Managing Internal Control in Changing Organizations through Business Process Intelligence – A Service Oriented Architecture for the XACML based Monitoring of Supporting Systems

Matthias Kehlenbeck, Thorben Sandner, Michael H. Breitner
Leibniz Universität Hannover
{kehlenbeck,sandner,breitner}@iwi.uni-hannover.de

Abstract

Organizations respond to opportunities and risks by strategic decisions. Strategic decisions ensure the sustainable existence of organizations, but require continuous organizational change. Organizational change includes the redesign of business processes. Processes are subject to internal and external requirements. Requirements include the alignment to strategic goals, the effective and efficient use of resources and the compliance with applicable laws and regulations. Their achievement is assured by embedding internal controls into processes. Many controls can be incorporated into supporting systems, as their access control functions allow the modeling of authorization and segregation of duties.

A model for the annotation of processes with controls, permissions and roles based on BPMN, COSO and XACML is presented. Additionally, a Service Oriented Architecture for the automated monitoring of controls and the timely communication of thereby detected control exceptions is proposed. The benefits of the approach are demonstrated in a prototype implementation and a corresponding case study.

1. Introduction

In order to achieve their strategic goals, to make effective and efficient use of their resources and to secure their reliability of financial reporting as well as their compliance with applicable laws and regulations, organizations perform risk management. Risk management is an ongoing process at every level of an organization designed to identify, assess and respond to potential risks for the entity [1]. Risk responses are incorporated into business processes by means of control activities. The management of internal control is therefore an integral part of risk management [2]. This management includes the design and implementation of monitoring procedures which ensure the effective operation of internal control over time as well as the identification and communication of internal control exceptions. Monitoring procedures include – amongst others – the periodic evaluation and testing of controls, the use of continuous monitoring software as well as the analysis of appropriate reports [3].

Provided that an organization utilizes IT systems to support its business processes, many control activities can be incorporated into these systems. In particular, their access control functions typically allow for the modeling of authorization and segregation of duties controls by means of permissions and roles. However, continuous organizational change (and increasing process orientation) entails frequent adjustments to business processes and thereby requires corresponding changes to systems and controls. Moreover, organizations often possess heterogeneous system landscapes and are thereby forced to model these controls in distinct repositories and using different access control languages. Therefore, the monitoring of these controls is complex and often time-consuming and compliance validation as a whole is still mainly a manual task [4], [5]. Thus, there is a need for capable automated detection and monitoring tools.

Corresponding approaches can be roughly distinguished by their employment phase: “after-the-fact” or “before-the-fact” [6], [7]. The after-the-fact phase is the classic application area of (i) manual audits (by consultants) and (ii) automated detection (with application support). A major drawback of after-the-fact approaches is that they entail adjustment costs. However, the before-the-fact phase contains (iii) compliance aware design and (iv) post design verification approaches which proactively try to avoid non compliance situations and thereby strive for the reduction of these costs. Due to the heterogeneous system landscapes, the implementation of these approaches is estimated as “extremely difficult” [6] though. As the frequency of monitoring and reporting correlates with the success of compliance management [8],
it is important that control exceptions are timely communicated to the right decision-makers. This timeliness provides the decision-makers with the necessary latitude for corresponding measures. An absence of these measures may lead to far-reaching consequences, e.g. damage to the organization's reputation, decline of the organization's credit rating or market value, fraud and fines. Consequently, the achievement of the organization's objectives is at risk.

This paper addresses the aforementioned issues by means of Design Science Research [9] and presents a synthesis between (ii) and (iv), as it enables the automated detection of control exceptions both after-the-fact and before-the-fact. In order to reduce complexity and time required, a model for the annotation of business processes with internal controls, critical permissions and roles based on existing standards is proposed. Additionally, an architecture for automated monitoring of authorization and segregation of duties controls as well as the timely communication of thereby detected control exceptions based on existing technologies is presented. Business processes are described using the Business Process Modeling Notation (BPMN) [10] in conjunction with the XML Process Definition Language (XPDL) [11], access control is described using the Extensible Access Control Markup Language (XACML) [12], internal control is described following the established Internal Control – Integrated Framework [13] respectively Enterprise Risk Management – Integrated Framework [2] (COSO) and control exceptions are formally defined using an Extensible Markup Language (XML) [14] based rule language, like the Rule Markup Language (RuleML) [15].

Decision-makers require meaningful information concerning the implications of control exceptions. In order to increase their acceptance, internal control can be presented in a coherent way with other characteristics and facts using a business intelligence (BI) system, thereby enabling a comprehensive view. Additionally, this enables IT specialists and process specialists to exploit drill-down functionalities for the location of the corresponding causes. To demonstrate the merits of this approach, we present a prototype implementation which enables the automated monitoring of controls and the timely communication of thereby detected control exceptions. It is realized by an orchestration of task-specific web services and employed in a SAP Enterprise Resource Planning (ERP) [16] and BI [17] environment. A practical application of the prototype is shown in a case study which refers to a typical financial business process which must consider several segregations of duties.

Section 2 presents an overview of related work. Section 3 describes the proposed model and section 4 the proposed architecture. The prototype implementation is presented in section 5 and a corresponding case-study is provided in section 6. Finally, section 7 concludes this paper.

2. Related Work

The IT infrastructure of today’s organizations consists predominantly of heterogeneous distributed systems. To stay abreast of this development, there have been several attempts to centralize the definition and control of access and authorization (e.g. [18], [19] and [20]). The step towards a standardized and platform independent approach was affected by the Organization for the Advancement of Structured Information Standards (OASIS) with XACML. Amongst others, this standard provides for a processing engine which makes authorization policies interpretable and delivers decisions about acceptance or rejection: the so-called "Policy Decision Point" (PDP).

Alam et al. [21] use XACML in their approach to make the provisioning of security policies among different domains easier. They present SECTET-PL, a specification language for permissions in the context of UML models which transforms access and authorization information. But being part of the SECTET framework for model driven security for B2B-workflows, their work put a focus on specifying permissions for web services.

Pistoia et al. [22] develop a formal model for Role Based Access Control (RBAC) policy validation and a static analysis model for RBAC systems which is capable to analyze static policy models. Through the use of XACML, the present approach allows the use of other access control models than RBAC. Additionally, it enables not only a static analysis but also a runtime analysis of policies (as described in subsection 5.3).

A method for integrating risks in business processes is presented by zur Muehlen and Rosemann [23]. The authors developed a taxonomy of process related risks and capture the risk-related information with an extended Event-driven Process Chain (EPC) Notation. Furthermore, Sadiq et al. [6] developed a language for the representation of control objectives and propose to annotate business processes with corresponding control tags. However, the present approach includes an access control model, uses an internal control model, which resembles the COSO model more closely and prefers a standard rule language.

There has been a couple of work on developing approaches or tools for analyzing BPMN or Unified Modeling Language (UML) [24] models with regard to security requirements, including [25], [26], [27] and [28]. However, no tools for the verification of role or user permissions against security policies are proposed.
Höhn and Jürjens presented Rubacon [29], an implementation to support model-based development and evaluation of software configurations to indemnify compliance with security policies. In particular, they analyze UML models of business applications and corresponding configuration data in terms of their relevance for security policies and compliance requirements. Limitations are the use of proprietary XML formats for access control and rule data and a tightly coupled architecture.

3. Proposed Model

Organizations structure their activities in business processes. Business processes embed controls and are partially supported by IT systems. Authorization and segregation of duties controls can be incorporated into these IT systems using their access control functions. The proposed model therefore consists of a business process sub model, an access control sub model and an internal control sub model. Figure 1 contains an overview of the proposed model.

3.1. Business Process Model

The analysis and the optimization of their business processes are essential abilities for organizations in competition. For this reason, process owners often possess extensive knowledge regarding their processes. However, in terms of the alignment of processes with regulatory requirements (e.g. Sarbanes-Oxley Act (SOX), Euro-SOX, Basel II) and the design and implementation of corresponding controls, they often require external assistance. Moreover, process owners require the assistance of IT specialists to adjust their supporting IT systems. In summary, the design and implementation of processes requires the participation of numerous people with different backgrounds.

The BPMN has been developed to facilitate efficient communication between participants with different backgrounds. Additionally, it provides a mapping to the Web Services Business Process Execution Language (WSBPEL) [30]. UML activity diagrams are considered less suitable, as they were developed for a different problem domain – software engineering – and do not provide a mapping to web services.

BPMN defines a diagram notation, but not an exchange format. However, XPDL can be used to exchange BPMN diagrams. As an XML Schema Definition (XSD) [31] is available for XPDL, code generators may be used to create the model implementation.

3.2. Access Control Model

XACML can be used to centralize the definition and control of access and authorization in organizations. Whenever an authorization request is made, the PDP delivers one of four possible decisions (permit, deny, not applicable or indeterminate). An important advantage arises when using the RBAC profile for XACML [32]. Without any adjustments to XACML, this profile enables to model the relationship between roles and permissions as they are typically found in IT systems. The XACML core concepts and relations used within the scope of this paper are specified in [12] but can be briefly described as follows:

- A rule refers to a target (i.e. actions, resources and subjects) and evaluates a condition (an expression) to an effect (permit or deny)
- A policy contains multiple rules and combines their effects to its decision.
- A policy set contains multiple policies and combines their decisions to its own. It may also include policies from other policy sets.

The features supported by the standardized XACML render the development of a proprietary format for the exchange between monitored systems and monitoring systems obsolete. There is also an XML Schema Definition (XSD) available for XACML.

3.3. Internal Control Model

While formally defined and standardized models for business processes and access control exist, corresponding models for internal control do – for the best of our knowledge – not. For this reason, the core concepts and relations of the established COSO model required within the scope of this paper were formally defined by an XSD. They are specified in [2] but can be concisely described as follows:

- Organizations set and pursue objectives. Their achievement can be endangered by risks.
- The organizations risk management identifies risks, prepares risk assessments and develops risk responses.
Risk responses are incorporated into business processes by means of control activities. To allow for the automatic detection of control exceptions, this paper additionally provides for the enhancement of control activities by formal definitions of control exceptions using a rule definition language. This internal control model does not impose any restrictions with respect to the concrete rule language except that it should support an XML representation, e.g., like RuleML. The rule language is used to describe which combinations of critical permission sets (and optionally other entities whose inclusion is beyond the scope of this paper) imply a control exception. Permissions are linked to XACML targets, in particular to actions (e.g., register) and resources (e.g., documents or transactions), by means of extended attributes. The use of a rule language renders the monitoring of segregation of duties controls easy and therefore countervails a weakness of XACML.

In the following, the developed formal definition of the internal control model is referred to as the Extensible Business Risk Description Language (XBRDL). It is presented as one possible internal control model. However, it bases upon the established COSO model and facilitates the formal definition of control exceptions.

3.4. Integrated Model

The integration of the defined internal control model (XBRDL) and the relevant parts of the adopted models (XACML, XPDL and e.g., RuleML) for the creation of the proposed model is illustrated in Figure 2. XPDL processes contain activities (e.g., post document) and participants (e.g., roles). These may possess extended attributes just like XBRDL permissions which are used to link XPDL participants to XACML (role) policy sets and XPDL activities to XBRDL control sets and permission sets. This approach enables the use of existing business process modeling tools, e.g., TIBCO Business Studio [33]. The proposed model is presented as one possible solution for the problem domain. However, it excels at the seamless integration and efficient utilization of existing and prevalent models and thereby enables the use of existing components and tools.

4. Proposed Architecture

In order to minimize possible dependencies between components and to maximize their exchangeability and reusability, a Service Oriented Architecture (SOA) [34] is proposed for implementations of the proposed model. Its individual components as well as their interfaces are illustrated in Figure 3 and are concisely described in the following subsections.
4.1. XACML PDP Component

The XACML PDP component evaluates which persons possess which permissions. It accepts incoming XACML requests, processes a repository of XACML (role and permission) policy sets and returns thereby evaluated decisions. Its repository may consist of policy sets concerning a single system or multiple systems and may refer to single or multiple system levels, e.g. operation system level, database level and/or application level. These policy sets may originate from the transformation of data from systems using proprietary access control models or represent data from systems with native XACML support. In the latter case, the productive XACML PDP may be used.

4.2. Reasoning Component

The reasoning component evaluates which persons infringe which controls. It accepts incoming XML encoded assertions and queries. With respect to queries, it returns the inferred results. The reasoner may be based on any suitable kind of logic, e.g. predicate logic or deontic logic. It may natively use a human readable logic programming language, e.g. Prolog [35], and translate between XML and this language, e.g. using Extensible Stylesheet Language Transformations (XSLT) [36], or natively use an XML based logic programming language. A distinct advantage which arises from the choice of a standardized language is the possibility to use existing tools.

4.3 Monitoring Component

The monitoring component detects control exceptions and publishes information to the data warehousing (DW) component. It accepts incoming XPDL business processes, XBRDL control and permission sets as well as XACML role (to user) assignment policy sets. The actions and resources linked to the XBRDL permission sets are combined with the subjects referred in the XACML role assignment policy sets and passed to the XACML PDP component. The latter evaluates these requests and returns corresponding decisions. These decisions as well as the definitions of control exceptions linked to the XBRDL control sets are passed to the reasoning component. Based on the corresponding assertions, the reasoning component infers and returns existing control exceptions. Finally, these control exceptions are published together with the original XPDL and XBRDL information to the DW component. The monitoring component may be configured by auditors.

4.4 Data Warehousing Component

The DW component is used for deep analyses and meaningful reports. It accepts data from the monitoring component and optionally other sources and provides this data in a consistent multidimensional model to analysis and reporting tools. Decision-makers use high level reports and encounter control exceptions with corresponding measures, while IT specialists and process specialists exploit available drill-down functionalities in order to identify their cause.

5. Prototype Implementation

In order to increase the degree of confirmation with respect to the feasibility and suitability of the proposed model and architecture, a prototype implementation in a SAP environment has been developed. Figure 4 contains an overview of the implemented prototype and the following subsections detail on its individual components.

5.1. XACML PDP Web Service

There are different XACML PDP implementations, each with different technical implementation details,
conformity and performance levels [37]. The prototype employs the implementation from SUN (SUN-XACML) [38], because it offers a high level of conformity and its performance shortcomings [39] are of no significance to the following case study. Furthermore, it’s comprehensive documentation and open source license has rendered the development of an encapsulating web service easy.

5.2. Reasoning Web Service

Several industry standards for the definition of rules exist. The Rule Markup Initiative [40] develops RuleML to increase the interoperability between these standards and thereby the corresponding rule engines. RuleML is formally defined by several XSDs and can be translated to other rule language, e.g. to JESS [41], using XSLT. Furthermore, it represents an integral part of the Semantic Web Rule Language (SWRL) [42]. The reasoning web service encapsulates OO jDREW [43], an open source reasoning engine with native support for RuleML.

5.3. Monitoring Web Service and Client

As the proposed 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) [44] tools. The monitoring web service uses this implementation to parse XACML, XBRDL as well as XPDL documents. Subsequently, it invokes the XACML PDP and the reasoning web service in order to detect control exceptions. Moreover, it performs an object-relational mapping for the entire model, creates an archive file containing corresponding flat files, and passes this file to the DW import web service. The monitoring web service may be invoked regularly, e.g. on a daily basis, and / or after changes. Relevant changes are, amongst others:

- an application developer changes the permissions required for an activity,
- a role administrator changes the permissions contained in a role,
- a user administrator changes the assignment of users and roles,
- a business process developer changes the participant associated with an activity or the role associated with a participant or
- an internal control auditor changes a control.

Provided that these changes are not immediately effective but consecutively transported through the stages of a multistage system concept (e.g. with separate test, quality and productive systems), control exceptions can already be detected before-the-fact. Therefore, problematic changes (e.g. caused by process optimization) can be prevented before they affect the productive system and thereby business objectives.

The monitoring client is used to configure and invoke the monitoring web service. It outputs a brief report regarding detected control exceptions. However, the business intelligence system is used for deep analyses and meaningful reports.

5.4. Data Warehousing Import Web Service

The DW import web service uncouples the monitoring web service from a particular BI system. In addition, it unpacks the received archive file to the right destination and optionally schedules a dedicated extraction, transformation and loading (ETL) process chain.

5.5. Enterprise Resource Planning System

The XACML role and permission policy sets used by the XACML web service and the XACML role assignment policy sets used by the monitoring web service originate from the access control data of a SAP ERP system. As SAP ERP is a leading business application and possesses a very sophisticated access control model, it is well suited for the following case study.

5.6. Business Intelligence System and Client

SAP ERP and the DW import web service are the data suppliers for a SAP BI system. Based on a history of snapshots, the SAP BI enables to analyze internal control under temporal aspects. Moreover, it is well suited for the provision of internal control information in a coherent way with other characteristics and facts to analysis and reporting tools. In particular, decision-makers, IT and process specialists may use the analysis and reporting clients of the SAP Business Explorer.

6. Case Study

The case study has been conducted for a subset financial process of an organization. This process is subject to compliance requirements and continuously supported by a SAP ERP system. The system is a multitenant system with over 1,200 users and is operated by the organization for about nine years. Figure 5 illustrates the activities performed for the case study from the description of the process to the creation of a report containing the detected control exceptions. These activities are detailed in the following subsections.
6.1. Process Description

In order to ensure information value, two requirements have been defined for the selection of a process for the case study: the process must contain (i) several participants and (ii) differentiated authorization assignments. In cooperation with the financials process owner of the organization, the “Documents and Payments” process has been selected and illustrated as in Figure 7. The process covers the registration and posting of documents as well as the preparation, review and execution of corresponding payment proposals. These activities correspond to transaction calls in the SAP ERP system. The use of a transaction requires certain permissions. These permissions are bundled up in roles and assigned to the different process participants in SAP ERP. Additionally, the role names are linked to the participants in the BPMN process description using standard modeling tools by means of XPDL extended attributes.

In this process, a compliance requirement which needs differentiated authorization assignments is to ensure segregation of duties. Some authorizations / roles must not overlap, e.g. a “Secretary of Department” may only register a document but is not allowed to post it.

6.2. Control Definition

Internal control is defined using XBRDL, e.g. the aforementioned segregation of duties control may be represented as illustrated in Figure 6.

```
subject_infringes_c (\$s c.regandpostregdoc) :-
  subject_has_ps (\$s, ps.registerdoc),
  subject_has_ps (\$s, ps.postregistereddoc).
```

The control (c.regandpostregdoc) is infringed by each person (s) which has both the permissions to register a document (ps.registerdoc) and the permissions to post a registered document (ps.postregistereddoc).

The control (c.regandpostregdoc) is linked to a formal definition of a control exception in RuleML. RuleML can be transformed from and to the Positional-Slotted Language (POSL) [45], which is easier to read for humans. The rule may be represented by the following POSL fragment:

```
subject_infringes_c (\$s c.regandpostregdoc) :-
  subject_has_ps (\$s, ps.registerdoc),
  subject_has_ps (\$s, ps.postregistereddoc).
```

This control is linked to a formal definition of a control exception in RuleML. RuleML can be transformed from and to the Positional-Slotted Language (POSL) [45], which is easier to read for humans. The rule may be represented by the following POSL fragment:

```
subject_infringes_c (\$s c.regandpostregdoc) :-
  subject_has_ps (\$s, ps.registerdoc),
  subject_has_ps (\$s, ps.postregistereddoc).
```

The control (c.regandpostregdoc) is infringed by each person (s) which has both the permissions to register a document (ps.registerdoc) and the permissions to post a registered document (ps.postregistereddoc). These critical permission sets are defined using XBRDL as well and linked to corresponding XACML targets. E.g. the permission to register a document may be represented by a XACML target with the action “register” and the resource “FV60”. Subsequently, process activities are linked to the names of controls and critical permissions using standard modeling tools by means of XPDL extended attributes.

While the definition of controls is a core competence of auditors, the definition of therein referred critical permissions may also be performed by IT specialists, in particular application developers.

6.3. Monitoring and Reporting

The monitoring web service combines the action values (e.g. “register”) and resource values (e.g. “FV60”) from the critical permissions with the subject values (e.g. a person “Copper, J.” with a role “secretaryofdepartment”) from the XACML role assignment policy sets and invokes the XACML PDP web service with respective XACML requests.
Based on the corresponding decisions, the monitoring web service passes assertions like the following to the reasoning web service:

subject_has_ps ('Cooper, J.', ps.registerdoc).

The rules contained in the controls are passed to the reasoning web service likewise. The reasoning web service therefore has the required information to infer answers for the query:

subject_infringes_c(?S, c.regandpostregdoc).

These answers are combined with the other information and passed to the DW import web service. The latter coordinates their import into SAP BI.

Figure 8 contains a screenshot of SAP Business Explorer Analyzer showing some example queries regarding processes, internal control, control exceptions and permissions. The presentation in a BI tool enables process owners to work with internal control reports within their familiar analysis environment. This has potential to increases the acceptance and usage rate. Furthermore, process owners are now independently and timely in the position to recognize control exceptions within their area of responsibility. In contrast to periodical audits made by varying consultants, BI reports deliver homogeneous information tailored to individual requirements for people with different backgrounds at any time. Furthermore, data histories enable process owners to analyze the status of their business processes under temporal aspects. This enables an easy monitoring of internal control, e.g. controls related to objectives set by the organizations business strategy.

7. Conclusion

Although the importance of risk management and the monitoring of business processes and internal control in organizations lately get recognized and high awareness is attached, the implementation of corresponding approaches remains difficult. These difficulties are addressed by a Design Science Research approach. A model for the enrichment of business processes with internal controls, user roles and permissions is presented. Additionally, an architecture for the automated monitoring of internal controls and the timely communication and deep analysis of thereby detected control exceptions using business intelligence (BI) is proposed.
The major advantage of this approach is the combination of a widely understandable business process notation (BPMN) with an established internal control model (COSO), an XML based rule language (e.g. RuleML) and a platform independent access control policy standard (XACML). The adherence to this standard models and technologies enables the reuse of existing components and tools. Furthermore, the use of BI offers merits for participants with different professions on multiple organizational levels. Amongst others, the easy access to information facilitates the individual development of competencies and problem awareness. Therefore, the approach does not only meet particular (short term) information needs which are considered important at the launch of corresponding projects (e.g. by sponsors from the top management), but thoroughly meet the needs of many participants in the long term. Consequently, both strategic and operational information needs can be adequately addressed likewise. This appropriately integrates risk management into the decision-making process and thereby contributes to the improvement of security, the mitigation of risks and the achievement of business objectives.

To increase the degree of confirmation with respect to the feasibility and suitability of the proposed model and architecture, a prototype with a Service Oriented Architecture (SOA) has been implemented in a SAP ERP and BI environment using model-driven architecture (MDA) principles. This prototype has been used in a comprehensive case study to outline the benefits of this approach.

Future research will be dedicated to the evaluation of the prototype with real-life workloads. The performance of the individual components will be measured in order to identify potential bottlenecks. Furthermore, the prototype is currently extended with automated transformations from proprietary access control models to XACML.

References


[20] PrivilEge and Role Management Infrastructure Standards Validation (PERMIS), www.permis.org