Modeling and Enforcement of Business Policies on Process Models with Maestro

Ivan Markovic[1], Sukesh Jain[1], Mahmoud El-Gayyar[2], Armin B. Cremers[2], and Nenad Stojanovic[3]

[1] SAP Research, Karlsruhe, Germany
{ivan.markovic,su.jain}@sap.com
[2] Department of Computer Science III, University of Bonn, Germany
elgayyar@cs.uni-bonn.de, abc@iai.uni-bonn.de
[3] FZI, Karlsruhe, Germany
nenad.stojanovic@fzi.de

Abstract. Business policies and rules govern and guide the business processes of an organization. Enterprises usually only document their business policies and rules in natural language. This makes the procedure of determining which business policies and rules apply to a certain process and their enforcement on this process very costly and cumbersome. We present a tool that supports formal specification of policies and rules and their automated enforcement on process models. We explain the research background underlying the tool, and we overview what will be demonstrated at ESWC.

1 Introduction

Directives within an organization exist to govern and guide its business processes and they are categorized as business policies and business rules. According to OMG's Business Motivation Model (BMM) [1], a business policy is a high level directive that exists to control, guide and shape how an enterprise realizes its courses of action. Business policies define what is allowed/not allowed to be done, and may indicate how or pose constraints on how it should be done. Business rules on the other hand are derived from and used as a mechanism for implementing business policies.

Usually enterprises either do not capture directives formally or just document them in natural language. This makes the procedure of determining which directives apply to a certain process and their enforcement on this process very costly and cumbersome. In order to support explicit specification and automated enforcement of business policies and rules on business processes, we have devised an approach supported by semantic technologies and the process-oriented enterprise ontology framework [2]. In our approach we semantically model business policy and business rules which implement a policy. The approach is based on the notion of anti-patterns for process models [3], i.e. explicitly capturing the violating scenario and looking for its occurence in a process model, rather than checking for the correctness of a process model. Thus, we explicitly capture violation of

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 873–877, 2009.

a business policy in terms of formalized business rules which implement this policy and enforce them as constraints on process models. The approach is prototypically implemented in the form of extensions to *Maestro*, an SAP Research modeling tool. We explain our approach, tool support and various technologies used to realize this tool using the *Separation of Duty* (SoD) policy [4] as an example scenario for its wide application as a security mechanism in the financial management domain.

There have been several attempts to support specification of business rules and their enforcement. ARIS Design platform offers predefined "ARIS Semantic Checks" to ensure modeling standards are adhered by business process models [5]. For programming additional checks, the user must be familiar with the underlying database schema in which models are stored and also with the scripting language. This results in logic of checks being hard-coded and scattered. In [6], the rules are modeled as simple IF/THEN textual rules and lack visualization. Moreover, the work does not show how complex nature of processes, i.e. abstraction levels and other enterprise models, are handled.

2 Business Policy and Rule Specification

In order to support the visual modeling of business policies and rules, we have developed a new modeling perspective within *Maestro*, named *Maestro4BPR* (see Fig. 1).

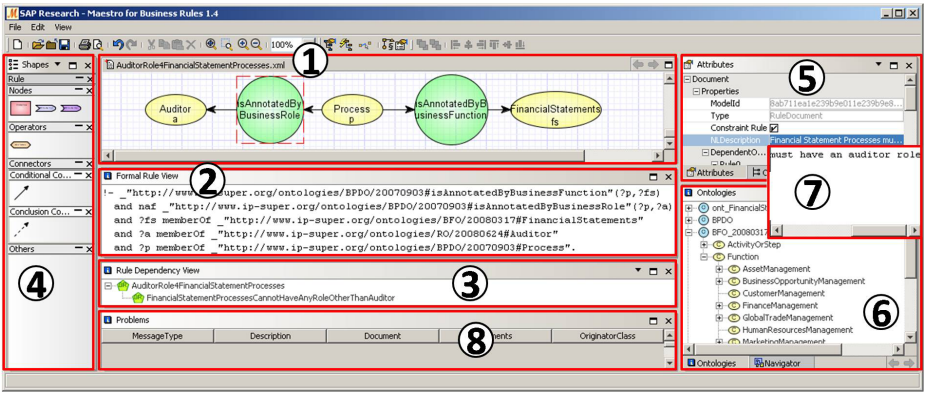

Fig. 1. Visual modeling of business rules using Maestro4BPR

The user interface of Maestro4BPR is built using the *Tensegrity Graph Framework*[1] which provides features for introducing new visual elements, adding new views, auto-layouting algorithms etc. The tool allows users to build business rules by dragging and dropping concepts, instances, relations and attributes

[1] http://www.tensegrity-software.com/

defined in any application ontology[2]. The *Ontologies* view (see no. 6 in Fig. 1) provides functionality of loading the required application ontologies for building a business rule. The ontologies are built using Flight variant of the *Web Service Modeling Language* (WSML) [7]. *WSMO4J*[3] provides the necessary Java API for ontologies described in WSML. The user does not need to have any prior knowledge of WSML language, as the rules are built by dragging elements defined in various application ontologies and additional elements provided in the *Shapes* view (see no. 4 in Fig. 1) to the *Modeling* view (see no. 1 in Fig. 1).

The business rule in Fig. 1 represents one of the violation scenarios of SoD policy. The oval and circle shapes (see no. 1 in Fig. 1) indicate respectively the concepts and relations between concepts defined in the application ontologies. The dotted square (see no. 1 in Fig. 1) around relation (circle) represents the *negation as failure* (naf) construct [7]. Business rule in Fig. 1 indicates that a violation occurs when any *process* '*p*' that is annotated with any instance of *FinancialStatements* business function '*fs*' and not (naf) annotated with any instance of *Auditor* role '*a*'. WSML Flight works with variables (*p*, *fs*, *a*) which are then checked with concrete instances for violation during enforcement phase. The concepts *FinancialStatements*, *Auditor and Process* are defined in the functional, organizational and process perspective in [2], respectively. In addition, users can specify natural language description (see no. 7 in Fig. 1) and other properties of the modeled rule in the *Attributes* view (see no. 5 in Fig. 1).

Visually modeled business rule is automatically transformed to WSML Flight logical expressions in the background (see no. 2 in Fig. 1), thus hiding the complexity of the formal language from the user. The tool provides possibility for users to specify other related business rules which need to be enforced along with the currently modeled one (see no. 3 in Fig. 1) in the *Rule Dependency* view. The related business rule in Fig. 1 (see no. 3) depicts violation scenario when *FinancialStatements* processes have other roles in addition to the *Auditor* role. It complements the first business rule in that it prevents the users from specifying other additional roles to *FinancialStatements* processes. This helps the users to organize related business rules. In addition, *Maestro4BPR* assists users by providing them feedback while modeling a business rule through the *Problems* view (see no. 8 in Fig. 1).

When modeling a business policy, the tool allows to model its context (applicability criteria) by dragging and dropping instances defined in application ontologies, e.g. assigning an organizational unit instance would mean that the modeled business policy is applicable to business processes involving this organizational unit. In addition, the user can define independent business rules and other business policies that together implement the currently modeled business policy. For the SoD scenario, in addition to the above mentioned business rule for *FinancialStatements* processes, similar business rules are required for *Accounting* processes to ensure that only the *Accountant* role is allowed to perform

[2] Application ontologies are domain-specific ontologies built using the process-oriented enterprise ontology framework [2].

[3] http://wsmo4j.sourceforge.net/

them and a business rule ensuring that no users defined in the organizational structure model have both roles i.e. Accountant and Auditor. These independent business rules together implement SoD business policy. Finally, business policies and rules are formalized as instances of *Business Policy & Rule Ontology* [2].

3 Policy Recommendation and Enforcement

Not all business policies are applicable for each business process, hence the user needs to be assisted in finding the relevant ones for a given process model. *Policy Recommender* application (see no. 1 in Fig. 2) addresses this issue using matchmaking technique to provide a ranked list of relevant policies for the currently modeled process. This application is an extension of the *Maestro4BPMN* modeling view used for designing processes in the BPMN[4] notation (see Fig. 2).

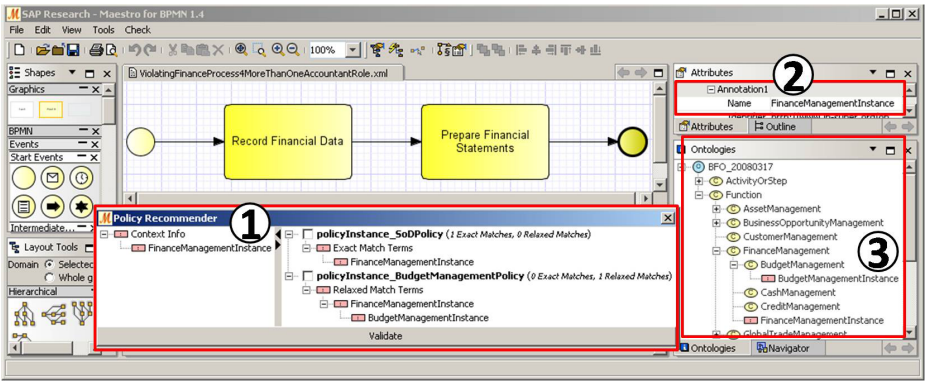

Fig. 2. Policy recommender application based on matchmaker

Matchmaker uses the context information of the currently modeled process (see no. 2 in Fig. 2), which is captured by annotating it with instances defined in the application ontologies. Each instance in the process context is looked for corresponding matching instances in the policy context. Ranking is done based on a number of instances in the process context with a corresponding matching instance in the policy context and the type of match. We distinguish two types of match, namely *Exact* and *Relax*. Exact match occurs if the same instance is used in process and policy contexts. Relaxed match occurs if an instance in policy context is either from the same concept, super concept, subconcept, or any ancestor concept in the concept hierarchy of an instance in process context (see no. 3 in Fig. 2 for concept hierarchy). Exact match is ranked higher compared to relaxed (see no. 1 in Fig. 2). Query mechanism provided by underlying formalization is used to perform the matchmaking.

[4] www.bpmn.org

Users can select from a list of business policies (see no. 1 in Fig. 2) for enforcement. The enforcement of all specified business policies against a process model is performed by considering them as knowledge base and checking for consistency of the knowledge base via logical reasoning using the *Integrated Rule Inference System* (IRIS) [8] inference engine. WSML2Reasoner[5] is used to transform ontologies in WSML to the underlying IRIS language. On violations, users are provided with natural language description of violated business rules in the *Problems* view (see no. 8 in Fig. 1).

4 Demo Scope

During the demonstration, we will first show how a user can assign the context information to an SoD policy. In the next step, the user will switch the perspective to *Maestro4BPR* in order to model the business rules which implement the policy. After each of the 3 implementing business rules are visually modeled, we show their translation to the underlying formal representation. Finally, for a given process model we search the relevant policies based on their context annotations, select the matched SoD policy and enforce it on the process.

5 Conclusion

We have presented extensions to *Maestro* which demonstrate tool support for modeling and enforcement of business policies and rules on process models. This enables the automation in ensuring compliance of organization's business processes with the established business policies and rules. We illutrate the tool support using an SoD business policy scenario.

References

1. `http://www.omg.org/spec/BMM/1.0/PDF` (2008)
2. Markovic, I., Hasibether, F., Jain, S., Stojanovic, N.: Process-oriented semantic business modeling. In: Wirtschaftsinformatik (2009)
3. Koehler, J., Vanhatalo, J.: Process anti-patterns: How to avoid the common traps of business process modeling. IBM WebSphere Dev. Tech. Journal 10(2), 4 (2007)
4. Congress of the United States: Public Company Accounting Reform and Investor Protection Act (Sarbanes-Oxley Act). Pub. l. no. 107-204, 116 stat. 745 edn. (2002)
5. Davis, R., Brabänder, E.: ARIS Design Platform. Springer, Heidelberg (2007)
6. Happel, H.J., Stojanovic, L.: Ontoprocess - a prototype for semantic business process verification using SWRL rules. In: Demosession of the 3rd (ESWC 2006) (June 2006)
7. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language WSML: An Overview. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 590–604. Springer, Heidelberg (2006)
8. Bishop, B., Fischer, F.: IRIS — Integrated Rule Inference System. In: ARea Workshop (2008)

[5] http://tools.sti-innsbruck.at/wsml2reasoner/