2010

An Implementation of a Process-Oriented Cross-System Compliance Monitoring Approach in a SAP ERP and BI Environment

Thorben Sandner
Leibniz Universität Hannover, sandner@iwi.uni-hannover.de

Matthias Kehlenbeck
Leibniz Universität Hannover, kehlenbeck@iwi.uni-hannover.de

Michael H. Breitner
Leibniz Universität Hannover, breitner@iwi.uni-hannover.de

AN IMPLEMENTATION OF A PROCESS-ORIENTED CROSS-SYSTEM COMPLIANCE MONITORING APPROACH IN A SAP ERP AND BI ENVIRONMENT

Journal:	*18th European Conference on Information Systems*
Manuscript ID:	ECIS2010-0209.R1
Submission Type:	Research-in-Progress Paper
Keyword:	IT compliance, IT risk management, IS security, Business process management

AN IMPLEMENTATION OF A PROCESS-ORIENTED CROSS-SYSTEM COMPLIANCE MONITORING APPROACH IN A SAP ERP AND BI ENVIRONMENT

Sandner, Thorben, Leibniz Universität Hannover, Institut für Wirtschaftsinformatik, Königsworther Platz 1, 30167 Hannover, Germany, sandner@iwi.uni-hannover.de

Kehlenbeck, Matthias, Leibniz Universität Hannover, Institut für Wirtschaftsinformatik, Königsworther Platz 1, 30167 Hannover, Germany, kehlenbeck@iwi.uni-hannover.de

Breitner, Michael H., Leibniz Universität Hannover, Institut für Wirtschaftsinformatik, Königsworther Platz 1, 30167 Hannover, Germany, breitner@iwi.uni-hannover.de

Abstract

Compliance to regulatory demands has become a crucial matter for organizations. Non-observance may lead to far-reaching consequences, e.g. damage to reputation, decline of credit rating or market value, fraud and fines. The success of compliance management correlates with the frequency of monitoring and reporting and is affected by complex and often time-consuming manual validation tasks. To address this problem, organizations implement corresponding IT solutions. However, the often heterogeneous system landscapes, the different information sources and their integration represent major challenges.

This paper presents an implementation of a novel process-oriented and cross-system compliance monitoring approach. The approach is based on a model which provides for the annotation of business processes with internal controls, critical permissions and roles as well as an architecture which provides for the automatic detection, timely communication and deep analysis of control exceptions. It solely relies on established standards (i.e. XACML, BPMN, COSO and SWRL) and existing technologies. The implementation has been deployed in a productive SAP ERP and BI environment. It automatically converts access control data from the proprietary SAP model and publishes control exceptions to the BI system. The effects and causes of these control exception can be appropriately analyzed using BI queries and reports.

Keywords: IT compliance, IT risk management, IS security, business process management, SAP R/3

1 INTRODUCTION

Triggered by a set of enterprise scandals as for example Enron or WorldCom, the compliance standards considerably raised during the last years. Many additional regulations such as Sarbanes-Oxley Act (SOX) or EuroSOX were released to ensure the reasonable acting of organizations. The implementation of these regulations is often a prerequisite for organizations to continue their work. The resulting investment cost required to introduce and operate corresponding measures is estimated in the U.S. as 32 billion U.S. dollars for the year 2008 (McGreevy et al. 2008).

Compliance management can be defined as the use of frameworks, standards and software to ensure compliance with legal requirements (Kharbili et al. 2008). Compliance may be achieved by embedding control activities into business processes and supporting systems. However, dynamic environments frequently require changes to processes and systems. The maintenance and monitoring of controls is a complex, time-consuming and often manual task (Bace et al. 2006), (Agrawal et al. 2006). As compliance management depends on the frequency of monitoring and reporting, the timely communication of control exceptions is an important success factor (Liebenau et al. 2006). Further factors are central approaches, proactive responses and automated processes (Chatterjee et al. 2008).

The present paper focuses on the use of software to ensure compliance. This software forms the technical infrastructure for the realization and traceability of compliance management. In most Information Systems (IS) publications to this subject, the research focus rather lies on exploratory problem-indentifying instead of developing concrete solutions (Syed et al. 2009). It is also criticized that the solutions to compliance issues are often implemented in isolation and do not adequately address the need for information from different data sources as well as the need for analytic data (Gericke et al. 2009). Although some systems provide strong internal control features, heterogeneous system landscapes render an integrated monitoring and reporting very difficult.

Echoing these criticisms, the present paper describes a prototypical implementation in a productive SAP environment. The prototype enables the automated and central monitoring of controls distributed over multiple heterogeneous systems. Compliance information is integrated into a central repository and forms the basis for the creation of homogeneous analyses and reports. This may prevent the definition of redundant controls and bundle competencies. Noncompliance situations may be avoided by analyzing the impact of changes to processes, permissions and roles on internal control before these changes become productive. The prototype relies on a Service-Oriented Architecture (SOA) and uses a Business Intelligence (BI) system for analysis. It is based on a model which provides for the annotation of business processes with internal controls, critical permissions and roles as well as an architecture which provides for the automatic detection, timely communication and deep analysis of control exceptions. Both are based on existing standards and technologies. Processes are described using the Business Process Modeling Notation (BPMN) (OMG 2009a) in combination with the XML Process Definition Language (XPDL) (WfMC 2009). Access control information is specified using the Extensible Access Control Markup Language (XACML) (OASIS 2009). Internal control is described following the established Internal Control – Integrated Framework (COSO 1992) respectively Enterprise Risk Management – Integrated Framework (COSO) (COSO 2004) and control exceptions are formally defined using the Semantic Web Rule Language (SWRL) (W3C 2004). Furthermore, access control information is automatically transformed from the proprietary SAP model to XACML. Other relevant information (e.g. customizations and parameters) can be transformed to OWL.

In the IS Research, there are two fundamental types of approaches: (1) the behavioral and (2) the design science research (DSR) approach (Hevner et al. 2004). Here, the DSR approach is selected. It focuses on the heuristic search for new and innovative artifacts. Artifacts consist of constructs, models, and methods and are converted to problem-related instances (March et al. 1995). In this paper, a situational adjustment of a generic artifact (to an SAP ERP and BI environment) is made and the usefulness of this developed artifact is evaluated using dynamic and architecture analysis.

The remainder of the paper is structured as follows. Section 2 gives an overview about the related work. In section 3 a model, architecture and implementation of a control monitoring system is presented. The transformation process from the monitored systems to the monitoring system is described in section 4. Section 5 contains an evaluation of the monitoring system. We conclude with a discussion about future work in section 6.

2 RELATED WORK

The increasing process orientation in consideration of business perspectives and security requirements let several works, e.g. Wolter et al. (2007), Wang et al. (2008), Basin et al. (2003) and Jürjens (2002), examine the implications of Unified Modeling Language (UML) (OMG 2009b) models or BPMN models in conjunction with security policies. However, they have different focuses and do not detail on control monitoring. Höhn and Jürjens (2008) deal with the analysis of UML models of business applications and corresponding configuration data in terms of their relevance for security policies and compliance requirements. Thematically closer are Wolter et al. (2007), that describe a mapping between BPMN and XACML meta-models for the automated derivation of authorization constraints, specifically an XSL Transformation (XSLT) (W3C 1999) that converts security constraints into XACML policies. However, the present approach contains a transformation between different access control models, not between a process and an access control model.

Pistoia et al. (2007) and Sadiq et al. (2007) discuss some other aspects of the topic. Pistoia et al. (2007) focus on the static policy validation for a Role Based Access Control (RBAC) model. However, the present approach uses XACML which allows the use of other access control models than RBAC. Additionally, not only a static analysis is possible but also a runtime analysis of policies. Sadiq et al. (2007) concentrate on a language for the representation of control objectives and annotate business processes with corresponding control tags. However, the present approach includes an access control model, uses a standard rule language and resembles the COSO model more closely.

Other approaches that have different intentions but are related to the employed technologies are Ferrini and Bertino (2009) as well as Kolovski et al. (2007). Ferrini and Bertino (2009) extend XACML with a framework that integrates OWL ontologies and XACML policies to support static and dynamic segregation of duties. Like the present approach, they combine XACML with OWL. However their main focus lays on the improvement of RBAC. Kolovski et al. (2007) have developed a description logic based analysis service for XACML policies. They combine XACML and description logic along with the reasoner Pellet (Clark and Parsia 2009) for verifying properties of XACML policies.

Kehlenbeck et al. (2010) describe an approach for the annotation of business processes with controls, permissions and roles based on BPMN, COSO and XACML. Additionally, they propose an architecture for the automated monitoring of controls and the timely communication of thereby detected control exceptions. The present approach adopts their model and architecture, supplements it with a conversion web service which automatically transforms access control data from the proprietary SAP model to XACML, implements this supplemented approach in a productive ERP and BI environment and evaluates this implementation.

3 MODEL, ARCHITECTURE AND IMPLEMENTATION

3.1 Model

Clearly structured business processes help organizations to achieve their goals. Many controls contained in these processes can be supported by IT systems. In particular, authorization and segregation of duties controls can be mapped to systems by means of their access control functions. This abets transparency and traceability. The present approach adopts the model developed by Kehlenbeck et al. (2010). It is illustrated in Figure 1 and concisely described in the following subsections.

3.1.1 Process model

The design and implementation of business processes requires the participation of numerous people with different backgrounds. The process owner often possesses extensive knowledge regarding his processes but frequently needs help with their alignment to regulatory requirements (e.g. SOX and EuroSOX) and the design and implementation of corresponding controls. Moreover, IT specialists have to map a substantial part of these controls to supporting IT systems. To facilitate efficient communication between these participants, the BPMN has been developed. BPMN may be exchange using XPDL, which is formally defined by an XML Schema Definition (XSD) (W3C 2008b).

3.1.2 Access Control Model

A variety of heterogeneous distributed systems exist in organizations. Consequently, there are efforts to centralize the administration and enforcement of access control (e.g. Damianou et al. 2001, W3C 2006 and PERMIS 2003). To tackle this problem, the Organization for the Advancement of Structured Information Standards (OASIS) offers XACML, a platform independent access control standard. It provides for a Policy Decision Point (PDP), which is a processing engine that makes authorization policies interpretable and delivers decisions about acceptance or rejection. A further interesting aspect arises by using the RBAC profile for XACML (OASIS 2005), which makes it possible to map the relationships between roles and permissions as they are typically contained in supporting IT systems. XACML is formally defined by several XSDs. Its comprehensive function range let it appear suitable for the exchange between monitored systems and monitoring systems.

3.1.3 Internal Control Model

In contrast to business process models and access control models, formally defined and standardized internal control models do not exist. To close this gap, the established COSO model has been formally defined by an XSD. Additionally, this XSD provides for the definition of control exceptions by any XML based rule definition language. The present approach uses SWRL as this rule definition language. SWRL combines the Rule Markup Language (RuleML) (RuleML Initiative 2009) and OWL. It is supported by several editors (e.g. Protégé) and reasoners (e.g. Pellet). SWRL rules are used to describe, which combinations of critical permissions and optionally other information (e.g. customizing and parameters) imply which control exceptions. Permissions can be linked to XACML actions and resources. Actions, resources and subjects are parts of XACML targets. The formal definition of control exceptions (e.g. infringed segregation of duties) is a precondition for their automatic detection. The internal control and permission model is referred to as the Extensible Business Risk Description Language (XBRDL).

3.2 Architecture and Implementation

The described model has been prototypically implemented in a SAP ERP and BI environment. The implementation is based on a Service-Oriented Architecture (SOA) (OASIS 2009b). This architecture provides for the loose coupling of components and thereby increases flexibility and facilitates reutilization. It is a variation of the architecture developed by Kehlenbeck et al. (2010), illustrated in Figure 2 and concisely described in the following subsections.

3.2.1 XACML PDP Web Service

The PDP is the core component of an XACML engine. The PDP examines incoming requests, determines applicable policy sets and returns a corresponding decision. It thereby decides whether a person is permitted to perform an action on a resource or not. Policy sets may originate from one or more systems of the IT landscape and may refer to single or multiple system levels, e.g. operation system level, database level and / or application level. Systems may either natively use XACML or a

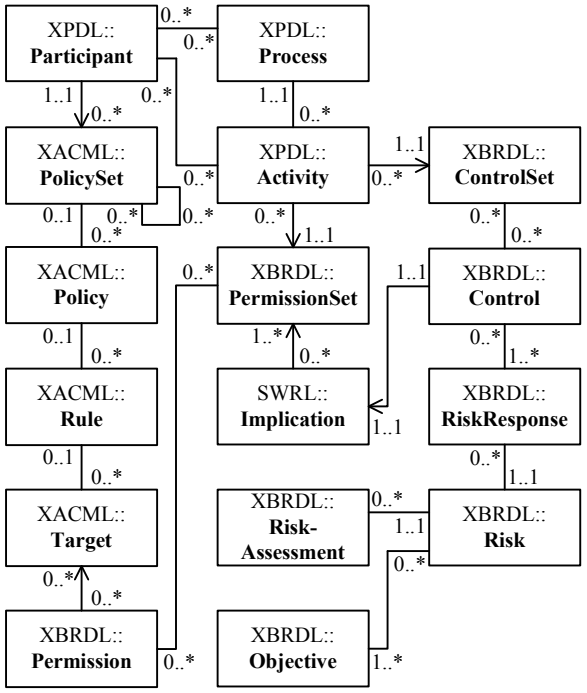

Figure 1. Used model as a UML diagram.

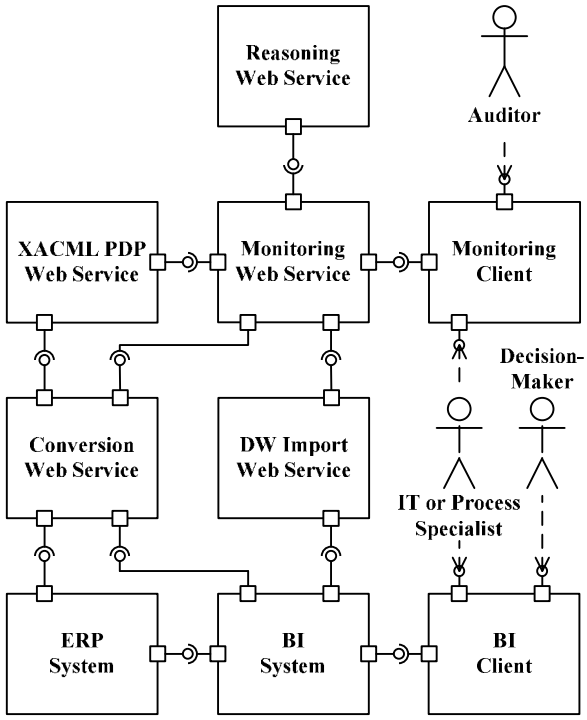

Figure 2. Used architecture as a UML diagram.

The diagrams are adopted from Kehlenbeck et al. (2010) except for the conversion web service.

proprietary access control model. The latter case requires a transformation from the proprietary model to XACML. Several implementations of XACML are available. They differ in licensing terms, technical maturity and performance (Scaglioso et al. 2008). The XACML PDP web service encapsulates the implementation by SUN (SUNXACML) (SUN 2004), as it offers a high level of conformity (Li et al. 2008), a comprehensive documentation and an open source license.

3.2.2 Reasoning Web Service

The reasoning web service accepts incoming OWL and SWRL assertions and builds up a corresponding knowledge base. Based on this knowledge base, it processes incoming SPARQL (W3C 2008a) queries and returns their result. In particular, it thereby evaluates which persons infringe which controls. The reasoning web service employs the Jena API (Jena 2009) in conjunction with Pellet.

3.2.3 Conversion Web Service

The conversion web service extracts access control and other information (i.e. customizations and parameters) from the SAP ERP and BI systems and transforms these to XACML permission policy sets, role policy sets and role assignment policies as well as OWL ontologies. SAP ERP and BI use the same proprietary access control model. The transformation is detailed in section 4.

3.2.4 Monitoring Web Service and Client

The monitoring web service accepts incoming XPDL business processes, XBRDL control and permission sets, XACML role assignment policy sets as well as OWL ontologies containing other information. As the model solely consists of formally defined sub models, it was easily possible to generate a model implementation based on the corresponding XSDs using Model Driven Architecture (MDA) (Ball and Craig 2008) tools. The actions and resources contained in the XBRDL permission sets are combined with the subjects contained in the XACML role assignment policy sets and passed to the XACML PDP web service. The latter evaluates these requests and returns corresponding

decisions. These decisions are converted to OWL and passed to the reasoning web service along with the received OWL ontologies and the SWRL rules contained in the XBRDL control sets. Based on this, the reasoning web service infers and returns existing control exceptions. Finally, these control exceptions are published together with the original XPDL and XBRDL information to the data warehousing (DW) web service. The monitoring web service may be configured and invoked using the monitoring client.

3.2.5 Data Warehousing Import Web Service

The information needs and their levels of granularity differ significantly for the involved participants. Decision-makers employ high level reports to survey the effects of control exceptions whereas IT and process specialists exploit drill-down functionalities to identify their specific causes. These requirements are met by the delivery of data to a corresponding warehouse and the subsequent use of business intelligence tools. The data warehousing import web services is used to uncouple this delivery from the monitoring web service to a particular business intelligence system.

3.2.6 Enterprise Resource Planning System

In order to test the prototype implementation in a meaningful and realistic environment, a commonly used business application has been selected, an Enterprise Resource Planning system. From the market of ERP vendors, a leading system, SAP ERP has been chosen. SAP ERP offers a very sophisticated access control model. Its transformation to XACML policies is therefore considered interesting.

3.2.7 Business Intelligence System and Client

Business Intelligence systems enable the performance of deep analyses and the production of meaningful reports. The provision of internal control information in a BI environment is considered suitable, as this allows its presentation in a coherent way with other characteristics and facts. Another inherent benefit of BI is the ability to analyze internal control under temporal aspects. The SAP BI system receives internal control data from the DW import web service and other data from the SAP ERP system. Decision-makers, IT and process specialists use the SAP Business Explorer as a client. The interaction of the individual components from the invocation of the monitoring web service to the import into SAP BI is illustrated in Figure 3.

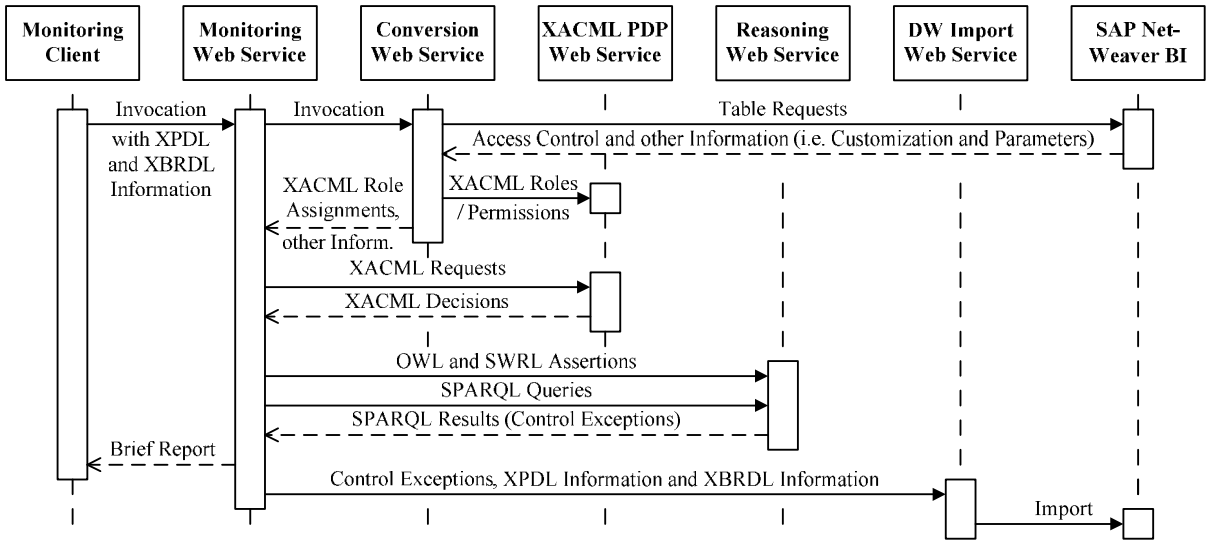

Figure 3. Interaction of the individual prototype components as a UML diagram. SAP BI and SAP ERP share the same access control model. The illustration therefore omits SAP ERP.

4 TRANSFORMATION

The developed conversion web service is able to extract access control data from SAP ERP and BI systems using (1) a SOAP connection to a corresponding web service, (2) a Java Database Connectivity (JDBC) connection to a database and / or (3) a folder of ALV files (a SAP internal XML format) which have been saved with the SAP GUI. At first, data is converted to an internal XML table format. ALV files are transformed to it using XSLT. The XML table format has been formally defined by an XSD. As the same XSD has been embedded into the Web Service Description Language (WSDL) file of the web service, extracted data can be directly marshalled to the XML table format using the Java Architecture for XML Binding (JAXB). The XSD has also been used to create a model implementation by means of the Eclipse Modelling Framework (EMF). This implementation is used to write data which has been extracted using JDBC to the XML table format and to read from data in the XML table format created by any of the three ways.

After the data has been converted to the internal XML table format, it is used to create (1) XACML role assignment policies, (2) XACML role and permission policy sets and (3) OWL ontologies. The role assignment policies and ontologies are sent to the monitoring, the role and permission policy sets to the XACML PDP web service. Figure 4 illustrates the activities performed by the conversion web service and its relation to other components, in particular the monitoring web service. The following subsections describe the SAP access control model and its transformation to XACML and OWL.

4.1 SAP Access Control Model

SAP access control is stored in a relational model and distinguishes between profiles and roles. Both profiles and roles are assigned to users and contain authorizations. Authorizations link permission objects with field values. However, profiles are used for access control enforcement, while roles are used for access control administration. When an administrator maintains a role, the SAP system automatically updates corresponding profiles. When these roles are assigned to users, these profiles are automatically assigned, too. In order to reduce overhead, only profiles are transformed to XACML policies. However, roles can be transformed to OWL ontologies. Figure 5 illustrates this model.

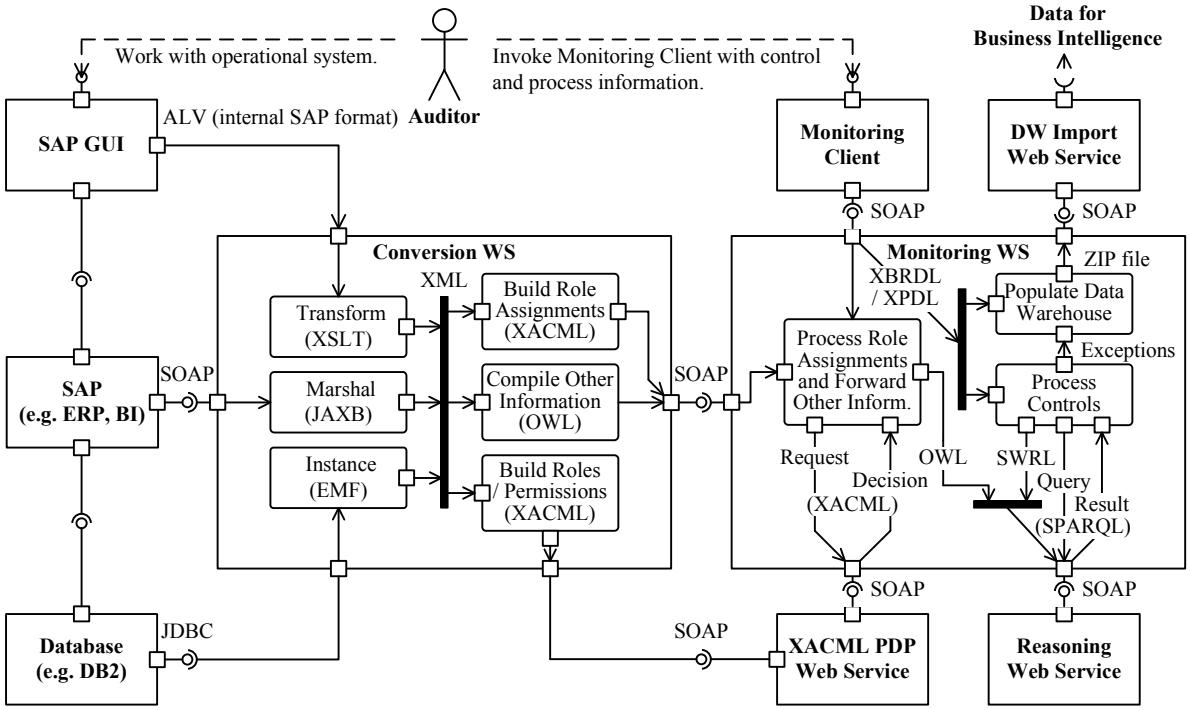

Figure 4. Conversion and monitoring web service as a UML diagram.

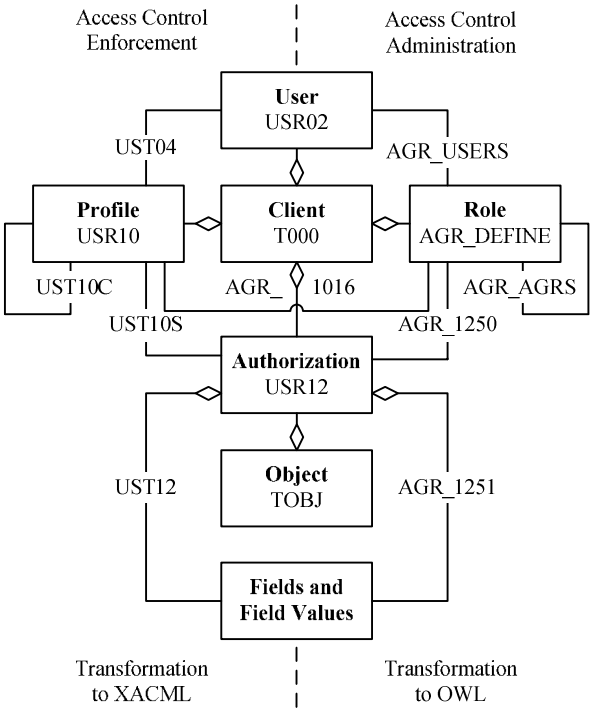

Figure 5. SAP access control model as a UML diagram.

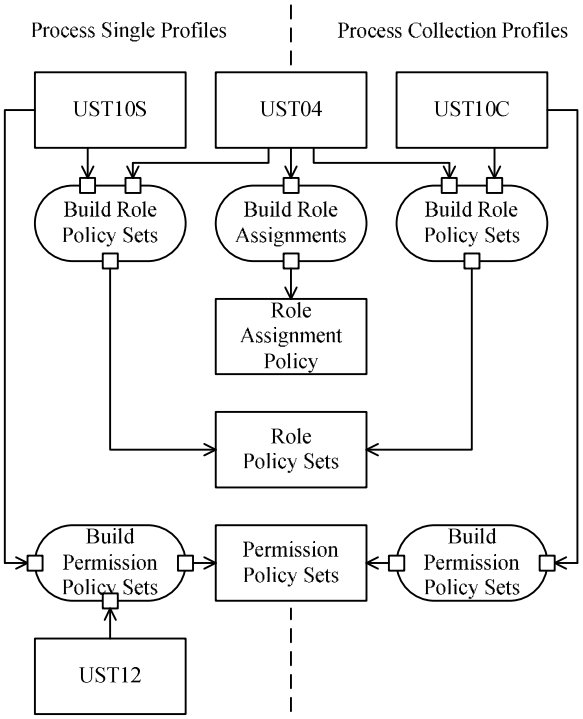

Figure. 6. Transformation from SAP to XACML as a UML diagram.

4.2 Transformation to XACML and OWL

The required information for the transformation of SAP profiles to XACML role assignment policies as well as XACML role and permission policy sets are contained in the tables UST04 (users and profiles), UST10C (collection profiles), UST10S (single profiles and authorizations) and USR12 (authorizations and field values). The transformation from SAP to XACML is illustrated in Figure 6 and consists of the following steps:

1. Table UST04 is converted to an XACML role assignment policy. SAP user names are mapped to XAMCL subjects and SAP profile names to XACML role policy set ids.
2. For each single profile in UST10S, a permission policy set is created. Mappings are as follows:
   a. SAP profile names are mapped to XACML policy and policy set ids,
   b. SAP authorization and object names are concatenated and mapped to XACML rule ids,
   c. SAP objects names are mapped to XACML resources and
   d. SAP field names and their values are concatenated and mapped to XACML actions.
3. For each collection profile in UST10C, an XACML permission policy set is created. The XACML permission policy sets corresponding to the contained profiles are included by reference.
4. For each profile in UST10S and UST04 or UST10C and UST04, an XACML role policy set is created. The corresponding XACML permission policy set is included by reference.

MANDT	PROFN	AKTPS	OBJCT	AUTH
700	T-P1281126	A	S_TCODE	T-P128112601

Table 1. A single profile in table UST10S.

MANDT	OBJCT	AUTH	AKTPS	FIELD	VON	BIS
700	S_TCODE	T-P128112601	A	TCD	F110	

Table 2. A field and field value in table UST12.

SAP field values support ranges and may contain wildcards. However, the numerous available XACML functions have made the mapping easy. Table 1 and Table 2 show a single profile and a field value in SAP, a corresponding XACML fragment is:

```
...
<Policy PolicyId="PP_T-P1281126"
 RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
 <Target/>
 <Rule Effect="Permit" RuleId="S_TCODE_T-P128112601">
  <Target>
   <Resources>
    <Resource>
     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">S_TCODE</AttributeValue>
      ...
     </ResourceMatch>
    </Resource>
   </Resources>
   <Actions>
    <Action>
     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">TCD_F110</AttributeValue>
      ...
     </ActionMatch>
    </Action>
    ...
   </Actions>
  </Target>
 </Rule>
 ...
</Policy>
...
```

The SAP access control enforcement is also influenced by certain customizations and parameters. E.g. transaction codes listed in the system parameter “auth/tcodes_not_checked” are not checked at all. A transaction code is an object (S_TCODE). These issues have to be addressed in the SWRL rules. The conversion web service uses OWL to add corresponding information to the knowledge base of the reasoning web service. OWL may also be used to add information regarding SAP access control administration.

5 EVALUATION

The evaluation of the developed artifact depends on the requirements of the business environment and its corresponding technical infrastructure (Hevner et al. 2004). For the present artifact, appropriate steps are to perform an architecture analysis, which examines how well it fits into the technical infrastructure and a dynamic analysis, which addresses qualities such as performance (Hevner et al. 2004). Another key aspect is to verify the feasibility of the chosen approach.

The business environment used for the evaluation is a SAP ERP multitenant system with more than 1,200 users. It is productive for about nine years. To simplify the technical evaluation of the controls with reference to a concrete organization structure, the evaluation focuses on the biggest client with 586 users. This client uses the SAP modules for finance, human-resources, controlling and materials management and has some central and several decentralized structures, which make a differentiated authorization concept with multiple segregations of duties necessary. To map these requirements to the system, the client maintains 1,190 roles. In order to enforce these roles, the SAP system automatically updates 1,197 profiles in the background.

The integration of the artifact into the technical infrastructure can be divided into three different parts: (1) its incorporation into the existing system landscape, (2) its upstream connection to the monitored systems and (3) its downstream connection to the BI system. The architecture of the artifact made it easy to deploy it to an existing application server and establish connections to the SAP ERP and BI systems. The flexible web services facilitate extensibility and maintainability. Upstream connections to the monitored system were established using SOAP but would have also been possible using JDBC.

Characteristic	Number	Kilobyte
Role Policy Set to User Assignments (RPAs)	1,200	3,342
Role Policy Sets (RPS)	1,200[1]	1,165
Collection Permission Policy Sets (CPPS)	140	159
Single Permission Policy Sets (SPPS)	2,780	128,375

[1] 3 RPS for CPPS and 1197 RPS for SPPS

Characteristic	Average	Minimum	Maximum	Standard Deviance
Number of Users per RPA	4.30	1	586	23.41
Number of SPPS per CPPS	5.71	1	81	7.59
Number of Permissions per SPPS	12.75	1	170	23.46

Table 3 and 4. Information regarding the XACML created by the conversion web service.

Downstream connections were established using SOAP as well. Additional systems may be easily connected without significant adjustments to model or architecture.

Table 3 and 4 contain quantitative information regarding the role assignment policies, role policy sets and permissions policy sets created by the conversion web service. Particularly interesting is the number of users per role policy set assignment. There exists a basis profile which is assigned to all users. Furthermore, each profile is assigned to at least one user. Finally, the discrepancy between the average and the standard deviation may be explained by the very sophisticated access control concept. The majority of the users only possess the basis and a few other profiles for their specific area of work. However, a few key users with several profiles and three technical users with almost all profiles exist in the system. The latter are the only users that possess collection permission policy sets.

The export of access control data from the productive ERP system to equivalent XACML policies resulted in a large number of files. This large number entails two negative implications. First, the large number of files impairs the performance. This has already been described in a similar extent by Liu et al. (2008) as well as Turkmen and Crispo (2008). Most other available PDPs have a scaling problem, too. Second, the large number of files let a manual administration appear extremely time-consuming. However, both implications are rather unimportant for the present approach.

Business processes and controls have been defined in cooperation with the financials process owner. The generated policy sets and the control exceptions detected by the monitoring system have been verified against the source data from the monitored system by random checks.

As the approach is based on established standards and existing technologies, standard software such as TIBCO Business Studio (TIBCO 2009) may be used. This renders the development of individual software unnecessary and thereby increases cost effectiveness.

6 CONCLUSION

Although compliance standards considerably raised during the last years, most corresponding IS publications focus on exploratory problem-identifying instead of developing concrete solutions. Even the few developed solutions are often implemented in isolation and do not adequately address the need for information from different data sources as well as the need for analytic data. In order to meet the need for suitable solutions, a prototypical implementation of an innovative compliance monitoring approach in a productive SAP environment is presented. The prototype enables the automated, central and proactive monitoring of controls distributed over multiple heterogeneous systems. It is based on a model which provides for the annotation of BPMN business processes with COSO inspired internal controls, critical permissions and roles as well as an Service-Oriented Architecture which provides for the automatic detection, timely communication and deep analysis of control exceptions. Both are based on existing standards and technologies. Control exceptions are formally defined using the SWRL rule language. Permissions and roles are defined using the XACML access control modeling language. They may originate from one or more systems of the IT landscape and may refer to single or multiple system levels, e.g. operation system level, database level and / or application level.

As the implementation has been deployed in a SAP ERP and BI environment, an automatic transformation from their proprietary access control model to XACML have been developed. Other relevant information (e.g. customizations and parameters) may be transformed to OWL ontologies.

The deployed implementation has been evaluated with access control data from an organization with 586 users and 1,190 roles. Business processes and controls have been defined in cooperation with the financials process owner. The results of the transformation and the monitoring processes have been verified against the source data from the monitored system by random checks.

Future research will be dedicated to the evaluation of the implementation in a field study. Furthermore, it will be extended with additional transformations from other proprietary access control models to XACML.

References

Agrawal, R., Johnson, C., Kiernan, J. and Leymann F. (2006): Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In Proceedings of the 22nd International Conference on Data Engineering, pp. 92-102, IEEE, Washington.

Bace, J. and Rozwell, C. (2006): Understanding the Components of Compliance, Gartner Report: G00137902.

Ball, M. and Craig, B. (2008): Object Oriented jDREW, http://www.jdrew.org/oojdrew/

Basin, D., Doser, J. and Lodderstedt, T. (2003): Model Driven Security for Process-Oriented Systems. In Proceedings of the eighth ACM Symposium on Access Control Models and Technologies, pp. 100–109, ACM, NY.

Chatterjee, A. and Milam, D. (2008): Gaining Competitive Advantage from Compliance and Risk Management. In Pantaleo, D. and Pal, N. (Eds.): From Strategy to Execution, pp. 167-183, Springer, Berlin.

Clark and Parsia (2009): Pellet: The Open Source OWL Reasoner, http://clarkparsia.com/pellet

COSO (1992): Committee of Sponsoring Organizations of the Treadway Commission (COSO): Internal Control – Integrated Framework, http://www.coso.org/guidance.htm

COSO (2004): Committee of Sponsoring Organizations of the Treadway Commission (COSO): Enterprise Risk Management - Integrated Framework, Executive Summary, http://www.coso.org/Publications/ERM/COSO_ERM_ExecutiveSummary.pdf

Damianou, N., Dulay, N., Lupu, E. and Sloman, M. (2001): The Ponder Policy Specification Language. In Proceedings of the International Workshop on Policies for Distributed Systems and Networks, pp. 18-38, Springer, London.

Ferrini, R. and Bertino E. (2009): Supporting RBAC with XACML+OWL. In Proceedings of the 14th ACM symposium on Access control models and technologies, pp. 145-154, ACM, NY.

Gericke, A., Fill, H.-G., Karagiannis, D. and Winter, R. (2009): Situational Method Engineering for Governance, Risk and Compliance Information Systems. In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, ACM, NY.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004): Design Science in Information Systems Research. MIS Quarterly. 28, 1, 75-105.

Höhn, S. and Jürjens, J. (2008): Rubacon: automated support for model-based compliance engineering. In Proceedings of the 30th international conference on Software engineering, pp. 875-878, ACM, NY.

Jena (2009): Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/

Jürjens, J. (2002): UMLsec: Extending UML for Secure Systems Development. In Proceedings of the 5th International Conference on The Unified Modeling Language, pp. 412–425, Springer, London.

Kehlenbeck, M., Sandner, T. and Breitner, M. H. (2010): Managing Internal Control in Changing Organizations through Business Process Intelligence – A Service Oriented Architecture for the XACML based Monitoring of Supporting Systems. In Proceedings of the 43th Hawaii International Conference on System Sciences (HICSS-43), 10 pages, CD-ROM, IEEE, Washington.

Kolovski, V., Hendler, J. and Parsia, B. (2007): Analyzing web access control policies. In Proceedings of the 16th international conference on World Wide Web, pp. 677-686, ACM, NY.
Kharbili, M. E., Stein, S., Markovic, I. and Pulvermüller, E. (2008): Towards a Framework for Semantic Business Process Compliance Management. In Proceedings of GRCIS 2008.
Li, N., Hwang, J. and Xie, T. (2008): Multiple-implementation testing for XACML implementations, In Proceedings of the 2008 workshop on Testing, analysis, and verification of web services and applications, pp. 27-33, ACM, NY.
Liebenau, J. and Kärrberg, P. (2006): International Perspectives on Information Security Practices. London School of Economics and Political Science, McAfee.
March, S. T. and Smith, G. F. (1995): Design and Natural Science Research on Information Technology. Decision Support Systems. 15, 4, 251-266.
McGreevy, M. (2008): AMR Research Finds Spending on Governance, Risk Management, and Compliance Will Exceed $32B in 2008. http://www.amrresearch.com/Content/View.aspx?pmillid=21310
OASIS (2005): Core and hierarchical role based access control (RBAC) profile of XACML v2.0, http://docs.oasisopen.org/xacml/2.0/access_control-xacml-2.0-rbacprofile1-spec-os.pdf
OASIS (2009a): eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/xacml/
OASIS (2009b): SOA Reference Model, http://www.oasis-open.org/committees/soa-rm/
OMG (2009a): Business Process Modeling Notation (BPMN), http://www.bpmn.org/
OMG (2009b): Unified Modeling Language (UML), http://www.uml.org/
PERMIS (2003): PrivilEge and Role Management Infrastructure Standards Validation (PERMIS), http://www.permis.org/
Pistoia, M., Fink, S.J., Flynn, R.J. and Yahav, E. (2007): When Role Models Have Flaws. In Proceedings of the 29th international conference on Software Engineering, pp. 478-488, IEEE, Washington.
RuleML Initiative (2009): Rule Markup Language (RuleML), http://ruleml.org
Sadiq, S., Governatori, G. and Namiri, K. (2007): Modeling Control Objectives for Business Process Compliance. Business Process Management. pp. 149-164, Springer, Berlin.
Scaglioso, P.G., Basile, C. and Lioy, A. (2008): Modern Standard based Access Control in Network Services: XACML in action. IJCSNS International Journal of Computer Science and Network Security Vol. 8 No. 12, pp. 296-305.
SUN (2004): Sun's XACML implementation, Version 1.2, http://sourceforge.net/projects/sunxacml.
Syed, A., Syed, N. H., Indulska, M. and Sadiq, S. (2009): A STUDY OF COMPLIANCE MANAGEMENT IN INFORMATION SYSTEMS RESEARCH. In Proceedings of the 17th European Conference on Information Systems.
TIBCO (2009): TIBCO Business Studio, http://developer.tibco.com/business_studio/default.jsp
Turkmen, F. and Crispo, B. (2008): Performance evaluation of XACML PDP implementations. In Proceedings of the 2008 ACM workshop on Secure web services. pp. 37-44, ACM, NY.
TIBCO (2009): TIBCO Business Studio, http://developer.tibco.com/business_studio/default.jsp
W3C (1999): XSL Transformations, http://www.w3.org/TR/xslt
W3C (2004): SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/
W3C (2006): Web Services Policy Framework, www.w3.org/Submission/WS-Policy/
W3C (2008a): SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/
W3C (2008b): XML Schema, http://www.w3.org/XML/Schema
Wang, X., Zhang, Y., Shi, H. and Yang J. (2008): BPEL4RBAC: An Authorisation Specification for WS-BPEL. In Proceedings of the 9th international conference on Web Information Systems Engineering, pp. 381-395, Springer, Berlin.
WfMC (2009): XML Process Definition Language (XPDL), http://www.wfmc.org/xpdl.html
Wolter, C., Schaad, A. and Meinel, C. (2007): Deriving XACML Policies from Business Process Models. WISE 2007 Workshops, LNCS 4832, 142–153.