An Integrated Solution for Runtime Compliance Governance in SOA

Aliaksandr Birukou[1], Vincenzo D'Andrea[1], Frank Leymann[3], Jacek Serafinski[2], Patricia Silveira[1], Steve Strauch[3], and Marek Tluczek[2,⋆]

[1] DISI, University of Trento, TN 38123, Italy
[2] Telcordia Poland, Poznan
[3] IAAS, University of Stuttgart, 70569, Germany

Abstract. In response to recent financial scandals (e.g. those involving Enron, Fortis, Parmalat), new regulations for protecting the society from financial and operational risks of the companies have been introduced. Therefore, companies are required to assure compliance of their operations with those new regulations as well as those already in place. Regulations are only one example of compliance sources modern organizations deal with every day. Other sources of compliance include licenses of business partners and other contracts, internal policies, and international standards. The diversity of compliance sources introduces the problem of compliance governance in an organization. In this paper, we propose an integrated solution for runtime compliance governance in Service-Oriented Architectures (SOAs). We show how the proposed solution supports the whole cycle of compliance management: from modeling compliance requirements in domain-specific languages through monitoring them during process execution to displaying information about the current state of compliance in dashboards. We focus on the runtime part of the proposed solution and describe it in detail. We apply the developed framework in a real case study coming from EU FP7 project COMPAS, and this case study is used through the paper to illustrate our solution.

Keywords: compliance governance, business process, monitoring, SOA, complex event processing.

1 Introduction

During the last decade several companies, such as Enron in US, Fortis and Parmalat in Europe, unexpectedly collapsed. In response to those events, new regulations for protecting society from financial and operational risks of companies have been introduced. The goal of those regulations is to avoid similar bancruptcies in the future, and companies must comply with them. Compliance become more and more important in modern organizations [12]. In this paper, we use the term "compliance" in the sense of the conformance of a company in

⋆ This work was supported by funds from the European Commission (contract no. 215175 for the FP7-ICT-2007-1 project COMPAS).

P.P. Maglio et al. (Eds.): ICSOC 2010, LNCS 6470, pp. 122–136, 2010.

fulfilling compliance requirements, i.e. constraints or assertions that are the results of the interpretation of the compliance sources. Modern organizations deal with three main types of compliance sources: legislature and regulatory bodies (e.g., Sarbanes-Oxley Act, Basel II, Solvency II), standards and codes of practice (e.g., ISO9000, ISO/IEC 27002, internal regulations), and business partner contracts (e.g., licenses of service providers).

The diversity of compliance sources introduces the problem of compliance governance in an organization. Compliance governance refers to the overall management approach for controlling the state of compliance in the entire organization and, in general, consists of: (1) selecting the sources to be compliant with and designing corresponding compliance requirements; (2) (re-)designing business processes compliant with the selected requirements; (3) monitoring compliance of processes during their execution; (4) informing interested parties (managers, auditors) on the current state of compliance; (5) taking specific actions or changing the processes in cases of (predicted or happened) non-compliance.

There are solutions for automating one or several steps of the compliance governance, i.e. deriving requirements from sources (Global Information Rules Database[1]), modeling and automating design time compliance checks [10], monitoring [17] and informing interested parties [20]. However, the existing approaches rarely deal with different types of compliance sources and cover only a few steps of the compliance governance.

There are several research challenges arising when speaking about an integrated solution for compliance governance: (i) Is it possible to create a system dealing with the whole process of compliance management, from selecting compliance sources to dealing with cases of non-compliance? (ii) Is the service-oriented technology mature enough to be used as the basis for such a solution? (iii) Can we reuse the knowledge about achieving compliance within the company, or, even, across companies?

With the research challenges above in mind, we propose an integrated solution for runtime compliance governance in SOA. The framework is based on the service-oriented technology and includes tools for: modeling compliance requirements for different compliance sources; linking the requirements to the business processes; monitoring process execution using Complex Event Processing (CEP); displaying the current state of compliance in a Compliance Governance Dashboard (CGD) and analyzing cases of non-compliance in order to find what causes such situations. In the description of framework we focus on the runtime aspects, such as process execution and monitoring, but the design-time aspects (modeling processes and requirements) are also briefly described. For a number of issues (besides technical issues there are also organizational issues, legal responsibility, acceptance of an active role of the technology in the work practices), in this paper, we do not address the issue of taking specific actions for achieving compliance (also known as *enforcement*) and process re-design. This topic deserves dedicated research. Therefore, our framework covers selection and modeling compliance requirements and business processes, monitoring the compliance

[1] `http://www.grcroundtable.org/grc-grid.htm`

at runtime and informing interesting parties on the state of compliance. The framework and the prototypes of the licensing Domain-Specific Language (DSL) for expressing compliance requirements, the business process engine, CEP-based monitoring tool, the warehouse, the dashboard, etc. have been applied in a real case study in the context of the EU FP7 project COMPAS[2] (Compliance-driven Models, Languages, and Architectures for Services). The case study focuses on checking compliance of telecom service provider to licenses of its business partners.

This paper is continuation of our work on the compliance governance. Previously, we introduced: compliance governance lifecycle and conceptual model [9], which we adapt in the presented framework; a model-aware repository and service environment (MORSE) [25], a licensing DSL [3], an approach for developing compliance governance dashboards [20], and algorithms for root-cause analysis [7], which are used withing the proposed framework. This paper connects the proposed pieces within an integral runtime compliance governance framework and shows how the whole framework is applied in the case study scenario.

The paper has the following structure: in Section 2 we review existing approaches for compliance governance in SOA. Section 3 introduces the scenario we use through the paper to illustrate our solution. Section 4 presents the compliance governance lifecycle in an organization, while Section 5 presents our solution for runtime compliance governance, according to the considered lifecycle. We conclude the paper in Section 6.

2 Related Work

Our approach is different from related work as it enables the adaption to various domains of compliance by extending the conceptual model for compliance governance introduced in [9] and customizing the related components in the compliance governance architecture accordingly. We deal with the domains of Quality of Service (QoS), security, and licensing, while most of the existing approaches in the field of compliance governance in SOAs are focusing on one single specific compliance domain. For example, the approach presented by Kuster et al. [13] is limited to the compliance of business processes with respect to data object lifecycles. A data object lifecycle is specified as a model, which captures allowed states and state transitions for a particular data object. The generated process model complies to the object lifecycle based on automata theory.

Most of the scientific publications regarding compliance involves annotation of business processes. For instance, Wolter and Schaad [27] investigated an extension for the Business Process Model And Notation (BPMN) [19], enabling the modeling of task-based authorization constraints and supporting resource allocation patterns such as separation of duties and role-task assignments. In contrast to our approach, this later focuses on task-based access control, which is a subtopic of the compliance domain regarding business process security. Sadiq [23] presents an approach based on a formal contract language to specify and describe

[2] `http://www.compas-ict.eu/`

compliance constraints, and to define compliance rules to annotate business processes. Namiri et al. [18] propose a semantic-based approach for modeling and implementation of internal controls in business processes, focusing on the separation of business and internal control processes. An approach focusing on the integration of semantic constraints in process management systems and its usage for the verification of the integrated semantic constraints is introduced in [14]. Those approaches only consider the modeling phase of compliance constrains or controls, lacking support for runtime compliance checking and monitoring.

The current studies involving policy-based frameworks are also restrict to the modeling phase and far from having a full and well defined framework to manage compliance. They have been extending and integrating semantic of business process and compliance policies in the form of ontologies in order to provide compliant business process [15], [16]. In fact, the same lack of completeness is also present when policy frameworks (e.g., IETF, Ponder, KAoS, Rei and WS-Policy) are adopted to manage compliance in SOA as describe in this survey [26]. Hence, a lot of open issues are still around in the compliance field.

The work of Governatori et al. [10] checks compliance of business process to regulations. They propose a framework for assessing if a given business process complies with a set of regulatory control objectives. The compliance governance framework proposed in this paper aims at an integral management of compliance of all business processes in an organization. Differently from Governatori et al., whose framework provides diagnostic support for business process design, our framework focuses on the aspect of compliance of process instances, with the current status of compliance being updated on dedicated CGDs.

Business Activity Monitoring (BAM) aims at providing aggregated information suitable for performing various types of analysis on data obtained from the execution of business activities. For example, tools such as Oracle BAM, Nimbus and IBM Tivoli aim at providing their users with real-time visual information and alerts based on business events in a SOA environment. The information provided to users comes in the form of dashboards for reporting on key performance indicators (KPIs) and violations of service level agreements (SLAs). The compliance management part of these tools, if any, comes in the form of monitoring of SLA violations, which need the SLA formal specifications as one of its inputs.

In the context of our research it is worth to mention event-based related work, since our framework checks compliance taking in consideration the content of the events produced during the execution of business processes or as a result of CEP. The following works present solutions to monitor and evaluate process events, but not taking into account their compliance. Michelson et al. [17] presented a complete report overview about event-driven architecture (EDA) in SOA environments. Their content is composed of many definitions and concepts involving events, as well as strategies to process them in a SOA. Additionally, they also describe event flows and the main components expected in an EDA. Many of those components are presented in our solution (e.g., repositories, events, process engine). However, even if with some similarities, the approaches are different, in the sense that Michelsons work does not focus on and mention compliance.

Sriraman et al. [24] also claim the business utility and agility provided by the union of SOA, EDA and model driven architecture (MDA). They present different perspectives containing SOA, EAD, and MDA together with different domains (e.g., user, development, business) and views (e.g., user centric view). They also show how to implement the proposed architectures in Java. However, also this work does not explicitly comment or focus on event-based compliance monitoring. Still, both paper are important to understand the role of events and how they can be useful in a business process environment.

Giblin et al. [6] propose a compliance meta-model for uniform description and management of compliance policies and show how subsets of compliance sources, expressed in terms of the meta-model, can be (semi-)automatically transformed into event monitoring rules. While the experience of authors in generation of rules from requirement is definitely useful for this step in our framework, we go beyond this, providing runtime monitoring and informing interested parties on the state of compliance.

Robinson [22] proposes a generic framework for defining, monitoring, and modifying (based on feedback) requirements in information systems. This work lies in the area of system verification, while our framework rather deals with compliance to requirements coming from different sources.

3 Motivating Scenario: Advanced Telecom Services

In this section, we describe the Advanced Telecom Services scenario we use through the paper to illustrate our solution. This scenario is one of the case studies of the EU FP7 project COMPAS. The scenario deals with a service "WatchMe" that provides customers with on-demand aggregated audio and video streaming content. Service clients can use the service to see videos with soundtracks in different languages. This service is provided by a fictitious company called Mobile Virtual Network Operator (MVNO).

The case study focuses on particularly challenging environment: a provision of advanced telecom services by a mobile operator that does not have its own network, but uses existing networks of other operators to provide services. Therefore, network infrastructure and many applications that provide the MVNO service components are owned and managed by different enterprises, which include third party application providers, network carriers, and the MVNO company. We place the proposed architecture inside the MNVO company for managing and monitoring the compliance with the licenses of content providers.

In this scenario, the WatchMe service serves as a content aggregator placed between customers (cellphone owners) and the audio and video streaming third party providers. For example, customers access the WatchMe service to see sport events with audio comments in the language they prefer. The service processes customer requests and provides streaming of the selected audio and video content. In the scenario, we assume the MVNO company is providing synchronization between video and audio. The process describing the services offered by the company (presented in Figure 1) includes the following operations:

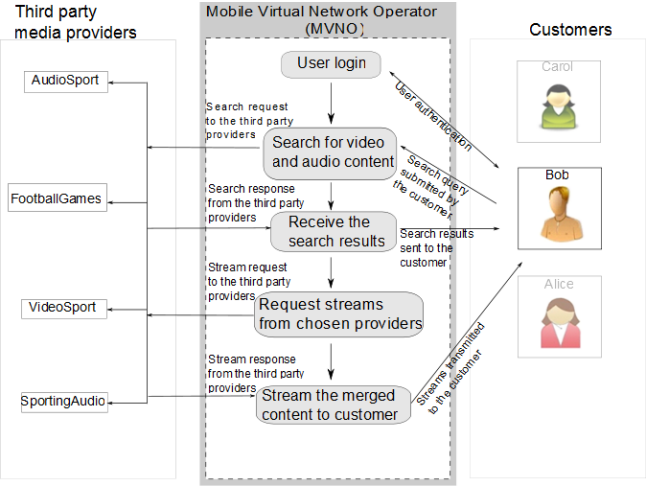

Fig. 1. The business process of the WatchMe service

- authorization of a customer,
- processing search queries for audio and video streams received from customers and forwarding them to third party providers,
- collecting the results of the queries from the providers,
- merging all the results into a single list,
- sending the merged list of results to the customer,
- receiving requests for specific audio and video streaming content from the customer,
- acquiring requested video and audio endpoints from the selected providers,
- receiving streams from the acquired endpoints, merging them online and streaming the resulting content to the customer.

The terms and conditions of using the WatchMe service are regulated by appropriate licenses between MVNO (the WatchMe service provider) and its customers, and between the third party providers and MVNO. In this scenario, we focus on the latter, which is the compliance of MVNO to the licenses of third party providers. Licenses of audio and video providers specify conditions related to various payment plans, as well as to types of allowed compositions of audio and video streams. We consider two payment plans in this scenario. The Time-based plan allows MVNO to acquire and resell any stream for an unlimited number of times in a certain period, based on the amount paid to the media supplier. The Pay-per-view plan allows the company to acquire and resell a certain number of streams based on the amount paid to the supplier, without time constraints. In both plans, the composition permission specifies predefined combinations of video and audio providers, i.e., video streams from VideoSport can only be combined with audio streams from AudioSport, a company from the same media group.

Table 1. Licensing compliance requirements of the Advanced Telecom Services scenario

Compliance Requirement	**Description of Compliance Requirement**	**Control**
Pay-per-view plan	When the WatchMe company subscribes for the Pay-per-view plan it acquires a *limited* number of streams based on *the amount paid* to the media supplier.	When WatchMe company subscribes for the Pay-per-view plan it has to pay *29.90 euro first* and *then receive 300* streams from the media supplier.
Time-based plan	When the WatchMe company subscribes for the Time-based plan it acquires *any* number of times *any* possible streams in a certain period, based on *the amount paid* to the media supplier.	When WatchMe company subscribes for the time-based plan it has to pay *89.90 euro first* and then receive an *unlimited* number of times *any* available stream from the media supplier *in a 30 days* period starting from the contract start date.
Composition permission	*Only pre-defined combinations* of video and audio streams from providers are allowed due to the licenses specified by the video provider.	Video streams from *Football Games* can be *assembled* with audios streams from *AudioSport or SportingAudio*. Videos from *VideoSport* can *only be assembled* with audio streams from *AudioSport*.

All licensing compliance requirements for the business process of the WatchMe service are listed and described in Table 1. For each requirement we list the control, which describes what has to be done to realize the corresponding compliance requirement. The compliance sources from where requirements have been derived are licenses of the content providers. In order to model the requirements, we use Licensing DSL, developed in COMPAS [3]. For the sake of simplicity we focus on the composition permission compliance requirement throughout this paper and use it to show the application of our framework to the Advanced Telecom Services scenario.

4 Compliance Governance Lifecycle

Figure 2 shows the overall compliance governance lifecycle considered in the COMPAS project. The compliance governance lifecycle starts with the step of internalization of the external compliance sources, such as regulations, business contracts, standards. This step is performed by a compliance officer.

The next step is the design or modeling of business processes and compliance requirements that must be met by the processes. At this step, requirements are derived from internalized external sources and also from internal policies defined by the organization. This step involves a process analyst, a compliance officer and a technical specialist.

In COMPAS the compliance requirements are modeled in DSLs [1] using the corresponding DSL Editors. For instance, in the Advanced Telecom Services scenario we use the Licensing DSL [3] , which is an extension of the Open Digital

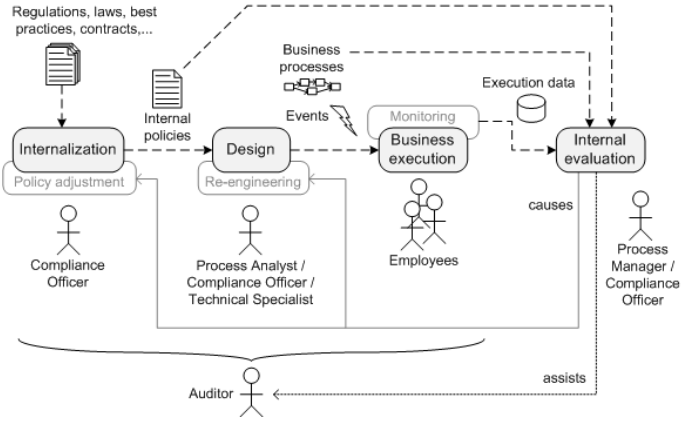

Fig. 2. The compliance governance lifecycle

Rights Language (ODRL) [21], for modeling the composition permission compliance requirement. Other DSLs include QoS [1] and Security [4] DSLs. The processes are specified using the View-based Modeling Framework (VbMF) [11], which is a Model-driven Software Development (MDSD) software framework based on the Eclipse Modeling Framework (EMF). The EMF Models specifying the business process as well as the compliance requirements specified in the corresponding DSLs are the input for the Code Generator, a component integrated in VbMF to generate (semi-automatically) business processes defined in BPEL. In addition to the BPEL process the configuration artifacts, e.g., CEP rules for monitoring components are generated depending on the concrete compliance requirements the execution of the business process has to conform to. The framework currently does not deal with the problem of conflicts and redundancy among the selected requirements, introduced in [5], but, rather, aims at fulfilling all specified compliance requirements. Conflicts and redundancy can be detected at later stages, for instance, applying root-cause analysis.

All artifacts used for the generation of the compliant business process and the configuration artifacts such as compliance requirements, the EMF models, and process models are stored in the Model Repository, which is part of the Model-Aware Service Environment (MORSE) [25]. For the unique identification of each artifact stored in the Model Repository we use Universal Unique Identifier (UUID). Thus this important information might be requested for finding the cause in case a compliance violation is detected during compliance monitoring, by querying the Web service interface of the Model Repository. Finally, the compliant BPEL process containing the UUIDs is deployed in the process engine and the configuration artifacts containing UUIDs are deployed to the corresponding compliance monitoring and checking components.

The third step of the lifecycle is business execution, where employees participate in execution of a business process. During such execution, the process emits

events that are used for the monitoring, and also produces data about process execution. Such data, together with models of the business processes and compliance requirements is used by a process manager or a compliance officer at the fourth step: internal evaluation. During this step the compliance of the process is assessed and the data is analyzed in order to find what causes non-compliance. The results of the analysis assist an auditor and can be also used for process re-engineering and re-thinking of initial requirements. These two latter steps are out of the scope of this paper.

The reader can find the detailed definitions of terms and concepts of the compliance governance in COMPAS, stemming from an effort of the whole team of the COMPAS project at `http://www.compas-ict.eu/terminology.php`. An initial version of the compliance management lifecycle and of the terminology has been presented in [9].

5 Runtime Compliance Governance Framework

In this section we describe the compliance governance framework for monitoring the compliance of business processes at runtime and show how to apply it in the Advanced Telecom Services scenario.

5.1 Runtime Compliance Governance Architecture

Figure 3 shows the components of the runtime compliance governance architecture, described in the following. Runtime governance starts with deploying a BPEL business process that contains the UUIDs of the process model and

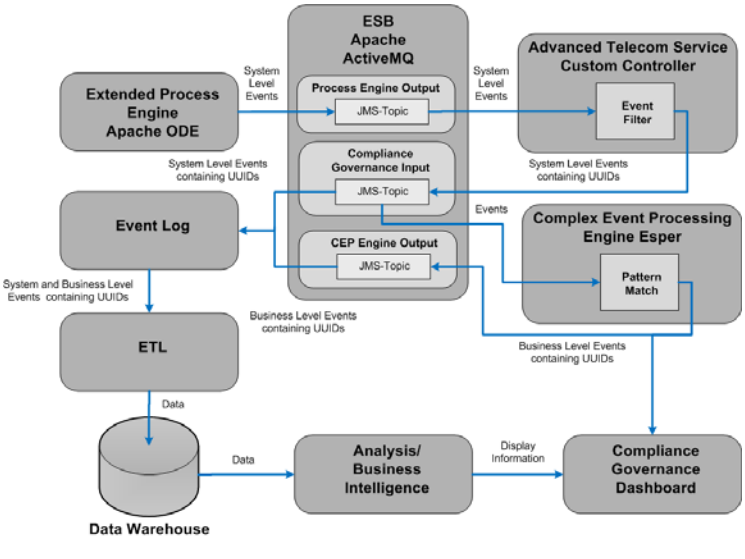

Fig. 3. Runtime compliance governance architecture

those of the activities relevant for monitoring and checking of the compliance requirements to the *Extended Process Engine Apache ODE*. After the deployment a `Process Deployed` system-level event containing the BPEL file of the process including UUIDs is emitted and published to the Java Message Service Topic named Process Engine Output within the *Enterprise Service Bus Apache ActiveMQ*, used as messaging infrastructure. The *Advanced Telecom Service Custom Controller (ATSCC)* is subscribed to this JMS-Topic and therefore receives all events emitted by the Apache ODE. The purpose of the ATSCC is to select pre-defined events, e.g., `Activity Completed` system-level events, emitted by the engine that are related to the deployed process.

The system-level events augmented with the corresponding UUIDs passing the ATSCC internal event filter are published to the JMS-Topic named Compliance Governance Input. Both the *Event Log* and the *CEP Engine Esper* are subscribed to this topic to receive all system-level events relevant to runtime compliance monitoring and checking. The goal of CEP is to provide the possibility for finding complex event patterns within the low-level streams of events generated by the Business Process Engine or/and other Business Activity Monitoring tools. The CEP Engine Esper processes system-level events to create higher-level business-level events, for instance, subtracting timestamp of `ActivityStarted` event from the timestamp of `ActivityFinished` event for the calculation of the duration of an activity. The resulting business-level events also contain UUIDs, which are UUIDs of the CEP rules and generated semi-automatically during design phase using VbMF. Due to the fact that one business process may have to be compliant to several different compliance requirements affecting not necessarily a disjoint set of activities the UUIDs of the monitoring artifacts, e.g., CEP rules are additionally required for the sufficient querying of the Model Repository for drill-down. This enables a unique identification, because the relationship between a concrete compliance requirement and the corresponding CEP rule is always one-to-one as specified in the conceptual model [9]. The results of CEP are shown on the online tab of the *Compliance Governance Dashboard*, allowing for near real-time detection of violation patterns of events, which could lead to violation of any of the licenses signed with their contractors. Therefore, the runtime overhead of using CEP is required for the fast detection of patterns of events leading to violations. Such detection might prevent major financial losses for the company.

The Business Level Events augmented with UUIDs are published to the JMS-Topic named CEP Engine Output. The *Event Log* storing the system-level events augmented with UUIDs and Business Level Events containing UUIDs is subscribed to both JMS-Topics Compliance Governance Input and CEP Engine Output. The *ETL* extracts, transforms and loads the data including UUIDs from the Event Log and stores it in the *Data Warehouse*. After this the *Analysis/Business Intelligence* component retrieves the data from the Data Warehouse and executes the analysis on the data. In case a compliance violation is detected the Model Repository might be queried for drill-down to retrieve the corresponding compliance requirements, EMF models, and CEP rules uniquely identified by the

corresponding UUIDs. Finally, the results of the offline compliance monitoring and checking are displayed in the *Compliance Governance Dashboard*.

5.2 Compliance Governance in the Advanced Telecom Services

In the following, we use the four steps of compliance governance to explain how our framework is applied in the Advanced Telecom Services scenario.

```
<!-- the Scope of rights clause of the license -->
<o-ex:permission>
    <!-- allows composition under conditions listed below-->
    <sl:composition />
</o-ex:permission>
<!-- Financial terms license clause -->
<o-ex:permission>
    <!-- Allowed combinations of audio providers -->
    <o-dd:play>
        <o-ex:requirement>
            <wm:combinations>
                <wm:type>ApprovedAudioProviderOnly</wm:type>
            </wm:combinations>
        </o-ex:requirement>
        <o-ex:constraint>
            <o-ex:context>
            <o-dd:name>ApprovedAudioProviders</o-dd:name>
                <o-ex:constraint>
                <o-ex:context>
                    <o-dd:name>AudioSport</o-dd:name>
                    <o-dd:uid>ASport</o-dd:uid>
                </o-ex:context>
                </o-ex:constraint>
            </o-ex:context>
        </o-ex:constraint>
    </o-dd:play>
</o-ex:permission>
```

Fig. 4. The composition permission expressed in the Licensing DSL for the VideoSport provider

Step 1. Selecting compliance sources and compliance requirements. Figure 4 shows how the composition permission requirement (selected for the running example, as we discussed in Section 3), is modeled in the Licensing DSL.

Step 2. Designing business processes compliant with the selected requirements. The business process is modeled in EMF using the VbMF [1,2]. This EMF model as well as the composition permission compliance requirement modeled in Licensing DSL, as shown in Figure 4, serves as input for the Code Generator component, which is integrated in VbMF.

This step is still under development in COMPAS, the goal is to have a process model annotated with events that will be emitted during the execution. Such events will be used during the execution to check compliance. Currently, attaching events and generating rules requiring to monitor the compliance requirements is done manually. The result of the semi-automatic generation is the business

process in BPEL containing the UUIDs of the process model itself as well as of the activities relevant for compliance checking. Moreover the CEP rules will be generated for processing the corresponding system-level events for creation of business-level events. Additionally the configuration file for the ATSCC specifying the type of events not to be filtered out and the configuration artifacts for the Analysis/Business Intelligence component are generated.

Step 3. Monitoring compliance of processes during their execution. In order to be able to quickly react to any compliance violation, it is essential to monitor business processes online. For this purpose we chose CEP as a perfect solution for efficient and fast detection of events that match violation patterns. Business process engine generates the events at every step of process execution, according to the annotations. A specialized CEP engine catches and uses them for the evaluation of predefined rules. The rules can be used to specify any complex patterns (including temporal logic), various operators (mathematical, logical) and operations for filtering and aggregation. Finally the configuration artifacts are deployed on the corresponding component involved in compliance monitoring and checking and the BPEL process is deployed on the extended Apache ODE.

The following rule for monitoring violations of composition permission is used to detect patterns of video and audio request events that are not compliant with a license.

```
select * from pattern [ every ( VidProvVideoSport = Event
(name = 'WatchMeGetVideoStreamEvent' AND VideoProviderID= 'VideoSport' )
AND ( AudProvAudioSport = Event ( name = 'WatchMeGetAudioStreamEvent'
AND NOT (AudioProviderID = 'AudioSport') )))]
where AudProvAudioSport.sessionID =VidProvVideoSport.sessionID
```

In this case, the pattern includes combinations of `WatchMeGetAudioStream` events from the audio stream of AudioSport and from the video stream of VideoSport for a given session. The query has to match only the events related to the same session (matching is done by "sessionID" property of the events). The system-level events emitted by the ATSCC as well as the Business Level Events generated and emitted by the CEP Engine are afterwards stored in the Event Log as described in Section 5.1. The ETL component extracts the data from the Event Log and loads it into the Data Warehouse. Then the Analysis/Business Intelligence component checks compliance based on the data. In case a compliance violation is detected the Model Repository may be queried in order to perform a drill-down.

Step 4. Informing interested parties on the current state of compliance. The current state of compliance of the processes of the organization is shown in offline and online dashboards. Using the monitoring table in the online view, it is possible to verify event violations detected on the fly and take actions to avoid violations in the future. Such view is mainly used by technical project resources that could change the business process implementation to correct wrong behaviors. Using the offline view, composed of Key Compliance

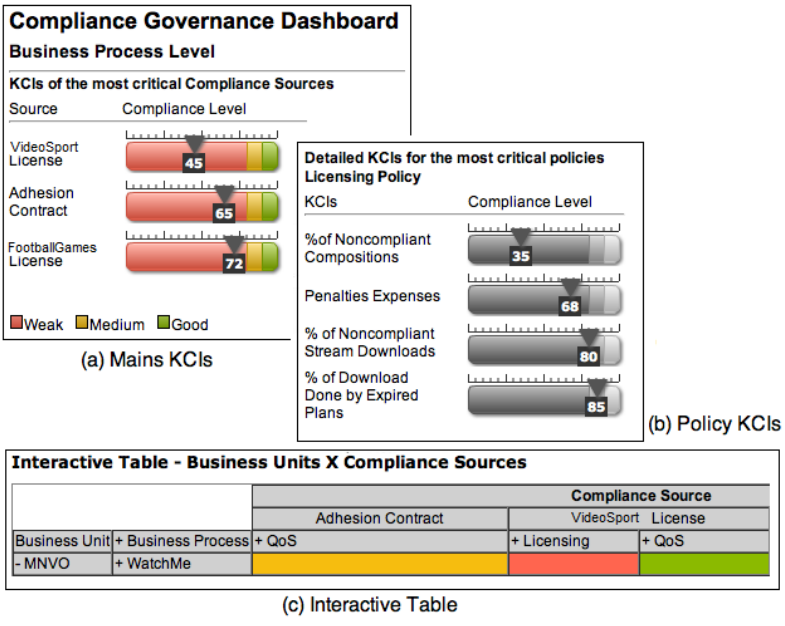

Fig. 5. The current state of compliance of the WatchMe Business Process displayed at the dashboard

Indicators (KCIs) widgets and an interactive table, it is possible to quickly check violations in different perspectives (e.g., business or compliance) and summarization levels (e.g., compliance source, requirement, or policies, which group related requirements, such as licensing requirements). In our example of monitoring the composition permission, ad-hoc KCIs can be defined and their values will be displayed in the dashboard. Having both business and compliance perspective and different summarization levels, it is possible to show high-level information (e.g., KCIs of compliance sources) useful for CEOs and CFOs and low-level information (e.g., list of events violations per compliance requirement) to technical experts. Figure 5 (a) illustrates the KCIs of the different compliance sources from the Advanced Telecom Services scenario in descendant order, where the first widget always contains the compliance source with the highest compliance performance (the worst case). CGD also provides indicators for the compliance requirements concerning licensing (Figure 5 (b)) and an interactive table (Figure 5 (c)). The later also allows users to drill-down KCIs from the highest level information until the lowest level. The values showed by the KCIs are calculated based on the data stored into the Data Warehouse (DW), which were previously temporally stored into the Event log. More details about the CGD design and implementation are available in [20] or at the CGD website[3].

[3] `http://compas.disi.unitn.it/CGD/home.html`

6 Conclusion and Future Work

We have presented an integral framework for runtime compliance governance supporting all the steps of the compliance governance lifecycle: from selecting compliance sources to runtime monitoring and reporting on violations. This addresses the first research question posed in the introduction: *(i) Is it possible to create a system dealing with the whole process of compliance management, from selecting compliance sources to dealing with cases of non-compliance?* In this paper we presented runtime aspects of such a system, while design aspects have been presented in [1], [2].

Since the solution is service-oriented, we also address the second question: *(ii) Is the service-oriented technology is mature enough to be used as the basis for such a solution?* The service-oriented technology seems to be capable of dealing with the matter, since the solution has been tested in a real case study and we are currently working on testing it in another real case study dealing with the loan approval scenario.

Future work includes support of other compliance domains, such as compliance to security or QoS requirements and addressing the third research question: *(iii) Can we reuse the knowledge about achieving compliance within the company, or, even, across companies?* In this regard, we are studying the application of business process fragments [8]. We are also planning applying the presented solution in different settings in order to evaluate its performance and feasibility for real-time business processes.

References

1. COMPAS Deliv. D1.2: Core Meta-models, Templates, and Languages (2009)
2. COMPAS Deliv. D1.3: MDSD Software Framework for Business Compliance (2009)
3. COMPAS Deliverable D5.3: Final Goal-oriented Data Model (2009)
4. COMPAS Deliverable D5.4: Reasoning Mechanisms to Support the Identification and the Analysis of Problems Associated with User Requests (2009)
5. Awad, A., Weidlich, M., Weske, M.: Consistency checking of compliance rules. In: Business Information Systems. ch.10, vol. 47, Springer, Heidelberg (2010)
6. Giblin, C., et al.: From regulatory policies to event monitoring rules: Towards model-driven compliance automation. Technical report, IBM Zurich (2006)
7. Rodríguez, C., et al.: Analyzing compliance of service-based business processes for root-cause analysis and prediction. In: Proceedings of ESW 2010, Springer, Heidelberg (2010)
8. Schumm, D., et al.: Integrating Compliance into Business Processes: Process Fragments as Reusable Compliance Controls. In: Proc. of the Multikonferenz Wirtschaftsinformatik (MKWI 2010), Universitätsverlag, Göttingen (2010)
9. Daniel, F., et al.: Business compliance governance in service-oriented architectures. In: Proceedings of the IEEE Twenty-Third International Conference on Advanced Information Networking and Applications (AINA 2009), Bradford, UK (May 2009)
10. Governatori, G., et al.: Detecting regulatory compliance for business process models through semantic annotations. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. ch. 2, vol. 17, Springer, Heidelberg (2009)

11. Tran, H., et al.: Modeling Process-Driven SOAs - a View-Based Approach. In: Cardoso, J., van der Aalst, W. (eds.) Information Science Reference (2009)
12. Henry, T.: Product for managing governance, risk, and compliance: Market fluff or relevant stuff? Report of Burton Group (March 2008)
13. Kuester, J., Ryndina, K., Gall, H.: Generation of business process models for object life cycle compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 165–181. Springer, Heidelberg (2007)
14. Ly, L.T., et al.: Integration and verification of semantic constraints in adaptive process management systems. Data Knowl. Eng. 64(1), 3–23 (2008)
15. El Kharbili, M., et al.: Policy-based semantic compliance checking for business process management. In: Proceedings of the Workshops co-located with the MobIS2008 Conference,CEUR Workshop Proceedings, aarbrücken, Germany. CEUR Workshop Proceedings, vol. 420, pp. 178–192 (November 2008) CEUR-WS.org
16. El Kharbili, M., et al.: Towards a framework for semantic business process compliance management (2008)
17. Michelson, B.M.: Event-driven architecture overview. Report of Patricia Seybold Group (2006)
18. Namiri, K., Stojanovic, N.: Pattern-based design and validation of business process compliance. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 59–76. Springer, Heidelberg (2007)
19. Object Management Group (OMG). Business Process Model And Notation (BPMN). Version 1.2, OMG Specification (January 2009)
20. Silveira, P., et al.: On the design of compliance governance dashboards for effective compliance and audit management. In: Proc. of the 3rd Workshop on Non-Functional Properties and SLA Management in SOC, NFPSLAM-SOC 2009 (2009)
21. Iannella, R.: Open Digital Rights Language (ODRL). Version 1.1, (Septmeber 2002)
22. Robinson, W.: A requirements monitoring framework for enterprise systems. Requirements Engineering 11(1), 17–41 (2006)
23. Sadiq, S.W., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
24. Sriraman, B., Radhakrishnan, R.: Event driven architecture augmenting service oriented architectures. Report of Unisys and Sun Microsystems (2005)
25. Holmes, T., et al.: Monitoring and analyzing service-based internet systems through a model-aware service environment. In: Pernici, B. (ed.) Advanced Information Systems Engineering. LNCS, vol. 6051, pp. 98–112. Springer, Heidelberg (2010)
26. Phan, T., et al.: A survey of policy-based management approaches for service oriented systems. In: Proceedings of the 19th Australian Conference on Software Engineering (ASWEC 2008), Washington, DC, USA, pp. 392–401 (2008)
27. Wolter, C., Schaad, A.: Modeling of task-based authorization constraints in BPMN. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 64–79. Springer, Heidelberg (2007)