Requirements for Business Process Legal Compliance Monitoring

Oleg Svatos
University of Economics in Prague
Faculty of Informatics and Statistics
oleg.svatos@vse.cz

DOI: 10.20470/jsi.v8i2.301

Abstract: In this paper we focus on modeling and data collection requirements when thinking of business process legal compliance monitoring. The aim is not to suggest concrete implementation but to provide complete overview what it means to monitor legal compliance and what models and data it requires. Unlike most of the current research we try to look at this issue from global point of view so that we can provide requirements that reflect the whole complexity of this topic and not only specific requirements of one technological platform.

Key words: business process compliance, business process management, process state diagram

1. Introduction

Business process management has over the last years become one of the important trends in the business management in all branches of the industry. One of its topics, which importance rises as regulations intervene more and more into business process operations, is the business process compliance with regulations in force.

The important characteristic of regulations, a business process has to be compliant with, is that they not only bring into the business procedures, which have to be obeyed, but also their enforcement, which is done by a third party (usually public offices or courts). This puts emphasis on carful and regulation compliant business process management, since, if anything goes wrong, it may not only hurt the performance of the business processes, but it also may result in a public proceeding or a lawsuit. The level of regulation varies and it is different for each business process, depending on how much a regulator or involved parties felt that they had to make the accomplishment of a business process or its parts judicially enforceable. The level ranges from simple manufacturing business processes bound by supply or delivery agreements to business processes of public administration that are completely defined in regulations.

Compliance of a business process with relevant regulations can be checked at the time of modeling or any time later when it is being performed. The modeling stage is very important stage where the process compliance should be ensured first, but as the related research shows (Van der Aalst, 2011), Ly, Maggi, Montali, Rinderle-Ma & van der Aalst (2015), that the models tend to be closer to ideal reality and the real business process performance is much more complex and usually behaving little bit differently. That is the place where the business process legal compliance monitoring enters the game and becomes so a crucial part of business process management.

The compliance topic is discussed in current research a lot, but we would like to approach this topic from a little bit different perspective than the current research does. We think that the focus on technologies, technological detail and business process execution prevents from looking at this topic from broader perspective mainly because of the following reasons:

- There is different level of detail of the business processes, at which they are being managed by the management, and at which they are modeled for execution. Certain level of abstraction is necessary in order to keep the models easy to understand to the management. The detail of the business processes designed for execution is for the management members too high.
- Not all business processes are suitable for workflow engines. Analysis based only on business processes for execution underestimates the real world business process complexity.
- Legal compliance is complex problem that significantly overlaps the borders of one IT system and so certain level of abstraction is necessary.
- There is usually significant heterogeneity of the environment business processes go through. Some parts of one business process may be part of one workflow engine, some of another
engine and most of it just supported by relevant applications individually without any workflow. Focus on one technology neglects the real world complexity.

The aim of this paper is to provide general overview what it means to monitor legal compliance of business processes and what it requires in terms of models and data. Unlike the current research, which focuses mostly on selected technological details, we aim to bring complete conceptual framework for the business process legal compliance monitoring topic. Concrete implementation is out of scope of this paper and so we stay in our analysis at conceptual level abstracting from technological details.

2. Stages of Compliance

As stated in the introduction the compliance of a business process with relevant regulations can be checked at the time of modeling or any time later when it is being performed.

Kharbili, de Medeiros, Stein, & van Der Aalst (2008) together with others like Sadiq & Governatori (2010) split implementation of compliance checking mechanism, with focus on business process model execution, into three stages: design-time compliance checking, runtime compliance checking and backward compliance checking.

On the conceptual level we can reduce this categorization to two phases, as the middle phase “runtime compliance checking” is related directly to technological process execution, we try to abstract from at the conceptual level. It makes sense at technological level where run time and post runtime make difference in used technologies but at conceptual level it does not.

Most of the relevant research has been done in the area of the checking models for regulation violation at very detailed technological level – business processes ready for execution. The evaluation of this research we can find in Becker, Delfmann, Eggert, & Schwittay (2012) or in Svatoš (2013b), which findings are similar. They state that the current approaches mainly focus on special modeling techniques and/or a restricted set of types of compliance rules. Most approaches abstain from real-world evaluation, focus on single technology driven issues, and that raises the question of their practical applicability. Svatoš (2013b) in addition suggests his solution in form of a new business process modeling language.

The research in backward compliance checking is done from two perspectives. Regular conformance checking and legal compliance checking through a model modeled in some relevant type of logic.

Regular conformance checking is well established through process mining, where the business process model fitness towards the reality is evaluated (Van der Aalst, 2011). This type of conformance checking is not sufficient for complete legal compliance monitoring as Svatoš (2013b) shows that the regular process modeling languages are not capable of capturing all conditions a regulation may put on a business process.

Compliance verified on basis of logical models provides us with number of options. Its development is driven by missing capability of contemporary process modeling languages to capture explicitly rights and duties imposed by the regulations. Most research has been done in field of deontic logic which recognizes permissions and obligations. Padmanabhan, Governatori, Sadiq, Colomb, & Rotolo (2006) presented a framework in which one can define policies/ business rules using deontic assignments to represent the contractual relationships, i.e. relations among actors (Strategic Dependency models), Governatori, Milosevic, & Sadiq (2006) introduced formal rule language called Formal Contract Language (FCL) based on deontic logic or (Siena, 2010) presented the Nomos framework, again, based on the deontic logic. Data from reality is then used as facts and the rules defined in the logical model are then used to evaluate its compliance. The weakness point of this research is the primary focus on differentiation permissions and obligations in case of the activities, leaving the other compliance rules with almost none attention. Deontic logic is not the only one applicable. Compliance verification can be also done using regular first order logic (Svatoš, 2012) or temporal logic (Förster, Engels, Schattkowsky, & Straeten, 2007), (Šabatová, 2015). Important factor is that the combination of model symbols and logic has to be able to capture all relevant conditions a regulation may put on a business process.

As stated in the introduction the importance of compliance monitoring, i.e. backward compliance checking, grows. Designed business process models are ideal visions of reality (Van der Aalst, 2011) and when they are deployed into reality, which is far more complex, usually the very strict implemented rules are softened so that necessary flexibility is enabled. This way the users are allowed
to do decisions which may result in regulation violations even though it was not possible in the original business process design. We do not argue this is necessarily wrong – the models have to be simple easy to understand, which is one of reasons for our choice of conceptual level. We believe that is more beneficial for the compliance monitoring, to use simple conceptual models, which capture the business process as set milestones that have to be reached, rather than detailed descriptions how the business process should be performed as it is required when modeling for business process execution. This is where we would like to extend the current research.

3. Backward Compliance Checking

In order to be able to evaluate business process legal compliance, the backward compliance checking requires two things: models, which capture how the reality should behave like and how the regulations are affected, and the relevant data, which show how the reality actually behaved. Our focus is not on the technology we need to use to gather the data, while there is great number of studies on how to store the required data, how to make the available in real time etc. We rather start with specification what one exactly needs in order to be able to check the legal compliance. It is not about gathering all data available, but about getting exactly only the required data and about having the proper models to have something to compare the actual business process performance to.

![Fig. 1: MMABP Model of Reality (Řepa, 2007, p. 196)](image)

Related research like Sadiq, Governatori, & and Namiri (2007) or Becker, Bergener, Delfmann, Eggert, & Weiß (2011) lists four categories of compliance rules we should be focused on: flow, data, resource and time.

We recognize this categorization but we do not think that all these compliance rules share the same common base for compliance rules modeling - especially the data.

The data are different point of view at the reality than the other three categories. Foundation for this differentiation we can find in Řepa (2007) in the Fig. 1.

On basis of the Methodology for Modeling and Analysis of Business Processes (MMABP) approach to modeling we can differentiate between two interconnected perspectives of compliance: business process compliance and object compliance. Each of these has different modeling and data requirements.

Business process compliance focuses on compliance of reality with modeled business process and legal requirements on behavior, captured either as a part of the business process model or by some separate model on basis of deontic logic. This includes the mentioned flow, resource and time rules.

Object compliance focuses on compliance of the structure and state transitions of the real objects with legal requirements on objects, captured in the class diagrams and state transition diagrams. This fits the data compliance rules.
What concrete models to use and what data is necessary to get we will discuss in the following two sections. First there are discussed specific requirements for models and the modeling languages used. Afterwards there is discussed what data the models require in order to be able to use them as a mean for compliance evaluation. In each case we will suggest what concrete modeling language to use and what requirements the compliance monitoring puts on modeling or data retrieval in case of the selected modeling languages.

4. Model Requirements

First step for the compliance monitoring is to have a model which captures how the regulations fit into the business process and how the reality should look like in order to be compliant with the regulations. This requires from modeling languages to be able to capture clearly and completely the relevant requirements the regulation imposes in the model.

In accordance with the discussion in the previous section, we can divide the modeling requirements into two following sections.

4.1 Business Process Compliance

The question what process modeling language to use is not trivial. First it is necessary to decide what level of detail should the model be and for what purpose. Whether it is going to be very detailed process model, which can be then executed by some workflow engine or conceptual model, where most of the technical detail is being abstracted from and which serves as a tool for the management. As we have reasoned at the beginning of this paper, we consider the focus on a conceptual model more appropriate for this topic than focus on detailed, technology bound business process models.

After specifying the level of detail, there is necessary to find business process modeling language that allows precise capturing not only of the flow of activities in relevant detail but also the rights and duties from the regulations, which are undividable part of compliance monitoring. The final model has to provide us with sort of a map (Van der Aalst, W.M.P., 2009) that specifies how the regulations affect managed business processes: what duties and rights of the parties involved have to be taken into account when performing the business process, how the business processes should be performed in order to be compliant or, when there is violated some regulation, how to get back into compliance if possible. Another option is to create a separate model, which is based on for instance deontic logic and to use it in combination with a regular process modeling language like the BPMN (Object Management Group, 2013).

The main issue is that the currently popular business process modeling languages do not support explicit capturing of rights and duties, and so there were introduced many extensions based on deontic logic like above mentioned FCL, PENEOPE or NOMOS. Summary papers like Becker, Delfmann, Eggert, & Schwittay (2012) or Svatoš (2013b) do not find their results satisfying and for instance Svatoš (2013b) recommends instead of focusing on attempts to extend the popular business process modeling languages to use specially tailored business process modeling language for this purpose - the Process State Diagram (PSD). This recommendation and also its focus on conceptual modeling instead of process execution led us to decision to consider it as the basis of our analysis. The PSD not only provided us with ability to capture all relevant information (including the rights and duties) necessary for monitoring legal compliance of business processes in one business process model (Svatoš, 2013a), but also with necessary integration with state machine diagrams as depict in Fig. 1. Its construction based on first order logic also enables possibility of automated reasoning based on facts and rules represented by the model.

4.2 Data Compliance

As noted at the beginning of this in section, the complex approach to monitoring business process legal compliance requires not only business process models but also class diagrams and object state transition models. Their purpose is to capture the required structure of objects, associations and conditions of object state transitions imposed by the regulations so that the real objects and their state transitions can be validated against them.

In case of the class diagrams and state transition diagrams the selection of what modeling language to use is simple as there is one dominant and widely accepted standard in this area - the UML (Object Management Group, 2015), which includes Class diagrams and State machine diagrams.
Class diagram is well equipped for capturing compliance rules in form of an object structure definition, definition of its attribute data types and definition of possible object associations with other objects including the cardinalities, against which the real objects can be validated. In order not to get confused, the class diagram definition is not the original definition from the source IT system, but conceptual definition of characteristics which the real validated object has to fulfill in order to be compliant with a regulation.

State machine diagram is a solid tool for validation of regulated object states and validity of their transitions. State machine diagram provides the users with the ability to capture lifecycle of an object, including the conditions, that has to be fulfilled so that an object state transition may occur. The condition, which represents the condition imposed by the relevant regulations, may not only contain the event which is the initiator of the transition but also other conditions which have to be fulfilled. Conditions may be captured by a formal language specially developed for this purpose - the Object Constraint Language (Object Management Group, 2014). Alternatively we suggest possibility of using the class diagram for condition definition which describes how an object has to like so that the transition may occur in compliance with the relevant regulations. The point of time when a condition of object state transition should be true is quite exact – when the initiating event occurs. In case of the class diagram the point of validation is not that clear since it only defines the structure requirements. Our suggestion is then to make the validation of the real objects against the class diagram definitions as a part of the object state transition conditions. This way it is obvious at what time the evaluated real object should meet the class diagram definition.

5. Data Requirements

There are two main issues bound to data requirements: what data we need to collect and what data is available.

First of all, the used modeling language specifies what type of data interests us as its ontology tells us for which data to look in reality. In general, looking at the Fig. 1, we can differentiate three models, each with option of different modeling languages. We will go over these in the following two sections, and analyze the data requirements for legal compliance monitoring on the basis of the modeling languages, we have selected to be the most appropriate for each model in the previous section.

Primary sources of data, which we can consider throughout our analysis, are the IT systems. As the goal should be computer aided compliance monitoring, we depend on having all necessary data there, otherwise they are undetectable for us. Still, having all the possible date available in some IT system does not mean automatically a success. The effort is complicated by the fact that there are usually heterogeneous technological environments that use different and usually independent workflow engines or systems. On top of that many activities are done outside of any workflow engine only with support of concrete applications. Data collection for business process legal compliance monitoring then requires management of different sources of data and different meanings of the data received. It is important to map the models and the data correctly, taking into account possible different meanings of data produced by different IT systems and the meanings of symbols of the used modeling language.

Heterogeneous environment also brings another important issue for business process compliance monitoring. How to differentiate one business process instance or one object instance from another? We have to help ourselves with the data and identification which is already available. Usually throughout one system we can track a business process and the data it uses by its system surrogate keys, but over several IT systems we have to incorporate also natural keys (Kimball, Margy, 2013) that prevail over the different environments (for instance a contract number). As guiding posts should serve us data identification on input or output of an activity, business process surrogate keys when the part of Process Aware Information System (PAIS) (Dumas, ter Hofstede, van der Aalst, 2005) and natural keys of objects. All these have to be taken into account when preparing the data requirements for business process compliance monitoring.

In accordance with the previous section, we can divide the data requirements topic into two sections.

5.1 Business Process Compliance

In general a business process model consists of actions, states and events (Řepa, 2007) that can be supplemented by activity associated objects. First we will go over these in general way; later on we will discuss them on concrete example of the PSD.
5.1.1 Actions
Actions, or activities, are the basic stones of a business process model as they capture how one action should follow another. When looking for data in reality, against which we could test whether some action was performed or not, we have to take into consideration, that the actions are not as simple as they seem. They are not just symbol for objects’ methods. When we look at UML Activity Diagram or BPMN, we can see that an action not only starts and finishes successfully, but it also may fail or be cancelled. If we take into consideration also an action implementation in business process execution frameworks as tasks, they can be even suspended, continued, skipped, etc. (for a task lifecycle see Weske (2012)). This means that when we talk about monitoring an action we have to talk about monitoring its lifecycle. An action is always at some of its states and the monitoring is about monitoring how an action changes its states (events indicating this change) overtime and whether these events fit the predefined models. We will continue the discussion of these in the events.

5.1.2 States
In a business process model we can find references to states primarily in conditions of decisions (splits and joins of the sequence flow) or final states of a business process. There are two ways how we can treat the concept of state in the business process modeling. One refers to states from an object, activity or other business process element lifecycle, the other refers to a fact that an object’s attribute has some value. Object states are just attributes with special meaning as the actual object state is stored as an attribute value. Both these perspectives have to be taken into account when gathering data for business process compliance monitoring as it is important to follow in what state the business process elements are, and what values objects’ attributes have so that it is possible to evaluate compliance of decisions done in splits and joins of sequence flows.

5.1.3 Events
Events represent process initiators and also synchronization points of the business process with its environment (Řepa, 2007). They signify a time related event, an object state change, an activity state change or other business process element state change that has occurred at given time.

IT systems can provide us with two types of events: primary and derived (Steininger, 2009).

- Primary event is such event that is identifiable in the actual real world and it is not result of interpretation of sequence of other events.
- Derived event is such event that is result of interpretation of sequence of primary events, derived events or combination of primary and derived events.

We can illustrate this differentiation on an activity. In reality when we observe an activity, we can say without any context about it only two things - whether is someone performing it or not. When we look at different states an activity may go throughout its life in process modeling languages like BPMN, PSD or UML Activity diagram there are many more states. These states are derived since they are dependent on other events and their interpretation. For instance, how do we know that an activity has been accomplished? The objective of the activity was achieved. If not and it is not possible any more, than it has failed, or, if someone or something cancelled the activity, it has been cancelled. All these states represent finished activity and only the relevant events and their interpretation can tell the difference among them.

The source of the derived events can be either the interpretation of the model done while checking the reality compliance with the rules captured in the business process model or the PAIS - workflow engines but also single applications aware of the business process management. The important thing is that the derived events originating in PAIS are result of encoded rules within the software that interpret the primary events inside the system. In case of the business process compliance it is important to consider whether we trust the encoded rules and work with the derived events or we rather check the encoded rules too and work with the primary events. This decision puts different requirements on what detail has to be captured in the business process model and what data has to be collected.

5.1.4 Activity Associated Objects
Most of the business process modeling languages allow capturing of additional activity context. This usually includes actors, input or output data etc. These associations do not have an impact on the business process flow, but, from the legal compliance monitoring perspective, they are important for
unambiguous activity instance identification in the reality and monitoring compliance with resource regulation rules. The proper definition in the model and having available this data with the collected activity events is crucial. An activity performance in reality is not then validated only against modeled activity events, but also against its associated context.

The analysis done in this section shows that in order to be able to monitor business process compliance there is necessary to collect in reality two types of data. Time stamped events, which should not only contain information on the events itself but also information on the associated objects, and states of elements referenced in the business process model including actual values of attributes in case of the object elements.

We will review these general requirements on the example of the PSD in the next section.

5.1.5 PSD Data Requirements

In section 4 we chose the PSD to be the appropriate business process modeling language to base our analysis on. In this section we will try to define data requirements specifically for the PSD.

PSD works with explicit events and states of four concepts (Svatoš, 2012) defined in a lifecycle for each of them:

Activity lifecycle consists of ten basic states. Before an activity is performed, it can be performable, not performable or skipped. Activity is performable when there are no objections that would make an activity performance impossible, an activity is not performable when there are objections which make an activity performance impossible or an activity may be skipped since even though the performance of the activity was scheduled, it was for some reason skipped. Events, indicating change to one of these three states, are derived and their occurrence is result of an interpretation based on sequence of relevant events.

When an activity is performable it may be started. Once started it may be suspended and continued afterwards. An activity is started when an actor starts performing the activity or an interpretation of relevant sequence of events says so. An activity is suspended when an actor stops performance of the activity which has not been accomplished, failed or cancelled yet or an interpretation of relevant sequence of events says so. An activity is continued when the activity was in suspended state and an actor continues performing it or an interpretation of relevant sequence of events says so.

When an activity is started, suspended or continued, it may finish through three different states. An activity is accomplished when performance of the activity has reached its goal indicated by occurrence of defined events. An activity is failed when performance of the activity is stopped since the goal became impossible, again, indicated by occurrence of defined events. An activity is cancelled when event, which makes the actor abandon the performance of the activity, occurs.

Finished activity can be undone. An activity is undone when states of all relevant objects are the same as they were before the activity was performed i.e. when an interpretation of relevant sequence of events says so.

The activity states overview shows that all events representing change of an activity state can be derived and only few of them (started, suspended and continued) can be in role of primary event even though their interpretation may be dependent on the actual activity state.

An activity may have associated objects, which describe an actor or data input/ output.

The time point lifecycle consists of two states. First it is set the time that represents point in time. When the set point in time occurs, the state of the time point changes to final state occurred.

The lifecycle of time limit consists of six basic states. First it is set the time that represents the period length. Immediately as the time limit is started, the state of the time limit changes to started. When started or continued, it can be suspended or it may pass. When suspended it waits to be continued. When the time limit is started, continued or suspended it can be also reset and then it would wait to be started again.

All events representing change of a time point (set, occurred) or time limit state (set, started, suspended, continued, reset, passed) can only be derived, with the exception of events occurred and passed. These two may be also primary as they may also occur as a result of natural time run and not only as a result of some interpretation of relevant sequence of events.
**Norm** is a concept which represents rights and duties of individual parties taking their part in the business process. Its defined lifecycle allows monitoring whether the parties act according to the imposed duties and rights or they do not (see Svatoš, 2013a). The lifecycle consists of four states. First the norm comes into effect. When in effect it may be revoked, realized or not realized. All events representing change of its state are derived.

A norm may have associated object that identifies the person in whom is vested the given right or duty.

**Objects** do not have standardized lifecycles as the other PSD concepts since the objects are individual for each case and concept and unlike the other three concepts they can have more than one lifecycle (Řepa, 2012). Events representing change of its state can be primary or derived depending on the nature of the object’s lifecycle.

As far as the events are concerned, for compliance monitoring based on the PSD it is not necessary to collect all events mentioned in the model. In order to be able to tell which events are important for compliance monitoring using the PSD, it is necessary to differentiate between internal and external events in the model (Řepa, 2007).

Internal events are such events that originate in the evaluated business process model. In case of the PSD all succeeding events, which are noted at the end of the precedence link, are internal events and immediately as the condition of the precedence link becomes true, they are cast. This also means that all internal events in the PSD are derived as they are result of interpretation of sequence of events.

External events are such events that originate outside the evaluated business process model. Evaluated business process is started with such event and it also has to synchronize itself with these events during business process performance if required. In case of the PSD these are such events, which are part of the condition noted at the beginning of the precedence link and do not have corresponding casting event noted as succeeding event at the end of another the precedence link in the same business process model.

When evaluating process compliance using the PSD, the minimum required data for the model evaluation are the external events, which can be either derived or primary. Depending on the level of compliance monitoring strictness, we have to consider whether the derived events are sufficient for the compliance check or it is necessary to go deeper and look for the primary events, on which basis the source IT systems do their interpretation and deliver the derived events. As the business process model has to contain all important information for the business process management i.e. to be complete, it is usually just matter of our ability to collect such data from the IT systems.

Internal events are implicit parts of the business process model against which is the reality validated and in case of the PSD it is not necessary to gather these from IT systems unless we need to check that the IT system does the interpretation (i.e. condition evaluations of precedence links ending with an internal event) according to the business process model. In this case we have to be careful since it may happen that a business process model contains events which are not part of any IT system since they are not necessary for the business process implementation. One of the clear examples in the PSD is the norm. One can find the events of norms usually only in the business process model as they are not important for business process implementation, but they are important for its legal compliance management. The events of norms are then evaluated internally during the business process compliance evaluation on basis of collected external events and the rules captured in the business process model. This can be the case also of some other objects and their lifecycle events.

In summary, in case of the PSD events there are required data in form of time stamped events (all external events and optionally internal, which are implemented in the IT system) indicating state transition in a lifecycle of an activity, object, norm or time point or time limit, which has to be accompanied by associated objects (appropriate version to the timestamp of the event) in case of events indicating change of an activity state or a norm state.

In case of the states the PSD works not only with states related to an activity, object, norm or time point or time limit lifecycle, but also, in case of the object, with all object attribute values, which may be referenced in conditions noted at the beginning of a precedence links. If referenced in the business process model, their actual values at the time of condition evaluation are required too. As a consequence of internal/external event differentiation discussed above, it necessary to differentiate which lifecycles are present only in models and which are also implemented in the IT systems. It should not happen that a lifecycle would be partly in IT system and partly only in the model - the lifecycle would become untraceable. All lifecycles, which are relevant for compliance monitoring, have
to be captured in a state transition models and these models have to be complete. The ones, selected for implementation in the IT systems, have to be then implemented in them completely. This also answers the question, which internal events can be collected from IT system e.i. those that are indicating a state transition in one of the implemented lifecycles in the IT system.

In case of a lifecycle, which is only in a model, the checking mechanism has to keep the track of the current state of concepts (for instance the norm) and provide it when referenced in a condition.

5.2 Data Compliance

As far as the data compliance is concerned, we assume that the modeled world consists of objects (Fig. 1). The required data from IT systems for data compliance evaluation are then the real objects with attributes and their values that are referenced in the models which depict the data regulation rules. Same as in the business process compliance, there are required only data of those objects which are also implemented in the IT systems.

We will discuss the data requirements of data compliance on the basis of the two selected models in section 4.

5.2.1 Class Diagram Requirements

The Class diagram allows evaluation whether the actual object structure and associations fit the modeled definition. There is validated existence of defined object attributes, attribute values according to defined data types and validity of references to other objects and from other objects to the validated object according to the defined cardinalities of the associations. In order to validate an object’s structure there is necessary to collect data on object class name, attribute names and values. For validation of object relations there is necessary to collect data on names of associated objects and their class names.

Class diagram captures the structure of reality and therefore there is not necessary to collect events for its compliance evaluation. Application for the Class diagram we can find in the conditions of the object state transitions in the State machine diagram.

5.2.2 State Machine Diagram Requirements

The State machine diagram allows validation of object state transitions compliance. For instance when a contract is about to become valid, the conditions bound to the state transition in the model state, what properties such contract has to have so that the contract can really become valid (from regulation perspective).

In order to evaluate validity of an object state transition there is necessary to have data on the event initiating the object state transition, current state of the object and values referenced in the state transition conditions i.e. values of referenced objects’ attributes, states of other concepts (when integrated with the PSD) and values necessary for class diagram evaluation; all data versions as of the timestamp of the initiating event.

The listed data requirements show that character of collected data is almost the same as in the business process compliance case (events, states, attribute values of objects). The difference makes the inclusion of the Class diagram as a part of the state transition condition. This extends the data requirements by the ones specified above. Existence of the primary and derived events differentiation and its consequences mentioned in the business process compliance has an effect on State machine diagram evaluation too. Careful decision, which events can be derived and which events should rather be primary, is necessary.

6. Conclusions

In this paper we have looked at the business process compliance from little bit different perspective than most of the current research does. Instead of looking at the technological detail, how business process legal compliance can be implemented in ideal environment, we provide more complex point of view, which abstracts from the technological detail and focuses on the questions which precede the actual implementation. What legal compliance monitoring requires in terms of models and data when taking the complexity of the reality into account.
We started with discussion of the contemporary research done in the business process compliance and identified two stages at which the conceptual business process compliance can be done: the design-time compliance checking stage and the backward compliance checking stage. We have reviewed the current state of research on the design-time compliance checking, which is currently the place of interest of most of the researchers, and backward compliance checking, which is studied in current research in limited extend.

As we have stated our focus on compliance monitoring at the beginning of the paper, we have focused on the backward compliance checking in the succeeding analysis. On the basis of related research we have identified two perspectives from which one can look at the complex legal compliance monitoring: business process compliance and data compliance. Each of these has different modeling and data requirements.

Having the legal compliance split into two perspectives we have continued the analysis accordingly.

We have discussed the model requirements for both of the perspectives and suggested concrete modeling languages which fit the specified requirements the best. In case of the business process compliance there is emphasized importance of ability of the business process modeling language to capture construction specific for regulations mainly the explicit depiction of rights and duties. As the appropriate business process modeling language was chosen the Process State Diagram (PSD), which was evaluated in the related research as well fit for the specified requirements. For the data compliance there was required ability to capture structure of objects, associations and conditions of object state transitions required by the regulations. For this purpose there were selected standard modeling languages: the UML Class diagram and the UML State machine diagram as their capabilities fit the requirements.

After the discussion on models and their requirements we moved on to the data requirements specification. In case of the business process compliance there were analyzed standard concepts the business process modeling languages consist of and specified two types of data that are necessary to collect for legal compliance evaluation. Time stamped events, which should not only contain information of the events itself but also information on the associated objects, and states of elements referenced in the business process model including actual values of attributes in case of the object elements. This concept was extended in concrete example of the PSD. There were identified events, which have to be collected in order to be able to perform business process compliance evaluation and discussed requirements on lifecycles and their implementation in the IT system. The data compliance requirements built on the findings in the business process compliance and extended the required data only with information specific for structure compliance evaluation i.e. Class diagram evaluation.

Further research has to be done in bringing this conceptual framework into concrete technological implementation. This paper should provide such research with a point to start from.

Bibliography


Kharbili, M., De Medeiros, A., Stein, S., & Van Der Aalst, W., 2008: Business Process Compliance Checking: Current State and Future Challenges. In Informatics: Modellierung betrieblicher Informationssysteme (Mobs) (pp. 107-113)


Steininger, T., 2009: *Automated Assertion Transformation Across Multiple Abstraction Levels*, München: Technischen Universität München


Svatoš, O., 2013a: Capturing rights and duties in a business process model. *Central European Review of Economic Issues*, Volume 16, Number 2. ISSN 1212-3951


JEL Classification: L20, M15, K0