A Model-driven Approach for Internal Controls Compliance in Business Processes

Kioumars Namiri[1], Nenad Stojanovic[2]

[1] SAP Research Center CEC Karlsruhe, SAP AG, Vincenz-Prießnitz-Str.1
76131 Karlsruhe, Germany
Kioumars.Namiri@sap.com

[2]FZI Karlsruhe, Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
Nenad.Stojanovic@fzi.de

Abstract. Enterprises require mechanisms to ensure that their business processes implement and fulfill internal controls in context of regulatory compliance such as Sarbanes Oxley Act. In this paper we propose an approach for the modeling and implementation of internal controls in business processes. The approach is based on the formal modeling of internal controls, thus it can serve as the basis for usage of logic mechanisms in the compliance verification process.

1 Introduction

The advent of regulatory compliance requirements such as Sarbanes Oxley Act 2002 (SOX)[1] requires the implementation of an effective internal controls system in enterprises. COSO[2] defines the internal controls as a process designed to provide reasonable assurance regarding the achievement of objectives in effectiveness and efficiency of operations, reliability of financial reporting and compliance with applicable laws and regulations. We focus on the Application Controls (AC) controlling business processes and propose the introduction of an abstraction layer above a business process, in which these controls are formally modeled and evaluated against existing process models and instances. We see several advantages of such an approach:

- It enables usage of formal methods for the verification of a business process's compliance.
- Consequently the compliance can be performed automatically based on the current state of a process
- The changes of the controls will not affect the design and execution of the original business processes
- Non-experts can built on top of the domain model provided to design controls for business processes

2 Motivating Scenario

The internal controls compliance of a purchase ordering process (PO) depends on enterprise specific risk assessment carried out by auditing consultants (see Table 1)

Table 1 Risk assessment on Purchase Ordering Process (PO) for an enterprise

Control Objective	Risk	Application Control
Prevent unauthorized use of PO Process	Unauthorized creation of POs and payments for not existing suppliers	Double Approvals of POs higher than $5000 (*Double-Check-Control*).

[1] Pub. L. 107-204. 116 Stat. 754, Sarbanes Oxley Act (2002)
[2] Committee of Sponsoring Organizations of the Treadway Commission (COSO)

In: M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, N. Stojanovic (eds.): Semantic Business Process and Product Lifecycle Management. Proceedings of the Workshop SBPM 2007, Innsbruck, April 7, 2007, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-251/

3 Domain Model for Internal Controls Compliance

The design of a control should control the way a business process is executed. A (re)design of a business process causes an update of risk assessment on a business process, which may lead to a new or updated set of the controls incl. new tests. The business process monitoring and verification techniques may be used to assess the design of controls and serve as an input to the compliance certification (See Figure 1).

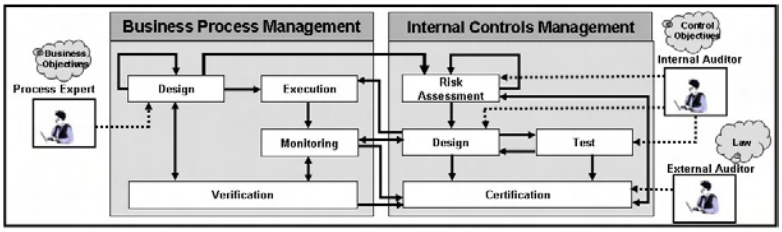

Figure 1 Relations between BPM and Internal Controls Management

The main entities for the process of internal controls compliance is described in following and illustrated in Figure 2a: Identify all **significant accounts** in the company. Identify for those accounts all **business processes** affecting them. Define for each relevant business process a set of **control objectives** specific to the enterprise. Assess the **risks** for the enterprise by their identification for each control objective. Design and implement based on the risk assessment a set of **controls** in order to prevent or detect the occurrence of the identified risks.

An *Application Control* (AC) controls different dimensions of the way a business process is enacted, namely the execution of its *activities*, the *Business Documents* involved and the *agents* performing an *activity* including their *authorities* (See Figure 2b).

For each *AC* at least one *Recovery Action* must have been designed, which reacts on the violation of a control. It does *not* change the designed business process logic; it rather blocks the transaction and may send a notification to an assigned responsible agent.

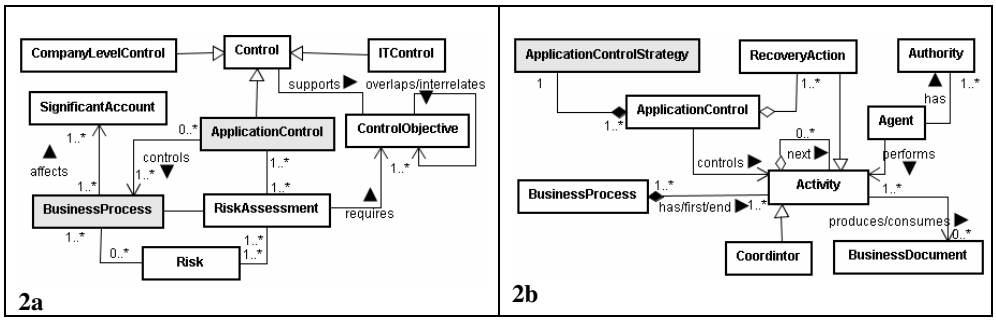

Figure 2a - The upper domain model of the Internal Controls Compliance
Figure 2b - Relationship between an Application Control and a Business Process

Application Control Strategy Model

An *Application Control Strategy* defines the way a control monitors the behavior of one or more activities inside a business process (Figure 3). In order to become active an *AC* requires to be triggered according to the *state* of the process parameters in a *scope*. We define further two elements of an AC strategy: *scope* and *pattern* based conceptually on the work done by Dwyer et al [1]. Although their patterns are mainly used for defining formal requirements on program specifications, they can be applied to internal controls compliance and the monitoring requirements there. For a detailed description of the scopes and patterns and their semantics please refer to [1].

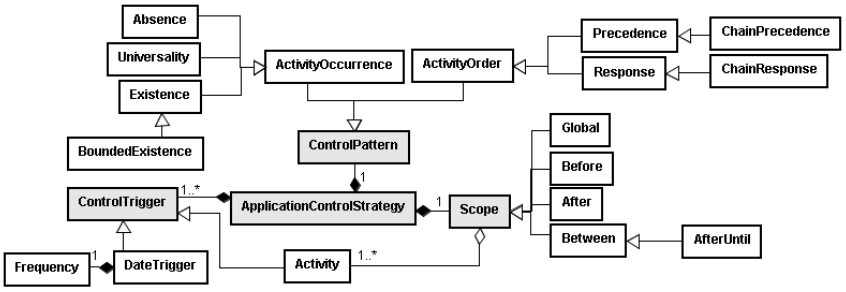

Figure 3 A Semi-formalization of the control implementation

4 The Approach

The abstraction layer above business process model we call the "Semantic Process Mirror" (*SemanticMirror*). According to assessed risks, a set of ACs is defined in this layer. During execution of a business process, this layer will be updated with information needed for the evaluation of defined controls in order to ensure that compliance checks will pass. The approach spans over there phases:

Phase 1: Semantic process mirror design phase
SemanticMirror represents a semantic layer placed on the top of the (usual) syntactical description of a business process (i.e. workflow). In this phase a model of the business process according to Figure 2b will be stored in the SemanticMirror. It will be used later during the phase 2 and 3 to infer whether the process is designed and executed according to a set of declaratively designed ACs in phase 2.

Phase 2: Application control design phase
In the following we present a set of formalizations needed for the automatic evaluation of ACs.
Control statement CS is a logical statement that describes how to carry out an AC *ac* in a business process *bp*:

$$CS(ct, bp, ac(x, cp), GS(bp, scope(M)), \text{action}_R) :=$$
$$O(ct) \wedge V(bp, ac(x, cp), GS(bp, scope)) \rightarrow Activity(bp, \text{action}_R),$$

where the formula for *CS* expresses that if a *violation V* for the given *ac* occurs (is true) after *occurrence O* of a *ControlTrigger ct* on a *Guarded Sequence GS*, then the corresponding *recovery action* action_R will be instantiated and executed on current instance of *bp* (the instance that generated the violation). We describe the parameters mentioned above: *Guarded Sequence* is a sequence of activities, which are along the *scope* of the *AC strategy* of an *ac* in a *bp*. The values for the violation of a control are calculated by evaluating the statement *ac* on the SemanticMirror, i.e. if the statement *ac* can be inferred from the set of facts contained in the SemanticMirror.

An *AC ac* expresses that a *control pattern cp* (See Figure 3) must hold if the logical condition on an entity *x* holds:

$$ac(x, cp) := condition(x) \rightarrow cp,\ x \in \{BusinessDocument, Agent\}$$

We show the formalization of the control pattern (cp) *BoundedExistence* of *n* (see Figure 3) for an *activity C* in the scope of activities defined by *GS*(*bp,scope*)*:*

$$\text{BoundedExistence}(n, C, GS\ (bp, scope)) :=$$

$$(\bigwedge_{i=0,..,n} \exists C_i \mid \text{InstanceOf}(C_i, C)) \wedge (\forall_{i,j=0,..,n} C_i, C_j \mid C_i != C_j) \wedge (\overset{n}{\forall}_{i=0,..,n} C_i \mid C_i \in GS\ (bp, scope))$$

Example: Applied on the Double-check control in the PO-Process (see scenario) the statement *ac* looks as follows:

$$\forall PO \mid BusinssDocument(PO) \wedge Amount(PO, amount) \wedge greater(amout, 5000) \rightarrow$$

$$BoundedExistence(2, ApprovePO, GS_{DoubleCheck}(P2P, Between(SelectSupplier, SendPO)))$$

Phase 3: Business process execution phase

This phase enables the bidirectional interaction between BPM and internal controls management (see Figure 1): The SemanticMirror will be updated by information about the current instance of the business process enacted and if an AC is violated, the recovery action defined in the control statement will be executed. KBAs represent conceptual abstraction of a log channel, which maybe used to update the SemanticMirror.

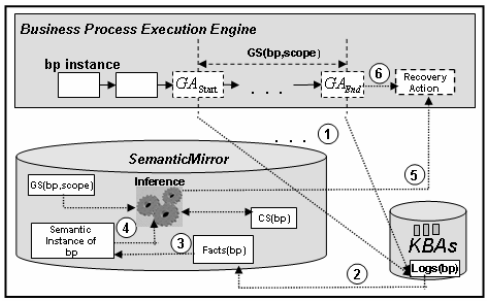

Figure 4 Business process execution phase

5 Related work and conclusion

In this paper we introduced a semantic based approach for conceptual modeling of internal controls required by regulations such as SOX. The controls are captured declaratively and checked during execution-time of business processes. On a conceptual level our work is related to [2], where a taxonomy of risks for business processes is provided. In [3] the logic behind the obligations and permissions on a business process and contracts is made using temporal deontic logic. [4] gives an overview and discusses the current industrial software products in this area and their limitations.

References

1. M. Dwyer, G. Avrunin, J. Corbett, Patterns in Property Specification for Finite-State Verification. In Proceedings of the 21st International Conference on Software Engineering, pages 411-420, May 1999
2. zur Muehlen, Michael; Rosemann, Michael. Integrating Risks in Business Process Models. In: Proceedings of the 2005 Australasian Conference on Information Systems (ACIS 2005), Manly, Sydney, Australia, November 30-December 2, 2005.
3. Guido Governatori, Zoran Milosevic, and Sahzia Sadiq. Compliance checking between business processes and business contracts 10th International Enterprise Distributed Object Computing Conference (EDOC 2006). IEEE Press, 2006, pp. 221-232
4. R. Agrawal, Ch. Johnson, J. Kiernan, F. Leymann: Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. Proc. 22nd Int'l. Conf. on Data Engineering ICDE'2006 (Altanta, GA, USA, April 3 – 7, 2006)