semantic patterns are proposed to avoid the occurrence of non-compliant situations. In a similar way, Weber et al. [22] introduce a formalism that acts on business process models in any typical workflow language annotated with predicate logic. A propagation algorithm is used to go along the process and its result is the input for compliance checking. The shortcoming of this approach is that the formalism does not support loops and computationally hard cases and it does not deal with resources and temporal aspects. The idea in [12, 13] is to enhance compliance management by leveraging compliance checking to a semantic level through ontologies. Namiri et al. [16] present a three-step compliance checking approach based on the introduction of a new layer with the representation of controls (i.e. rules) over the business process model. However, this technique requires much manual work by experts. The authors in [10,11,15] lean on the idea of accumulating effects across the tasks and use the concept of Ideal Semantics to assess compliance according to the degree of idealism the process results on. The approach described by Goedertier et al. [9] consists of declaratively capture the rules (with their own language called PENELOPE) and (re)use them to generate the business process model that will be checked for compliance. Ghanavati et al. [7] describe a constraint management framework for compliance checking focused on the hospital domain. The approach separately models the functional and operational aspects of business processes, the non-functional aspects of them and the regulatory policies and then tries to link them in order to find non-compliances. Its main flaws are its low scalability and the great need of manual compliance checking.

2.4 What languages will we use to model business processes and rules?

The term compliance involves two elements. On the one hand are the business processes that respond to enterprise needs and show what activities must be carried out in a company. On the other hand are the rules the company must fulfill. Both elements need to be modelled somehow in order to automatically check the degree of compliance between them, so proper languages must be chosen for that purpose.

Regarding business processes, most of the authors opt for BPMN as modelling language, since it is very intuitive and easy to use [17]. Some of them annotate the BPMN models with other languages that define the rules [8, 10, 11, 19, 22]. This way, the subject of the compliance checking will be a BPMN model enriched with the rules. Liu et al. [14] propose the de facto standard BPEL for business processes modelling. Ghanavati et al. [7] model them with Use Case Maps (UCM). These three notations have a graphical visualization and are supported by existing tools. Other authors prefer not to specify a language and say just that a directed graph or any business process execution language can be used for process modelling [6, 14]. Kharbili et al. [12, 13] say nothing about how to model the processes and focus on developing an ontology for rules.

As BPMN semantics is not well specified, since there is not a standard metamodel, a solution could be translating the BPMN into PNML (Petri Net Markup Language). Dijkman et al. [5] have developed a tool with this aim.

As far as rules are concerned, some approaches use formal languages to model them, such as FCL (Formal Contract Language) [10, 11, 19] and PENELOPE (Process ENtailment from the ELicitation of Obligations and

PErmissions) [9], predicate logic [22] and Conjunctive Normal Form (CNF) [8]. The main advantage of this kind of languages is that they can be more easily interpreted by the machine. On the contrary, they do not have a graphical representation, so the effort made by analysts is cumbersome. This is solved by languages such as BPSL (Business Property Specification Language) [14], GRL (Goal-oriented Requirement Language) [7], BPMN-Q [2] and PPSL (Process Pattern Specification Language) [6]. The two last ones are quite similar and very close to each other in terms of expressiveness. The main difference between them is that PPSL models constraints against UML Activity Diagrams and BPMN-Q does not.

Many approaches propose modelling both business processes and rules with graphical languages and then translating them into formal specifications. BPMN and BPEL models are usually translated into FSM (Finite State Machine) and graphical languages for rule definition are usually translated into LTL (Linear Temporal Logic), a widely used language for specifying temporal properties of software and hardware designs. These languages are suitable as input of most of the current model-checking tools.

3 Conclusions

The main intention of this paper was to act as a guide for those interested in facing the automatic (or semiautomatic) checking of the degree of compliance between the business processes and the rules that govern the business of an organization. The conclusions drawn from the four analysed points are the following: (1) Most of the proposed techniques focus exclusively on checking the compliance while designing the process, although sometimes they consider also run-time aspects. Only one of the approaches developed so far can cover both FCC and BCC. However, this approach is incomplete for our purpose because it handles only rules and say nothing about business processes. (2) Only two (control flow and time) out of the four elements that must be checked are taken into account in most of the approaches. This is partially due to the fact that languages such as LTL allows modelling these two aspects implicitly. (3) Although each approach has its own features that may make it better or worse with respect to others, the most common scenario regarding the procedure used to do the compliance checking consists of separately modelling the processes and the rules (using proper languages), translating these models into more formal languages and applying some model-checking algorithm to measure the degree of compliance. (4) Finally, we have found out that the most commonly used process modelling language is BPMN [17], and that, on the contrary, several languages have emerged in order to represent the compliance rules and they can be quite different from each other, usually depending on the kind of rules the authors have considered.

From this study we can also conclude that business process compliance checking is a complex problem and further research can be done on this topic. For instance, an interesting challenge could be how to model the rules, which is an important gap between processes and regulations. We could also think of including aspects such as data and resources in compliance checking, since they are excluded in most of the current approaches. Some work on data-related compliance problems that can appear in a business process model has been done, but there is no tool that integrates data-aware compliance into any existing compliance checking framework yet [3].

References

[1] M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello, M. Montali, and P. Torroni. Expressing and verifying business contracts with abductive logic programming. *Int. J. Electron. Commerce*, 12(4):9–38, 2008.

[2] A. Awad, G. Decker, and M. Weske. Efficient compliance checking using bpmn-q and temporal logic. In M. Dumas, M. Reichert, and M.-C. Shan, editors, *BPM*,

volume 5240 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2008.

[3] C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. On the identification of data-related compliance problems in business processes. In *JSWEB*, page in press., 2010.

[4] F. Chesani, P. Mello, M. Montali, and S. Storari. Testing careflow process execution conformance by translating a graphical language to computational logic. In *Artificial Intelligence in Medicine*, pages 479–488. 2007.

[5] R. M. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in BPMN. *Inf. Softw. Technol.*, 50(12):1281–1294, 2008.

[6] A. Forster, G. Engels, T. Schattkowsky, and R. V. D. Straeten. Verification of business process quality constraints based on visual process patterns. In *Theoretical Aspects of Software Engineering, 2007. TASE '07. First Joint IEEE/IFIP Symposium on*, pages 197–208, 2007.

[7] S. Ghanavati, D. Amyot, and L. Peyton. Towards a framework for tracking legal compliance in healthcare. In *Advanced Information Systems Engineering*, pages 218–232. 2007.

[8] A. Ghose and G. Koliadis. Auditing business process compliance. In *ICSOC*, pages 169–180, 2007.

[9] S. Goedertier and J. Vanthienen. Designing compliant business processes from obligations and permissions. In J. Eder and S. Dustdar, editors, *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science*. Springer Verlag, 2006.

[10] G. Governatori, Z. Milosevic, and S. W. Sadiq. Compliance checking between business processes and business contracts. In *EDOC*, pages 221–232. IEEE Computer Society, 2006.

[11] G. Governatori and S. Sadiq. The journey to business process compliance. In *Handbook of Research on BPM*, pages 426–454. IGI Global, 2009.

[12] M. E. Kharbili and S. Stein. Policy-based semantic compliance checking for business process management. In P. Loos, M. Nuttgens, K. Turowski, and D. Werth, editors, *MobIS Workshops*, volume 420 of *CEUR Workshop Proceedings*, pages 178–192. CEUR-WS.org, 2008.

[13] M. E. Kharbili, S. Stein, I. Markovic, and E. Pulvermuller. Towards a framework for semantic business process compliance management. Montpellier, France, 2008.

[14] Y. Liu, S. Muller, and K. Xu. A static compliance-checking framework for business process models. *IBM Systems Journal*, 46(2):335–362, 2007.

[15] R. Lu, S. Sadiq, and G. Governatori. Compliance aware business process design. In A. H. M. ter Hofstede, B. Benatallah, and H.-Y. Paik, editors, *5th International Conference on Business Process Management (BPM 2007)*, pages 120–131. Springer, 2008.

[16] K. Namiri and N. Stojanovic. Using control patterns in business processes compliance. In M. Weske, M.-S. Hacid, and C. Godart, editors, *WISE Workshops*, volume 4832 of *Lecture Notes in Computer Science*, pages 178–190. Springer, 2007.

[17] OMG. Bpmn 2.0 beta 1. Recommendation, OMG, 2009.

[18] A. Rozinat and W. M. P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 2008.

[19] S. W. Sadiq, G. Governatori, and K. Namiri. Modeling control objectives for business process compliance. In G. Alonso, P. Dadam, and M. Rosemann, editors, *BPM*, volume 4714 of *Lecture*

Notes in Computer Science, pages 149–164. Springer, 2007.

[20] M. Saeki and H. Kaiya. Supporting the elicitation of requirements compliant with regulations. In Z. Bellahsene and M. Leonard, editors, *CAiSE*, volume 5074 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2008.

[21] 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 *OTM Confederated International Conferences: CoopIS, DOA and ODBASE*, volume 3760, pages 130–147. Springer-Verlag, October 2005.

[22] I. Weber, G. Governatori, and J. Hoffmann. Approximate compliance checking for annotated process models. Technical report, 2008.