Event-Based Decision Management in Compliance Management - A Discussion Paper

Marwane El Kharbili

ARIS Research, IDS Scheer AG, Altenkesseler Str. 17, D-66115 Saarbrücken, Germany.
E-mail: marwane.elkharbili@ids-scheer.com

Abstract. Compliance is a critical enterprise management concern, particularly in business process-oriented organizations. We introduce an event ontology part of a policy and rules ontology for compliance modeling and enforcement. The policy and rules ontology is able to take decisions depending on the state of an enterprise model. In our work, events are used to support modeling complex decision-making patterns and propagation. We present an ontology for modeling events for compliance policies and rules and discuss its use in our compliance management framework.

1 Introduction

Compliance management has been identified as one core topic of research for making business process management (BPM) the enabler for more reliability and business sustainability. In [1], we motivated the need for compliance management in BPM and motivated the use of policies and rules for compliance modeling and enforcement. A framework for compliance management in BPM was introduced and its architecture discussed. In [2], we explained how using policies and rules together with domain policies can be used for modeling compliance. In modeling compliance policies and rules, events play an important role. In our framework, policies are used for modeling decisions and rules encompass the logic attached to making these decisions, depending on the context of policy enforcement. Rules are also responsible for deciding which actions need to be taken and by which entity under jurisdiction of a policy. However, modeling decisions is still made hard for the class of decisions which cannot be directly executed by a rule directly linked to a policy (simple decisions). The other class of decisions, called complex decisions, may request calling several rules not directly linked to the decision-making policy or may require propagating a decision to other dependent rules for further processing.

In our approach, we separate between decision-making and action-taking. Decisions can request other rules to make other decisions and/or request a responsible entity to execute some action (fig 1). Our goal is to allow modeling complex decision-making by using events as a mean of linking decision-making units (business rules) to other decision-making units as in figure 1. In the following, we motivate this use of events by giving an use case scenario and explain our approach to realizing this. We then show how our approach integrates in the SUPER semantic BPM (SBPM) framework and then conclude our proposal.

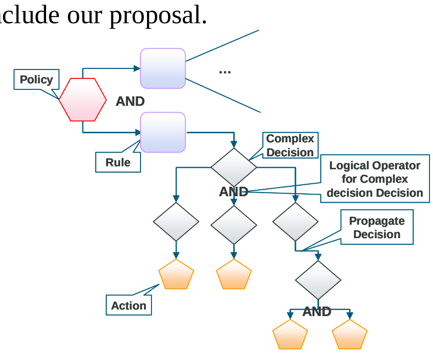

Fig1.: Decision-Making Tree: Policies, Rules, decisions, Actions.

2 Event Management in a Compliance Framework

Let us take the example of a security rule for intrusion detection. A system is under jurisdiction of a security policy P. As individual I accesses a resource of system S, an intrusion detection rule (called IDR) fires and (using certain intrusion detection logic provided by a security rule) decides that I is an intruder. Another rule (called NF), which decides which person should be notified about this intrusion and another rule (called IDP), which decides on further actions that need to be made depending on the resource accessed by the intruder should also fire. Rules IDP and NF are dependent on rule IDR. Using events, we can easily link rule IDR to the NF and IDP rules. This can be done by making rule IDR generate a well defined event and subscribing rules NF and IDP (configuring NF and IDP to listen) to this event. The event carries among other information the ID of the rule that has fired it and the protected resource being accessed. Events become thus the link that provides declarative automation in modeling complex decision-making in our compliance framework. We think that this idea can be easily generated to all rule frameworks for modeling complex decisions.

In [3], a survey of event-driven architecture (EDA) is given that also proposes event modeling for reacting to situations and making decisions. Here, formal modeling of situations is used, while we rely on enterprise model state-based context definition from our compliance framework [1]. In [4], the authors propose a logic based formalism for modeling events in an EDA. Both works recognize the same issues as we do. Next, we introduce a core ontology for modeling events for policies and rules (BPREO) of the compliance framework (fig. 2). Central concepts are the rule and policy concepts. Policies are used to model compliance measures and rules contain the necessary logic for enforcing these policies. A policy acts on a subject which itself can occur in a business process (e.g. artifact such as BP activity), or can be any type of resource (e.g. role individual or network printer) supported by the policy. Subjects have states and belong to an event scope, which is the set of subjects which can produce/be influenced by this event. For example, a resource access event is fired when the state of a resource changes from **unaccessed** to **accessed**. Rules and policies also belong to a logical event scope, which is the set of policies and rules which can trigger or are activated (policy)/invoked (rule) by this event. The input event type is used by rules as an invocation input and transports information necessary for the rule to correctly execute. The output event type is used as a mean to communicate to the compliance framework information about rule execution and transports instructions on how to further conduct the process initiated by the policy having been activated.

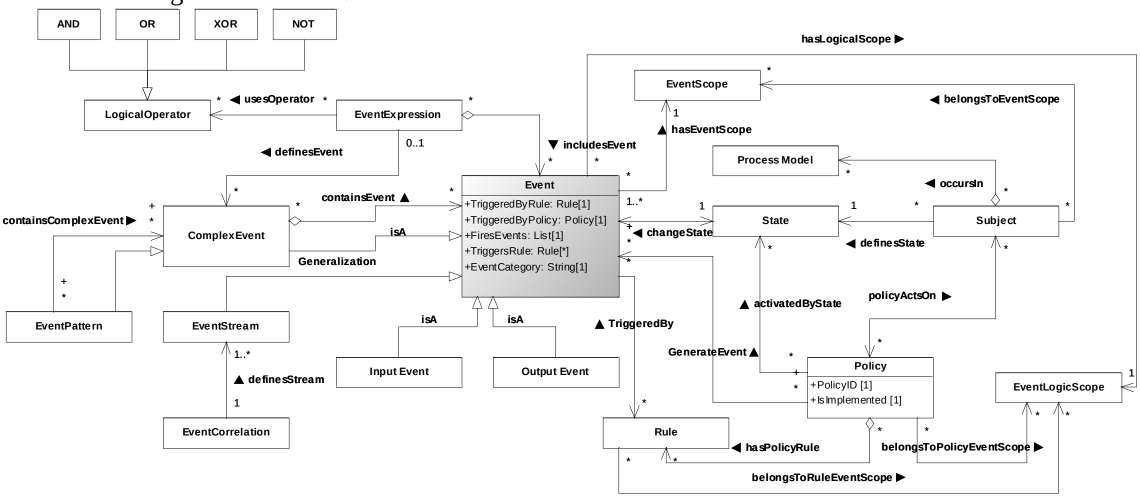

Fig 2. A Core Event Ontology for the Business Policy and Rules Ontology.

An event can itself fire other events if necessary. An event can be itself implemented as one of two types: (i) **Complex Events** and (ii) **Event Streams**. These both approaches to event processing are identified and implemented by existing major complex event processing (CEP) solutions. Events can be event expressions which combine atomic events in logical events using Boolean operators (**AND, OR, NOT, XOR**) to express complex events. We give in that simply using Boolean operators for complex events is not expressive enough. First of all, complex events may be defined not only using event occurrences (e.g. **Ev1 AND (Ev2 OR Ev3)**) but also using meta-data transported by these events (e.g. **Ev1.Att1 == Value1 AND Ev2.Att2 == Value2).** Complex events could thus be a combination of event occurrences and rules expressed on the state of the system modeled using event meta-data. This shows the high inter-dependency between events and rules, as events are used to model complex business decisions, and other rules are used to model complex events. Secondly a complex event could be itself a tree of logical expression of events, with each event being at the same time an event and a node that has a sub-tree of events attached to it. This view maps directly to fig. 1 where decision nodes could at the same time be business rule invocation and a call to some action taking entity. This means that the decision-making tree example given in fig. 1 could be implemented using a complex event in the form of an event tree, with rules attached to each node that contain policy enforcement logic. Each node in the tree is actually a Boolean expression of the events in its sub-tree. Event patterns form reusable sets of complex events expressed in generic way, possibly using place-holders for both event types in Boolean event expressions and rules on event meta-data, as well as pre-defined event trees.

Another way of modeling events would be event stream processing (ESP). The STREAM[1] system tackles just this issue (more fundamentally for data streams) and the open source ESPER[2] tool provides CEP/ESP functionalities. ESP engines rely on modeling correlations patterns between singletons in an event stream and offer a query language[3] for identifying patterns (logical and temporal constraints). BPM and Middleware vendors also incorporate tools for CEP such as ARIS Process Event Monitor[4] or TIBCO Business Events[5]. Our core ontology needs to interface with such systems in order to be able to process complex events as far as event processing doesn't require it to be done on a formal semantic level. This question will be evaluated in future works where we will design an architecture for event-based decision-making for compliance management, together with the problem of serializing policy and rule ontology events in formats processable by CEP and ESP tools. Another interrogation of ours is how to combine our event ontology with the SUPER EVO ontology of events for business process analysis (BPA). As policy-based compliance controlling (monitoring and analysis) is one component of our compliance framework, we will attempt using events for logging relevant information for later analysis purposes or even real-time compliance analysis.

3 Conclusion

We devised a core event ontology for supporting policy-based decision management, for use in compliance management. We will next seek to design an architecture for modeling compliance policies using complex event processing, and integrate this architecture in the SUPER platform for semantic business process management.

References

1. El Kharbili, M.; Stein, S.; Markovic, I. & Pulvermüller, E. Towards a Framework for Semantic Business Process Compliance Management. In *Proceedings of the workshop on Governance, Risk and Compliance for Information Systems, June 2008* (pp. 1-15).
2. El Kharbili, M.; Stein, S.; Markovic, I. & Pulvermüller, E. Towards Policy-Powered Semantic Enterprise Compliance Management. In *3rd International Workshop on Semantic Business Process Management (pp. 16-21)*, June 2008.
3. Kay-Uwe Schmidt, Darko Anicic, and Roland Stühmer: Event-driven Reactivity: A Survey and Requirements Analysis. *In 3rd International Workshop on Semantic Business Process Management (pp. 72-86)* June 2008.
4. Stojanovic N Anicic D. Towards creation of logical framework for event-driven information systems. In: *10th International Conference on Enterprise Information Systems, 2008.*

---

[1] Stanford's STREAM project: http://infolab.stanford.edu/stream/

[2] http://esper.codehaus.org/

[3] Continuous Query Language (CQL) from STREAM and Event Query and Causality Pattern Language from ESPER

[4] www.ids-scheer.com/en/ARIS/ARIS_Software/ARIS_Process_Event_Monitor/73867.html

[5] http://www.tibco.com/software/complex_event_processing/businessevents/default.jsp