

A Static Business Level Verification Framework for Cross-Organizational Business Process Models using SWRL^{*}

Kioumars Namiri¹, Mani-Maurice Kügler², Nenad Stojanovic³

¹SAP Research Center CEC Karlsruhe, SAP AG, Vincenz-Prießnitz-Str.1, 76131
Karlsruhe, Germany
{Kioumars.Namiri,Mani-Maurice.Kuegler}@sap.com

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

Abstract: In this paper we introduce a solution for the problem of the design-time verification of cross-organizational business processes. The configuration of the processes can be checked in order to examine whether it fulfills a certain set of business level requirements. A formal Ontology in OWL for these kinds of processes is provided, so logic mechanisms for the verification are applied. The business level requirements are expressed in SWRL. A demonstrable implementation integrated in an already existing business process modeling tool has been developed.

1 Introduction

Today most software applications have to be adapted in such a way that the implemented business processes (BPs) meet the requirements of the customer enterprises. Usually, equal BPs differ from company to company as a result of different and changing business environments. At the same time, most enterprises depend on business partners in the market to perform some parts of the business process (BP). The resulting BPs, which cross two or more enterprises, are called Cross-organizational Business Processes (CBPs) [GLKZJ06]. In real world situations not fully configured ready to run CBPs are delivered by ERP vendors to customer enterprises, since the vendors aim to cover the requirements of all of their customers. The customer company is then forced to configure the set of available external/internal business functionalities provided by partners and internal systems in CBPs in such a way that it fulfills the enterprises situation. One challenging need in this context is to provide a mechanism to ensure that CBPs are configured and upgraded consistently.

* This material is based upon work partially supported by the EU funding under the project ATHENA (FP6 - 507312). This paper reflects the author's views and the Community is not liable for any use that may be made of the information contained therein.

The vision is a top-down deployment of business level requirements (BLRs) on an enterprise application spanning over different BPs and a bottom-up verification of already configured BPs making sure they steadily fulfill the BLRs. We have addressed this problem by providing a rule based verification mechanism to verify whether the design of a CBP fulfills a set of rules. We have integrated the implementation in the CBP modeling tool called Maestro, which was developed in context of the EU-Funded Research Project ATHENA [ATHENA04]. The rules are expressed by SWRL-statements, which are added to an ontological Knowledge Base (KB) through a UI, which should enable non-technical business experts to express rules.

2 Motivating Scenario for Verification of CBPs

We use the carrier-shipper-scenario to exemplify the problem space. Two different business situations for two different customer enterprises (shipper 1 and 2 in Figure 1) are described. They lead to different configurations of the same core-shipping services *calculate rate*, *generate routing code* and *generate label*. These services are provided as Web services by a carrier company and are integrated into the standard order-to-cash-process in each customer enterprise. This is a result of a real world analysis of different shipping companies regarding their interoperability requirements. In case of process variant 1, the *rate calculation* and the *routing code calculation* are done during *sales order* whereas *label generation* is done after the goods are packed. In process variant 2 the *routing code calculation*, *rate calculation* and *label generation* are all performed after the goods have been packed. We provide a mechanism that enables a business user to express and verify the business level call dependencies for each process variant in SWRL. Once the SWRL statements are added to the KB, each time the technical design/configuration of the process is customized/updated, the business user can use the verification mechanism to find out whether the technical configuration of the CBP still satisfies the previously expressed BLRs on the CBP.

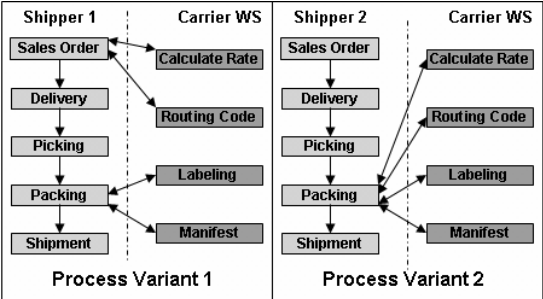


Figure 1: Two different CBP variants for the same Order-Process

3 Semantic Representation of CBPs

In the following we first give some notes on the nature of the verification problem before we further detail the Model Ontology developed.

3.1 Verification of CBPs

Verification of the BP configurations is realized using formal methods. These methods seek to establish a logical proof that a system works correctly, i.e. that it is correctly configured. A formal approach provides:

- a modeling language to describe the system;
- a specification language to describe the correctness requirements; and
- an analysis technique to verify that the system meets its specification.

The model describes the possible behavior of the system, and the specification describes the desired behavior of the system. To formally prove the correctness of a model, the first decision is about what claims to prove. In our case, the claim is that there is no violation of the requirements regarding binding of services. It means that the system for BP verification has to return an error value in the case that an activity does not comply with a predefined binding. That can be formally described as follows [NS06]:

$$\begin{aligned} \text{isCompliant}(X) &\leftarrow \neg \text{ErrorBinding}(X) \\ \text{ErrorBinding}(X) &\leftarrow \text{Activity}(X) \wedge \text{WebService}(Y) \wedge \text{Binding}(X, Y) \wedge \\ &\quad \neg M(\text{Property}(X), \text{Property}(Y)) \\ M: \text{Property}(A) &\rightarrow \text{Property}(WS)^n \end{aligned}$$

, where A is the set of activities from a BP, WS is a set of Web services, Property(x) is a function that retrieves characteristics of an entity x. A characteristic is defined according to the underlying process model.

3.2 OWL Model Ontology for CBPs

The model ontology defines all relevant classes, concepts, and relationships regarding CBPs. The main classes of the model ontology are described in the following:

The *BusinessProcess* class is related to nodes and edges by the properties *hasNode* and *hasEdge*, respectively. The class *Partner* stands for one business partner, who is linked to the *BusinessProcess* class by its property *hasPartner*. A *Node* represents any kind of node contained in a BP. By means of the properties *isPredecessorOf*, *isDirectPredecessorOf*, *isSuccessorOf*, and *isDirectSuccessorOf*, each individual of it is in some way linked to all other nodes within one BP. Its subclasses are *ActivityNode* (representing an activity in a BP), *SenderNode* and *ReceiverNode* (standing for a node that enables outgoing and incoming communication in a CBP, respectively), and *CoordinatorNode* (representing a node that aligns the execution of a BP). The class *Edge* links two individuals of the *Node* class with each other. An individual of the *Task* class refers to a task profile that can be attached to a node of a BP. The subclasses of this class are *UserTask*, *PrivateServiceTask*, *SenderServiceTask*, and *ReceiverServiceTask*.

4 Expression and verification of business level requirements

Such a BLR that can be captured could be, e.g., a constraint like which Web services have to be called in a certain activity of a BP.

Based on the carrier-shipper-scenario, the following example rule will be expressed: “After the sales order was created, the rate needs to be calculated.” Expressed in terms of literals and the vocabulary given by the model ontology described earlier, this rule could be defined as shown in Figure 2.

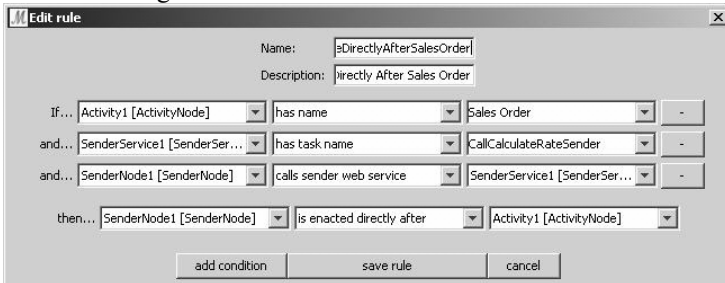


Figure 2: Add rule - dialog

The *Add rule* - dialog constrains the vocabulary that can be used for defining the rules through the combo boxes. Each literal of a rule consists of a predicate and a set of terms. The predicates that may be used for modeling the rules are confined to binary predicates, also called properties. The properties available in the *Add rule* - dialog reflect the properties defined in the model ontology. In order to make the *Add rule* - dialog more user friendly, it does not show the names of the properties as modeled in the ontology, but it displays a description of them that is easier to read for the user. The two terms that belong to every property are also called domain and range. Figure 3 identifies the domain, the property, and the range of an example literal displayed in the *Add rule* - dialog. The domain of a literal, with regard to the *Add Rule* - editor, is always some kind of variable or a string value.

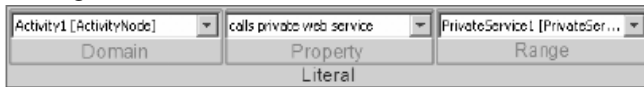


Figure 3: Structure of the *Add Rule* - dialog

In order to identify contradictions between the rules and the belonging BP, the individuals that are not rule compliant need to be identified by the ontological KB. Figure 4 shows a snapshot from the Maestro Process Modeling tool illustrating how the verification process is integrated visually into the Maestro tool. The snapshot shows that the red highlighted process step in the selected CBP violates the rule in the KB described in the Text area in the bottom part of the snapshot.

5 Overall Architecture and Implementation

The architecture of the prototypical verification tool consists of these three main parts:

- The Maestro application provides the UI for both designing the CBPs and the user friendly UI for expressing the SWRL statements.
- The KAON2 framework is used and is responsible for ontology and rules management and additionally for reasoning of the ontological KB, and
- the ontological KB containing the OWL ontologies and rules.

We have constrained the SWRL expressivity to DL-Safe rules [MSS04]. DL-Safe rules are sound and complete while still satisfying our requirements for representing the BLRs.

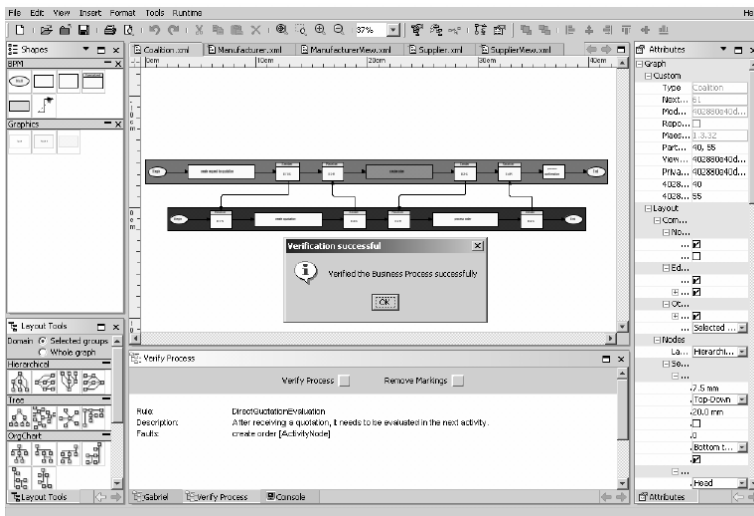


Figure 4: Graphical representation of the verification result in the Maestro tool

6 Conclusion and future research

In this paper we introduced a prototypical implementation of a SWRL based verification mechanism for the configuration of CBPs. The approach considers only the design time of BPs. Actually in the context of regulatory requirements such as Sarbanes Oxley [Sox02], the law requires that some rules/constraints are effective during execution time of BPs. In such cases a design time approach does not satisfy the requirements sufficiently. Thus, a next step would be to expand the approach so that it considers the runtime of BPs as well.

References

- [ATHENA04] ATHENA Consortium, December 2004, <http://www.athena-ip.org>, retrieved March 12th 2007.
- [GLKZJ06] Greiner, U.; Lippe, S.; Kahl, T.; Ziemann, J.; Jkel, F.W.: Designing and implementing crossorganizational business processes - description and application of a modeling framework. In: Interoperability for Enterprise Software and Applications Conference I-ESA, 2006.
- [MSS04] Motik, B.; Sattler, U.; Studer, R.: Query Answering for OWL-DL with Rules, Proc. of the 3rd Int. Semantic Web Conference (ISWC 2004), Hiroshima, Japan, November 2004.
- [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.
- [Sox02] Pub. L. 107-204. 116 Stat. 754, Sarbanes Oxley Act ,2002.