Applying Semantics to Sarbanes Oxley Internal Controls Compliance

Kioumars Namiri[1], Nenad Stojanovic[3]

[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: The advent of regulatory compliance requirements such as Sarbanes Oxley Act has forced enterprises to set up a process for managing an effective internal controls system. We propose the introduction of a semantic layer in which the process instances are interpreted according to the required compliance controls represented as rules. We analyze in this paper the requirements for the implementation of the approach using SWRL from software architectural perspective.

1 Introduction

Efficient compliance management has become a very important issue for the successful businesses. Regulation such as Sarbanes Oxley Act 2002 (SOX) [Sox02] requires the implementation of an effective internal controls[1] system in enterprises. The realization and test of effectiveness of the internal controls is considered to be expensive and time consuming [HFL05].

A part of the problem is the ad-hock implementation of the regulations, i.e. there is no clear separation of the business and control objectives in a business process. Indeed, current approaches for the automation of the internal controls integrate the controls too tight in the business processes making the processes and controls less adaptable and reusable. Formal modeling of internal controls can help in abstracting from concrete implementation and indirectly support the reusability of a solution.

[1] Internal controls is 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.

However, the main problem of a formal approach is the efficiency of its realization/technical implementation. Indeed, although a formal approach provides usually an elegant conceptual solution, its concrete implementation can suffer from the drawbacks like high-modeling effort and non-scalability, which are crucial issues for an industrial level implementation.

In this paper we introduce the requirements for implementing the approach and discuss them when using SWRL to develop the conceptual layer for the internal controls on top of the business processes.

2 Motivating Scenario

The internal controls compliance of a purchase ordering process (PO) depends on enterprise specific risk assessment carried out by consultants A control may look as follows: *"All Purchase Orders (POs) with an amount higher than 5000$ must be approved by 2 different purchasers (Double Check Control on Approve POs)".*

3 Realization of the Approach

We elaborate on the technical challenges we are facing with via the prototypical implementation of the approach. In our approach, the abstraction layer above business processes we call the "Semantic Process Mirror". According to assessed risks in an enterprise, a set of controls is defined in this layer. By executing a business process, the semantic process layer will be continuously updated with information needed for the evaluation of defined controls in order to ensure that compliance checks will pass.

3.1 Open issues

There are two open issues that have to be discussed from the implementation point of view: 1) How to design and execute business processes and 2) How to implement the SemanticMirror. The first issue is a fixed building block in the architecture in terms of an already existing ERP system/BPM Execution Engine. Second open issue will be discussed in the next subsection.

3.2 SemanticMirror implementation

The basis for our implementation of SemanticMirror is the semi-formalized domain model of the internal controls as introduced in [NS07]. We express the control as Event-Condition-Action (ECA) rules, since most controls are typical use case for ECA-rules. Further, we defined four requirements on the language and the used rule/inference engine for representing and executing the controls. In following we first describe each requirement and elaborate on each requirement how far SWRL as the most popular Semantic Rule language respectively a SWRL-Inference-engine supports them.

Requirement 1) Business Process Context awareness of SemanticMirror
The current state of SemanticMirror, i.e. the facts it contains, is heavily dependent on the current business process context running inside the BPM Engine/ERP system. We define the Business Process Context as a set containing the token showing on the last enacted activity in a business process instance and the current workflow relevant data, i.e. business documents, current user information etc. Thus the state in SemanticMirror is determined based on the context determined from outside the SemanticMirror. In following we explain further two interactions marked with A and B in the Figure 1a:

Interaction A: Rule/Inference Engine adds/updates facts in SemanticMirror
As a Rule representing a control is executed by a Rule/Inference Engine, it may add new facts, or update already existing facts in SemanticMirror.

Example: Considering the Motivating scenario (Section 2), if SemanticMirror determines that because a PO's amount (an already existing fact in SemanticMirror, e.g. *poId=4711)* is lower than 5000$, the Double Check Control is not violated even there is only one ApprovePO-Activity. This is signaled by creating the fact *POStatus*(*poId=4711,* ***status="APPROVEABLE"***). If a Purchaser has updated her PO by setting the amount to 6000$, then before the execution of SendPO-Activity the business process control flow is returned back to an earlier state in the process again. The Rule/Inference Engine determines now that for the same PO instance, the Control is violated and the fact has to be updated: *POStatus*(*poId=4711,* ***status="NOT_APPROVEABLE"***).

Interaction B: BPM Execution Engine adds/updates facts in SemanticMirror
The enactment of a business process inside a BPM execution engine may cause creation of completely new facts or update an already existing fact in SemanticMirror.

Example: A fact representing a PO already exists in SemanticMirror: *PO*(*poId=4711*, ***amount=4500***). The production planning realizes later that for the same material type a higher stock is required in the warehouse. Thus the purchaser updates that PO instance by increasing the amount of PO document to 6000$, the previously added fact to SemanticMirror has to be updated to: *PO*(*poId=4711****, amount=6000***).

When implementing SemanticMirror with SWRL, we have the situation that we have to synchronize the SemanticMirror as an open world system with monotonic assumption continuously by business process context provided through an existing close-world data-base-centric object-oriented system (BPM Execution engine/ERP). When it comes to synchronization of the open world SemanticMirror, the originally existing closed world system forces us to simulate the closed-world behavior in an open-world environment. This is because when evaluating a control. i.e. when executing a rule by the inference engine, we have to assume that every fact used by the rule is referring to the most recently provided business process context in BPM Execution engine. Otherwise, the execution of the rule in an outdated business process context may result in a different conclusion regarding the violation of a control.

The approach actually works in closed world environments, where each property of a specific fact maintains its most recent value; in other words when a new value for a fact is provided, the previous value is overwritten. But in an OWL/RDF implementation of the facts in the SemanticMirror, the approach does not work straight forward because of the monotony. This issue is discussed in [MKBL05]. However, both solutions proposed there to overcome the issue will increase significantly the required computation resources, coding efforts and complexity for the approach.

Requirement 2) Expressivity
The language must provide constructs to directly or indirectly express the control as ECA-rules. SWRL fulfills on an Expressivity level our requirements for implementing the approach.

Requirement 3) Actionable output to business processes
The approach requires that in case of determining a violated control, an advised activity by an internal auditor codified in the control in terms of the recovery action is executed. This means actually that in case of determining a violation in the SemanticMirror, the state of things in outside world, i.e. the current business process instance running in a BPM Execution engine, has to be updated according to the recovery action. SWRL allows using constructs in consequent–part of a rule statement or built-ins which are modeled in OWL previously. SWRL does not provide any direct mechanisms to invoke operations outside of the OWL/SWRL-knowledge base. When designing SemanticMirror with OWL and SWRL, we have to realize the architecture in such a way, that a separate component has to be developed, which queries the SemanticMirror in order to determine whether a control is violated or not (pulling the SemanticMirror) and accordingly updates the business process instance in the BPM Execution engine.

Requirement 4) Querying external backend systems
This requirement is closely related to the first requirement and similar on a technical level to the third requirement with the same results when using SWRL: During our analysis of different types of controls, we realized that it will be very expensive to keep the SemanticMirror completely synchronized with the heterogonous environment we are facing with.

The heterogeneity is given through different backend systems containing different operational data such as SRM (Supplier Relationship Management), CRM (Customer Relationship) systems etc, which contain relevant information about orders, contract, business transactions etc. Figure 1b illustrates the situation.

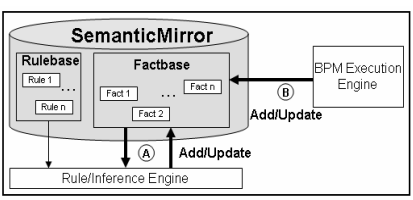

Figure 1a: Facts Management

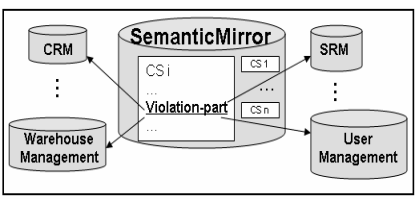

Figure 1b: Control Evaluation

4 Conclusion

Based on the discussion above, we decided to implement a prototype for the SemanticMirror not with SWRL. While we have shown that in a static environment when it comes to verification of the design of business processes [NS06], SWRL is a suitable candidate to be used, in the described context with frequently changing transactional environment, SWRL is not the first candidate of choice. We see in our compliance research the semantic technologies as a technologies issue at an implementation level. The current tooling support in OWL and SWRL context does not hide the logic complexity in such a way that we could develop a prototype and hand over it to a development team being more software developers than logic experts. Moreover, the current state of SWRL/OWL would force us to generate much more code to bridge the gap between a Closed-World-System (ERP/BPM Execution Engine) and an Open-World-System (SWRL/OWL based SemanticMirror). A SWRL-Based realization of SemanticMirror would force us to realize a Pull-Driven-Architecture, which complicated our system. We are currently in the process of implementing the approach in a prototype with a java based Business Rule Engine System, which will be subject to one of our later reports.

References

[HFL05] Hartman, T.; Foley & Lardner LLP, "The Cost of Being Public in the Era of Sarbanes-Oxley," June 2005

[MKBL05] Matheus, C.; Kokar, M.; Baclawski, K.; Letkowski; J.: Using SWRL and OWL to Capture Domain Knowledge for a Situation Awareness Application Applied to a Supply Logistics Scenario. In Proceedings of International Conference on Rules and Rule Markup Languages for the Semantic Web, RuleML-2005, Galway, Ireland, November, 2005

[NS06] Namiri, K.; Stojanovic, N.: Towards Business Level Verification of Cross-Organizational Business Processes. Proceedings of the Workshop on Semantic Business Process Management (SBPM 2007), Budva, Montenegro, June 2006.

[NS07] Namiri K.; Stojanovic N.: A Formal Approach for Internal Controls Compliance in Business Processes, 8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), In conjunction with CAiSE'07

[Sox02] Pub. L. 107-204. 116 Stat. 754, Sarbanes Oxley Act (2002)