On the identification of data-related compliance problems in business processes *

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

Ensuring the compliance of business processes with regulations is becoming increasingly important to organizations. In this scenario, data play an important role. Little work has been done on data checking in business processes and no standard definitions have been given to describe data-related compliance problems. The goal of this paper is three-fold: (i) to identify and organise the data-related compliance problems that may arise in a business process model and thus to introduce a common vocabulary for these problems, (ii) to analyse the capabilities of BPMN 2.0 for defining business process models with sufficient information about data to enable the checking of data-related compliance problems, and (iii) to describe the current situation of data-related compliance in terms of the existing automated support and envisage future work to deal with data-aware business process compliance checking.

Keywords: business process, compliance, compliance rule, data-related problem, BPMN.

*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.

1 Introduction

In the last few years organizations have shown great interest in making their business processes compliant with the rules they are governed by, e.g. regulations and frameworks. Regulations such as HIPAA (Health Insurance Portability and Accountability Act) and SOX (Sarbanes-Oxley Act) have emerged with the aim of avoiding fraud and financial scandals such as those occured with the companies Enron (USA) and HIH (Australia) in 2001. Besides these regulations, CMMI, ITIL, COBIT and an increasingly larger set of ISO rules have appeared aiming at guiding the organizations in the completion of their processes, that is, as best practice guides that explain how to implement the processes successfully, e.g. ITIL may be defined as a set of best practices to make the delivery of IT services easier. These regulations and frameworks together with organization specific business rules are materialized into a set of compliance rules with which the business processes must be compliant.

In this paper we asumme that these compliance rules are given in a process-oriented way, i.e. they are expressed in terms of concrete elements of the business processes, as it has been done in several approaches that have emerged to face business process compliance in organizations [1, 3, 9, 12, 14, 16, 20]. The way in which regulations and frameworks, usually expressed in natural language, are modelled is an important research challenge nowadays, but it is out of the scope of this paper.

As mentioned in [23], four aspects of the process must be taken into account, namely the control flow, the data objects, the resources and temporal constraints. Most of the techniques developed so far focus on verifying the control flow and temporal constraints, but they tend to forget the rest of aspects, which are important as well [12, 16].

Our interest is focused on those data-related compliance problems that may arise in a business process model. As described in [19], in the context of business process models, data objects provide information about what activities require to be performed and/or what they produce. A data object has a well-defined life cycle. Data objects are exchanged between the activities of a process, which can modify them and make them transition from one state to another. For instance, imagine a job application process. This process needs a document that consists of an application form sent by the person who applies for a job. This document will be received, then read and then, probably, its content will be assessed to decide whether to call the applicant for an interview or not. In the process model that describes the application procedure, the data of the application document will be represented as a data object associated with those activities of the process in charge of manipulating the document, e.g. checking the completeness of its content, assessing the data, and suchlike. During the execution of the process, the data object will have different states, e.g. checked, accepted, rejected. In the rest of this paper, the terms data, object and data object will be used in the same way to refer to the data objects of the business process models.

Some compliance rules concerning the management of data in the process may be defined in order to check the degree of compliance of the process with the desired behaviour, e.g. some activities must receive certain data objects in certain states to complete or certain state of an object will only be reachable from a concrete state. All these compliance rules must have been considered while modelling the process in order to state that this is compliant. Following with the example above, a compliance rule could state that the applicant must be notified when her application document has been accepted or rejected. Translating it into a process-oriented compliance rule, it would mean that whenever an activity makes the data object that represents the data of the application document transition to state accepted or to state denied, an activity in charge of notifying the result to the applicant must be executed.

Our goal in this paper is to identify and organise the data-related compliance problems and to analyse them in terms of these three questions:

- *What data-related information* must *appear in the process diagram to be able to check these problems?*
- *What information must the compliance rules related to these problems have?*
- *How much support has been implemented so far for these problems?*

The paper is structured as follows: in Section 2 we describe how BPMN 2.0 defines and represents data in process models. Section 3 identifies the data-aware compliance problem and introduces a list of data problems that must be considered when developing compliance checking techniques. These problems have been obtained from previous work and have been extended with new ones. The questions above are answered for every problem. Section 4 introduces related work on business process compliance. Finally, in Section 5 we draw a set of conclusions and introduce some future work.

2 Data definition and representation using BPMN 2.0

Business Process Modelling Notation (BPMN) is a standard (language) for business process modelling aimed at offering an intuitive graphical notation to specify business processes. We chose BPMN as the object of our study because it is the standard for business process modelling and hence finding out its deficiencies related to the modelling of data and the identification of data-related problems in models is very appealing.

In this paper we focus on the working proposal for BPMN 2.0 [19], which intends to correct the deficiencies of the previous version by extending the BPMN metamodel and its semantics to include different kinds of process diagrams, such as choreographies and conversations, and to enhance the management of artifacts (e.g. messages and data objects) in models. Activities represent points in a process where work is performed. They are executable elements such as tasks, sub-processes and call activities. Therefore, the term task refers to a concrete type of activity in BPMN. Every activity will implicitly have a data input set and a data output set, which can be empty sets. Moreover, every data item can be provided with a state and data structures, data types and contents can also be added. Data warehouses can be created and associated with data objects defined in the business process model.

In BPMN 2.0 pools and lanes have the same meanings as in the previous version. A pool usually represents an entity or a whole organization and lanes within a pool are roles or persons that participate in the process. Both pools and lanes can (and should) be given a name. A process always begins and ends in a single pool. BPMN 2.0 calls the models where two or more pools exchange messages *collaborations*. In collaboration diagrams each pool has its own execution flow or process and, therefore, its own data objects. Different persons may model different pools and lanes for the same collaboration, since granurality for these terms is not specified.

Figure 1 shows a very simple collaboration where a user authenticates himself/herself to a Web site. The overall collaboration consists of the following. If the user is already registered in the site, just a login stage is required. Otherwise, a registration request must be previously attended by the system to allow the user to authenticate later.

In the model, one pool with a single lane represents the user and another pool with also one single lane represents the system. Different processes are carried out in each pool. In the case that we had more information about the parties involved, a different division of pools and lanes could have been made. For instance, if two different roles of the system, named e.g. checker and notifier, took part in the system process, the pool that represents the system would have two lanes, one named checker and another one named notifier.

As explained in [19], a data object can appear multiple times in a process diagram. Each of these appearances references the same data object instance. Multiple ocurrences of a data object in a diagram are allowed to simplify diagram connection. Every appearance of a data object may be associated with one state. In the example of Figure 1 there is a total of three different data objects, named *Authentication*, *Request* and *RegistrationInfo*, each of which appears at least once in the model. As depicted in the figure, each data object goes through several states along the process. States are specified below the names of the objects. Data associations in the form of connectors are used to move data between activities. Data associations have one or more sources and one target.

Activities may read the contents of data objects and change some values, so the objects' states may also change. For readability reasons, BPMN 2.0 allows the specification of data inputs and data outputs instead of data objects. Data inputs have the same notation as data objects, except that they *must* contain a small, unfilled block arrow; they *must not* have incoming data associations. Data outputs have also the same notation as data objects, except that they *must* contain a small, filled block arrow; they *must not* have outgoing data associations. In Figure 1 there are four data objects, one data input and four data outputs. Finally, as shown in the figure, pools exchange messages, which may contain data. Nevertheless, every pool has its own data objects.

Although the model in Figure 1 is syntactically correct, the process may not work prop-

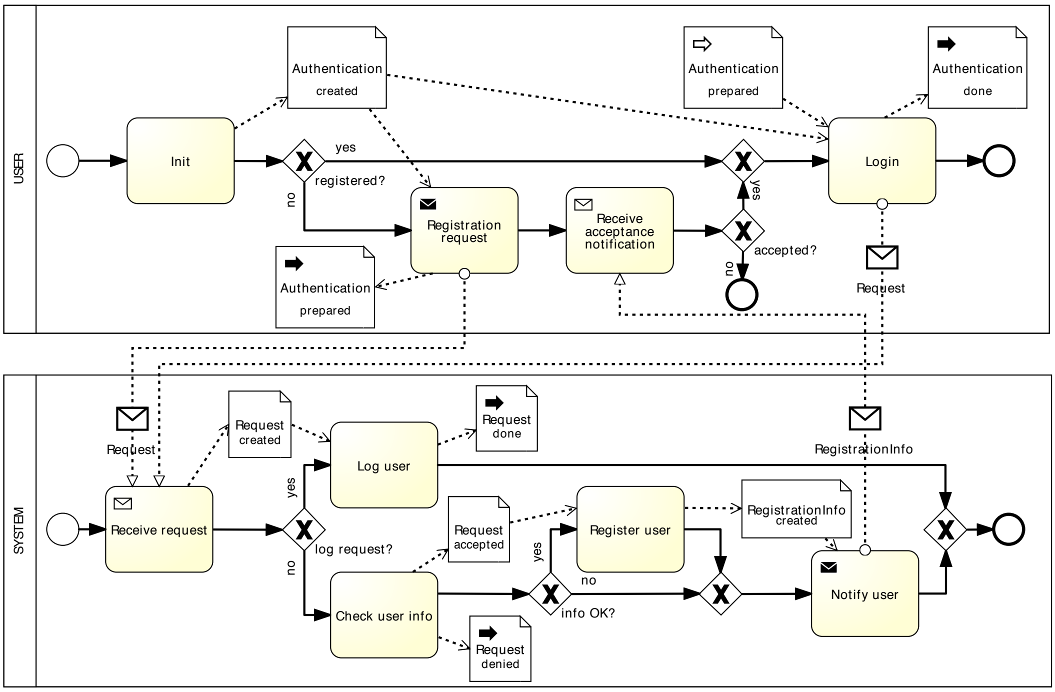


Figure 1: Collaboration of a user authentication in a Web site

erly without solving some data-related problems contained in it. We will define and remark on these problems later in this paper.

3 Data-related compliance problems in business processes

The term *compliance* implicitly involves the participation of two elements: the process or element whose degree of compliance we want to check, and the compliance rules the first element must comply with. The result of checking the compliance between two elements is that something is (fully) compliant, partially compliant or non-compliant with something. Some proposals have addressed business process compliance with regulations using different approaches [1, 3, 9, 12, 14, 16, 20], and their common point is the use of rules as the elements to check the compliance with.

Data-related compliance problems will be those in which data objects do not fulfill some data-related compliance rules previously described. These compliance rules may be expressed in different ways, as explained in the remainder of this section. BPMN 2.0 can be used to define the data objects used in business process models, i.e. information about their state, their contents, the activities that use them, etcetera. However, data object definitions must be accompanied by a compliance rule model that will include all the compliance rules required to detect every data-related problem.

In the last few years some work facing data problems in processes has emerged [2, 4, 21, 22, 25], but, to the best of our knowledge, most of these approaches are not focused on compliance problems but on defining data anomalies, whose meaning is far away from the concept of compliance. We have established an important separation between the terms anomaly and compliance problem, which will be further explained in Section 4. With the focus on business process compliance, we have classified the identified data-related compliance problems into three groups, according to the object of the compliance checking. Some of these data-related compliance problems have already been identified in [2, 22, 25] but others are described in this paper for the first time in the context of data-aware compliance. We will answer the questions raised in Section 1 for each problem defined.

- **Content of data:** this group contains problems related to the information stored in data objects and the activities that write or read them.
- **Relation between data states and activities:** they consist of compound conditions in which the state of the data objects and the activity execution order are involved. Therefore, control and data flows must be considered in this kind of data problems.
- **Evolution of data objects:** the problems related to the evolution of data objects are about the allowed transitions between states in each data object.

Table 1 summarizes the data-related compliance problems identified in the last years [4, 21, 22] and those new ones identified in this paper. They have been separated into the three groups described above. As depicted in the table, automated support has been developed for several problems, but not for all of them. For the problems extracted from [4, 21, 22], we have adopted the same names given by the authors.

3.1 Data-related compliance problems regarding the content of data

The compliance rules required to detect the problems of this group are about the data that a certain activity can and/or must use, e.g. descriptions of the data objects that can/must be input and output of every task. Therefore, in this kind of problems the importance is on the type of information managed by an activity, regardless of the state of the data objects.

At the model level, the compliance rule model required to define this kind of data-aware compliance problems consists of extra information about the relation between the tasks and the data objects represented in the process model, as well as the contents of these data objects. There are three kinds of data problems regarding the content of data.

- **Insufficient input data.** As explained in [22], this problem will appear when an activity does not receive all the data it needs to complete successfully. Concerning the diagram, those activities that use certain data must be connected to the corresponding input data object. Furthermore, the content of this data object must be well defined, i.e. in great detail, since the problem may be related to the object itself or to any piece of data contained in it. Compliance rules consist of full information about the input data *required* by every activity in the process.

  Imagine we have a compliance rule that states that in the process shown in Figure 1 task *Check user info* needs the user data contained in data object *Request* to verify them in order to decide whether the user is allowed to get registered or not. However, this data object is not defined as input data of the task in the model, so the task does not have sufficient information to execute and, thus, there is a compliance problem between the compliance rule and the process model.

- **Insufficient output data.** Similarly to the previous problem, sometimes specific information must be produced by an activity. This problem can be seen as a matter of incomplete data. Like before, the activities that produce certain data must be connected to the corresponding output data object and the content of this data object must be defined in depth. Compliance rules consist of full information about the output data *required* from every activity in the process.

Our Proposal		Sadiq et al. [22]	Ryndina et al. [21]	Awad et al. [4]
Content of data	Insufficient input data	-		
	Insufficient output data			
	Prohibited data			
Relation between data states and activities	Data rule violation			+
	Conditional *leads to* violation			+
	Conditional *precedes* rule violation			+
	Segregation of duty			
Evolution of data objects	Data object life cycle conformance		+	
	Data object life cycle coverage		+	

+ Implemented   - Not implemented

Table 1: Automated support for data-aware business process compliance

In our Authentication process example, a compliance rule could state that the *username* is needed both to log and to register the user. In the process model, a problem would exist if data object *Request* generated by task *Receive request* did not have this piece of data. In the presence of this data problem, the remains of the process would not execute properly or a failure would occur.

- **Prohibited data.** Some data objects may carry confidential information or data that may be considered private for any reasons, so only certain activities should be allowed to access them. The problem called prohibited data occurs when some activity receives data it should not receive. To check for the non-existence of this problem we would have to verify that the data input set of every activity is only composed by data objects allowed for that activity. As far as the business process model is concerned, the only requirement is to properly model the input data of the activities that use data objects. As compliance rules we would need to know which activities or roles can have access to certain information, i.e. the visibility of data in the process. This data-related compliance problem can be found at activity level and also at lane or pool levels, since there can be prohibitions that affect one specific role or, even, one whole organization.

  Figure 1 helps illustrate this problem. Let us suppose there is a compliance rule that states that data object *RegistrationInfo* has confidential information and can only be used by the tasks performed by the system. The figure depicts this object as input data of task *Notify user* in pool *System*. The problem appears when task *Notify user* puts the data in a message and sends it to task *Receive acceptance notification*, which belongs to pool *User*. A simpler example of the same problem could be that in which the compliance rule defined a list of tasks of pool *System* that were not allowed to access data object *RegistrationInfo*. If that list contained task *Notify user*, the detection of the problem would be obvious, since the data object appears as input data of that task.

3.2 Data-related compliance problems regarding the relation between data states and activities

In this group of problems the compliance rules are conditions concerning the control flow of the process and the states of the data objects used by the activities. With regard to the process model, in order to have enough information to be able to detect these problems, the state of the data objects must be specified and the activities that use data must be connected to the corresponding input and/or output data objects. The specification of the compliance rules depends on the type of problem. The three first data-related problems described below have already been described in [4], so we have adopted the same names given by the authors.

- **Data rule violation.** This problem is about the state of the data used by a certain activity. A compliance rule of this type states that an activity can execute only if certain data object in a certain state is received as input. The process will wait until this condition holds to continue the execution. It can also be applied to output data. In that case, it is necessary to specify the state that a certain output data object must have after being used by an activity.

  An example could be formulated as "the user request must be accepted for the registration task to proceed". As shown on activity *Register user* in Figure 1, it holds in the example.

- **Conditional *leads to* violation.** This problem is related to the task execution order and the state of the output data. In particular, this type of compliance rule specifies that, at any moment after a certain activity X transitions a certain data object D to a certain state S, a concrete activity Y *must* be executed before the end of the process. The problem will appear either if the former activity does not change the state of the data object into the state defined by the rule, or if, having done it, the latter activity mentioned in the rule is not executed before the end of the process. Consecution between the two activities involved in the compliance rule is not required.

  In Figure 1, one holding example could be stated as "the user must be notified if his/her registration has been denied (after checking his/her request)". Translating it into the example model, we could state that task *Check user info*, with output data object *Request* in state *denied* leads to task *Notify user*.

- **Conditional *precedes* rule violation.** The compliance rules for this problem state that before a certain task X is executed, another task that changes the state of a certain data object D to a certain state S must have been executed, i.e. activity X must never be executed if there is not a previous activity that changed the state of the data object D to the state specified in the rule. X may be a final event, meaning that the process can finish only if data object D has been in state S previously. Consecution between the two activities involved in the compliance rule is not required.

  An example can be formulated as "before registering a user in the system a request with the user data must be created". This statement has a direct translation into the model in Figure 1, which is that task *Register user* must be preceded by a task that has data object *Request* in state *created* as part of its output set. It holds in the example model.

An extension of the two previous problems consists of the explicit exclusion of an activity between the two activities that take part in the condition of the compliance rule. For instance, we could extend the example in the Conditional *leads to* violation problem to say that "the user must be notified if his/her registration has been denied (after checking his/her request) and no registration activities can be carried out between these actions".

Translating it into the example model, we could state that task *Check user info*, with output data object *Request* in state *denied*, leads to task *Notify user* excluding the execution of task *Register user*. Similarly, we could extend the condition defined as example in the Conditional *precedes* rule violation problem to say that "before registering a user in the system a request with the user data must be created and the user must not be notified so far". In the example model, the rule means that task *Register user* must be preceded by a task that has data object *Request* in state *created* as part of its output set, and task *Notify user* must not be executed at any point between those two activities.

- **Segregation of duty.** The term Segregation of Duty (SoD) is a well-known security principle in financial accounting systems. A typical scenario where it may take place is the opening of a bank account, where two different officers must sign the acceptance letter before going on with the opening process. The goal is to prevent fraud and errors by disseminating the tasks and associated privileges for a specific business process among multiple users.

  In the aforementioned data problems the problems were related to activities that took part in a process, but it did not matter whether people belonging to a single or to various roles in the process participated in the tasks involved in the compliance rules. In the segretation of duty data-related problem, the business process model must include at least two lanes. The generic compliance rule can be formulated as "two similar tasks carried out by two different persons, both of them updating the same data object and leaving it in similar states, must be executed for the process to continue". The most important point is that thanks to the completion of the two tasks the data object will transition to a certain state, which is required for the process to continue.

3.3 Data-related compliance problems regarding the evolution of data objects

The life cycle of a data object provides information about the states the object is allowed to have at a certain moment of the process execution and the next states it can transition to. It is, thus, an important element that could be added as compliance rule to ckeck data-related compliance regarding the evolution of data objects.

Therefore, we assume the analyst has previously generated the object life cycle of every data object of the business process. The goal is to check for two compound problems related to the life cycles represented in the process model. These problems have been identified in [15, 21], so we have adopted the same names given by the authors.

- **Data object life cycle conformance.** To avoid this data-related problem, everything modelled in the diagram for each data object (i.e. states and transitions) must appear in its object life cycle. Three compliance rules must be checked: (i) first state conformance, (ii) last state conformance, and (iii) transition conformance. Due to space limitation, we will define them with examples and we refer the reader to [15, 21] for further understanding.

  Figure 2 shows the object life cycles of two of the data objects represented in Figure 1. Regarding the process model with respect to these object life cycles, the Authentication process complies with the first state conformance condition because the first state of every data object in the model (i.e. the first state that that object has in the model) corresponds with a first state of its life cycle. However, last state conformance does not hold, since state *prepared* appears in the model as a final state of data object *Authentication* and it is not a final state in its object life cycle (Figure 2a). The same happens with states *accepted* and *denied* of object *Request* (Figure 2b). Transition conformance is also violated because transition *prepared-done* represented in the model is not a valid transition in the *Authentication*'s object life cycle (Figure 2a).

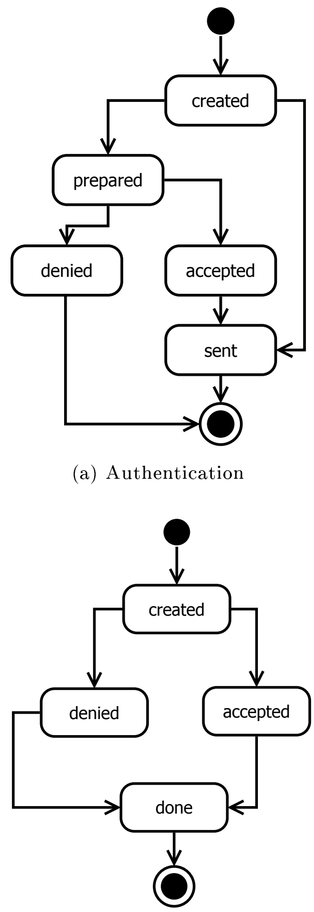
- **Data object life cycle coverage.** The three compliance rules included in this problem are similar to the previous ones, but in the opposite direction. Now, the objective is to ensure that everything represented in each object life cycle (i.e. states and transitions) is also modelled in the diagram for the corresponding data object. As before, we will define them with examples and we refer the reader to [15, 21] for further understanding.

  The Authentication process depicted in Figure 1 complies with the first state coverage condition for the two objects shown in Figure 2 because every first state of the life cycles is also a first state in the model for the corresponding data object. The process model does not comply with the last state coverage condition because the model does not include state *denied* of object *Authentication* as final state. Finally, it neither complies with the object life cycle transition coverage rule for several reasons. On the one hand, data object *Authentication* lacks transitions *prepared-accepted* and *prepared-denied* in the model of Figure 1. On the other hand, transitions *accepted-done* and *denied-done* of object *Request* do not appear in the process model.

Two subproblems are implicitly covered by the aforementioned checkings.

- **Invalid data state.** It consists of checking whether a certain state represented in the model exists in the object life cycle of the corresponding data object, e.g. state *done* of object *Authentication* is not in Figure 2a.

- **Mismatched data state.** It is about checking whether a transition between



(a) Authentication

(b) Request

Figure 2: Object life cycle of Authentication and Request

two states present in the process model matches with a transition in the corresponding object life cycle. Comparing the data in Figure 1 with the life cycles in Figure 2, we see that the transition from state *prepared* to state *done* for object *Authentication* does not appear in its object life cycle (Figure 2a).

There is not just one single way of solving all the aforementioned problems to get a compliant business process model from Figure 1. Figure 3 shows one possible solution, which includes the correction of the violations described for every compliance problem. Note that, for the sake of simplicity, all the examples given for the problems regarding the relation between data states and activities hold in the process model.

3.4 Automated support

Diagnosing and repairing all these data-related compliance problems is not a trivial issue. Some approaches have emerged, which cover some of the problems to a greater or lesser extend. Table 1 collects the results of our research on data-aware business process compliance with regard to the aforementioned problems. Problems *insufficient output data*, *prohibited data* and *segregation of duty* introduced in this paper had never been identified before in the context of data-aware compliance.

The automatic detection of the data-related compliance problems regarding the evolution of data objects can be carried out by means of state machines and ad-hoc algorithms [15, 21]. The work described by Kuster et al. in these papers involves automatically generating a business process model from the life cycles of the objects used in the process. This way the authors ensure that the process is compliant with all the object life cycles. A prototype has been implemented as an extension of IBM WebSphere Business Modeler.

As far as the compliance problems regarding the relation between data states and activities are concerned, BPMN-Q language can be used to query about data-aware compliance over the business process model, and then these queries can be converted into PLTL (Past Linear Temporal Logic) and TLQ (Temporal Logic Querying) can be used for compliance checking. This approach has been used by Awad et al. [4] to address three out of the four compliance problems belonging to that group and has already been integrated as part of their BPMN-Q query processor engine.

To the best of our knowledge, BPMN-Q can be used to define some types of compliance rules, but not all of them. For instance, we cannot express that certain information contained in a data object is required by certain activity, since BPMN-Q does not allow describing the kind of data the objects carry. Thus, stating rules such as those given as examples in the problems of Section 3.1 is not possible. We could think of an extension of BPMN-Q for this purpose.

4 Related work

The problem of ensuring compliance between business processes and regulations and business rules has been addressed in the last years from different perspectives. Some researchers propose a retrospective detection of compliance, i.e. "after-the-fact" or reactive detection, also known as Backward Compliance Checking (BCC). It usually consists of the comparison of the business process model with the results of several executions stored in log files. Alberti et al. [1, 5, 20] are good representatives of this approach. The main fault of BCC techniques is that they cannot modify the behaviour of the instance during its execution or prevent the occurrence of non-compliant situations, since the compliance checking takes place once the process execution is over.

To avoid this problem Forward Compliance Checking (FCC) appeared, which has a much more preventative focus. FCC techniques target the verification of rules at different moments, namely design time, post-design time and run time. Checking the compliance at design time involves trying to make the business process comply with the rules from the modelling phase, so we can prevent non-compliant

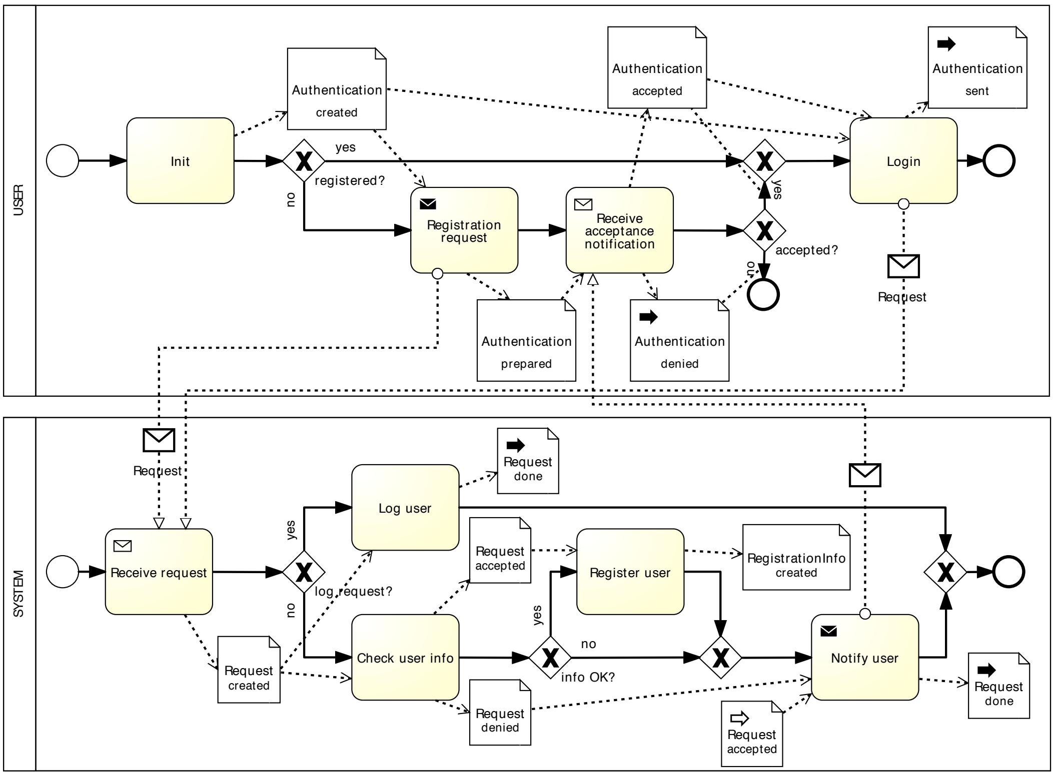


Figure 3: Collaboration of a user authentication in a Web site (without data problems)

situations while modelling the business process and the compliance rules. Post-design techniques check compliance issues right after modelling the process with the aim of correcting possible violations before executing the process. Run-time compliance checking techniques try to check the performance of rules at execution time. Most of the current work on business process compliance focuses on FCC, especially on desing-time compliance checking. [3, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 23, 24] have interesting work on FCC.

However, most of these proposals tackle only control flow problems while checking business process compliance, and important aspects such as data and resources are ignored.

As explained in Section 3, process compliance involves the necessity of compliance rules. Therefore, those problems that can be found in a process model with no need of compliance rules (e.g due to design mistakes or sintactic errors in the business process models) should be named in a different way. We have called the data problems that can be validated in the model *anomalies* to distinguish them from the data-related compliance problems. We are interested in data-related compliance issues and, hence, these data anomalies are out of the scope of this paper.

Nevertheless, most of the approaches facing data-related issues in processes [2, 4, 15, 21, 22, 25] focus on data anomalies and do not address compliance problems [2, 22, 25]. Sadiq et al. [22] explain the importance of managing the data requirements in business processes and introduce some ideas related to the modelling and validation of data, such as the importance of considering the type of data and their structure. They also state some data problems, most of which are data anomalies. These data-related problems are referenced by the authors in [25], who divide the same problems into three groups, each of which contains one or more scenarios. Sun et al. [25] explain the matching of every scenario with the anomalies in [22]. The anomalies in [2] are described independently from those in [22, 25], but there is a possible correspondence between them.

Furthermore, some validation algorithms are being implemented to correct the data anomalies in BPMN [25]. Awad et al. [2] are implementing a prototype of their approach in Oryx [6].

5 Conclusions

A study about data-aware business process compliance has been carried out in this paper. The first important conclusion we have drawn is that not every problem related to data in business processes can be considered a compliance problem. Furthermore, there is not a standard vocabulary to speak about data-aware compliance, so the first need is to adopt the same vocabulary to talk about data object management in business processes. We tried to give a first step in this direction by establishing a distinction between data anomalies and data-related compliance problems. Thus, data-aware business process compliance may be defined as the fact of ensuring the fulfillment of rules that specify the use of data objects by the activities of a business process.

Another conclusion drawn from this analysis is that a model of compliance rules must be provided to check the degree of data-aware compliance in the model. This model must provide a means to represent object life cycles and define privacy rules and data addressing. Moreover, we have realized that much information about the data has to be specified in the process diagrams in order to have complete descriptions of the data objects. This can make the diagrams quite unreadable. Therefore, the readability of models with a large amount of data in BPMN should be improved.

Finally, support for detecting some data-related problems has been implemented, but each proposal uses its own work environment and a different language to model the processes. We think that a framework that integrates the developed solutions and gives support for the implementation of the new identified problems and those that have not been implemented yet is necessary. This framework should provide support for the analysis of business process compliance regardless of the notation and the logic paradigm used.

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 N. Lohmann. Diagnosing and repairing data anomalies in process models. Technical report, Potsdam, Germany, 2009.

[3] 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.

[4] A. Awad, M. Weidlich, and M. Weske. Specification, verification and explanation of violation for data aware compliance rules. In *ICSOC/ServiceWave*, pages 500–515, 2009.

[5] 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.

[6] G. Decker, H. Overdick, and M. Weske. Oryx - an open modeling platform for the BPM community. In *Proceedings of the 6th International Conference on Business Process Management*, pages 382–385, Milan, Italy, 2008. Springer-Verlag.

[7] 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.

[8] 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.

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

[10] 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*, pages 5–14. Springer Verlag, 2006. Springer-Verlag Berlin Heidelberg 2006.

[11] 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.

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

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

[14] M. E. Kharbili, S. Stein, I. Markovic, and E. Pulvermuller. Towards a framework for semantic business process compliance management. 2009.

[15] J. Kuster, K. Ryndina, and H. Gall. Generation of business process models for object life cycle compliance. In *Business Process Management*, pages 165–181. 2007.

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

[17] R. Lu, S. Sadiq, and G. Governatori. Compliance aware business process design. In *BPM 2007 international workshops (BPI2007)*, 2008.

[18] 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.

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

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

[21] K. Ryndina, J. Kuster, and H. Gall. Consistency of business process models and object life cycles. In *Models in Software Engineering*, pages 80–90. 2007.

[22] S. Sadiq, M. E. Orlowska, W. Sadiq, and C. Foulger. Data flow and validation in workflow modelling. In K. Schewe and H. E. Williams, editors, *Fifteenth Australasian Database Conference (ADC2004)*, volume 27 of *CRPIT*, pages 207–214, Dunedin, New Zealand, 2004. ACS.

[23] 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.

[24] 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.

[25] S. X. Sun, J. L. Zhao, J. F. Nunamaker, and O. R. L. Sheng. Formulating the Data-Flow perspective for business process management. *Info. Sys. Research*, 17(4):374–391, 2006.