An Ontological Approach to Oracle BPM

Jean Prater, Ralf Mueller, and Bill Beauregard

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065, USA
{jean.prater,ralf.mueller,william.beauregard}@oracle.com

Abstract. A modern business process management (BPM) operates using common tenants of an underlying Service Oriented Architecture (SOA) runtime infrastructure based on the Service Component Architecture (SCA) and supports the BPMN 2.0 OMG[1] standard. Semantically-enabling all BPM artifacts, from high-level design to deployment and the runtime model of a BPM application, promotes continuous process refinement, comprehensive impact analysis, and reuse to minimize process and service proliferation. A semantic database can manage semantically-enabled BPM ontologies and models, enable machine-driven inference to discover implicit relationships in the models, and perform pattern-matching queries to find associations.

This paper presents an ontology for BPM based upon BPMN 2.0, Service Component Architecture (SCA) and the Web Ontology Language (OWL 2) that can support a wide range of use cases for process analysis, governance, business intelligence and systems management. It has the potential to bring together stakeholders across an enterprise, for a truly agile, end-to-end enterprise architecture.

Keywords: BPM Ontology, OWL 2, BPMN 2.0, SOA, SCA, SPARQL, Semantic Technologies.

1 Introduction

Semantic technology can be used by business process management (BPM) to define contextual relationships between business processes. This allows 'software agents' (programs working on behalf of people) to automatically find the right information or processes and enables machine-driven decision-making based on the established contextual relationships.

Organizations can optimize their information technology resources through a Service Oriented Architecture (SOA) approach that embraces common business processes and semantics throughout the enterprise. The challenge, however, with applications built on BPM and SOA is the number of representation formats used to define artifacts, such as the Service Component Architecture Assembly Model, Web Service WSDL definitions, and XSLT transformations. This makes even simple queries about the entire BPM model difficult and complicated.

[1] Object Management Group, see `http://www.omg.org`

J.Z. Pan et al. (Eds.): JIST 2011, LNCS 7185, pp. 402–410, 2012.

This heterogeneous modeling problem can be addressed with an ontology based upon the W3C OWL standard [7] that encompasses all artifacts of a BPM application. The ontology and associated models are stored in a semantically-enabled database. Support for W3C SPARQL query language allows pattern matching queries across the enterprise-wide model to find associations. Native inferencing in the database allows machine-driven discovery of new, previously undefined relationships within the model.

Business users can find, share, and combine information and processes across organizational boundaries more easily with the addition of contextual relationships. The combination of business process management and semantic technology driven by an underlying ontology makes it possible to:

- Evolve a business process model into a complete executable process in the same model.
- Analyze the impact on existing processes and web services of adding, modifying or deleting processes and process building blocks.
- Minimize proliferation of processes and services.

Oracle is actively involved in the standards process and is leading industry efforts to use ontologies for metadata analysis. Oracle is also investigating the integration of organizational and social aspects of BPM using the ontology FOAF[2]. BPMN 2.0 task performers can be associated with a FOAF Person, Group or Organization and then used in Social Web activities to enable business users to collaborate on BPM models.

2 Customer Use Case

The US Department of Defense (DoD), Business Transformation Agency (BTA), under the purview of the Deputy Chief Management Officer (DCMO) is on a mission to achieve an Architecture-driven Business Operations Transformation. A key principle in the DoD Business Transformation is developing a common vocabulary and standard processes in support of business enterprise interoperability. The use of primitives and reuse of process patterns will reduce overhead costs, process duplication and resources needed for building and maintaining enterprise architectures. By aligning the Department of Defense Architecture Framework[3] 2.0 (DoDAF 2.0) with Business Process Modeling Notation 2.0 (BPMN 2.0) and partnering with industry, the BTA is accelerating the adoption of these standards to improve government business process efficiency.

A vital tenet for success is ensuring that business process models are based on a standardized, semantically-enabled representation. This will enable analysis and comparison of end-to-end business processes and reuse of the most efficient and effective process patterns (style guide) and elements (primitives), throughout the DoD Business Mission Area.

[2] The Friend of a Friend (FOAF) project, see `http://www.foaf-project.org`

[3] See `http://www.bta.mil/products/BEA_6.2/BEA/products/2009-04-27 Primitives Guidelines for Business Process Models (DoDAF OV-6c).pdf`

The DoD implementation used Oracle BPM and the ontology for BPM discussed in this paper to generate RDF triples from the artifacts of the BPM Project. These artifacts included BPMN 2.0 process definitions, SCA assembly model, WSDL service definitions, XML-Schema and other metadata. The triples were managed in Oracle Database using the Semantic Technologies feature of Oracle Spatial [3] and a SPARQL endpoint was used to query the model.

3 An Ontology for BPM

An ontology for BPM encompasses and expands the BPMN 2.0 Ontology and Service Component Architecture (SCA) ontology. This ontology creates a composite model by establishing relationships between the OWL classes of the BPMN 2.0 ontology and the OWL classes of the SCA runtime ontology. The following diagram illustrates the layering of the BPM ontology on top of the BPMN 2.0 and SCA ontology and their relationship to other representation formats.

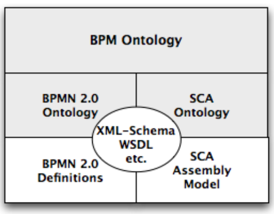

Fig. 1. The BPM ontology stack

Generally speaking, the ontology for BPM relates BPMN 2.0 tasks and events to corresponding SCA composite model entities like components, services and references. For instance, BPMN 2.0 Process, User and Business Rule tasks are related to components in the SCA composite model and BPMN 2.0 Send, Receive and Service tasks and Message Events are related to SCA Services and References. The BPM ontology creates appropriate relationships between these composite model artifacts. The relationships include the OWL representation of SCA wires. Figure 2 provides an example that illustrates the anatomy of a Business Rule Task "`Determine Approval Flow`".

One can see in Figure 2 how the BPM ontology relates BPMN 2.0 tasks to corresponding SCA composite model, services and references. At the top of the figure the Business Rule Task "`Determine_Approval_Flow`" is of type BPMN 2.0 "`BusinessRuleTask`". It is implemented by a SCA component "`ApprovalRules`" that is of SCA type "`DecisionComponent`" and related to another SCA Component "`RequestQuote`" through a SCA Wire.

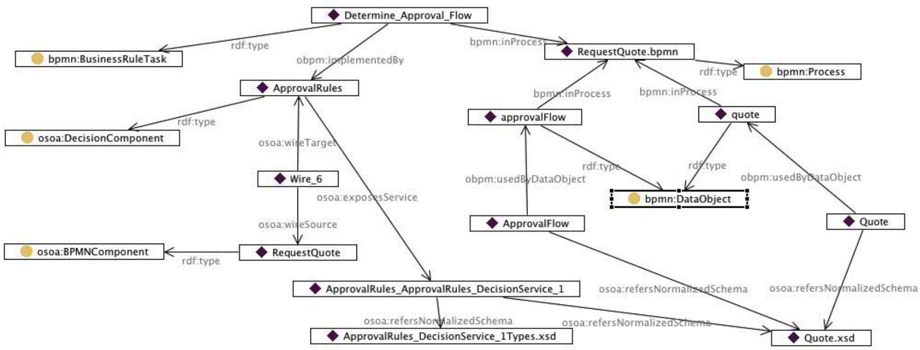

Fig. 2. Anatomy of a BPMN 2.0 Business Rule Task[4]

It is important to note that the SCA "`DecisionComponent`" in the middle left side of the figure exposes the Service in the bottom of the figure, "`…AprovalRules_DecisionService_1`" that in turn refers to the XML-Schema, "`Quote.XSD`" in the bottom right side of the figure. "`Quote.XSD`" is also referred to by data objects "`approvalFlow`" and "`quote`" in the BPMN 2.0 process "`RequestQuote.bpmn`" in the upper right side of the figure. This illustrates how the semantic relationships defined in the BPM ontology make it possible to perform a comprehensive impact analysis for process data objects, XML schema definitions and service contracts across the entire model.

4 An Ontology for BPMN 2.0

The OMG BPMN 2.0 standard specifies a serialization format for the semantic model of BPMN 2.0 processes in XMI and XML-Schema. The ontology for BPMN 2.0 is based on this model. Oracle BPM includes an implementation that comprises the following:[5]

OWL Classes and Properties for All BPMN 2.0 Elements That are Relevant for the Business Process Model.[6] The OWL classes, whenever possible, follow the conventions in the BPMN 2.0 UML meta-model. OWL properties and restrictions are included by adding all of the data and object properties according to the attributes and class associations in the BPMN 2.0 model.[7]

---

[4] Visualized using TopBraid Composer™.

[5] All of the classes of the BPMN 2.0 meta model that exist for technical reasons only (model m:n relationship or special containments) are not represented in the ontology.

[6] The work in [2] describes an ontology based on BPMN 1.x for which no standardized meta model exists.

[7] Oracle formulated SPARQL queries for envisioned use cases and added additional properties and restrictions to the ontology to support those use cases.

OWL Classes and Properties for Instantiations of a BPMN 2.0 Process Model. These OWL classes cover the runtime aspects of a BPMN 2.0 process when executed by a process engine. The process engine creates BPMN 2.0 flow element instances when the process is executed. Activity logging information is captured, including timestamps for a flow element instance's activation and completion, as well as the performer of the task.

The implicit (unstated) relationships in the ontology for BPM can be automatically discovered using semantic inferencing, for example as provided by the native inferencing engine included with Oracle Database Semantic Technologies. The explicit and implicit relationships in the ontology can be queried using SPARQL pattern matching queries and, in the case of Oracle Database, can also be queried using SPARQL patterns in SQL queries. [5] Example SPARQL queries are shown below:

Select all User Tasks in all Lanes

```
select ?usertask ?lane
 where {
   ?usertask rdf:type bpmn:UserTask .
   ?usertask bpmn:inLane ?lane
 }
```

Select all flow elements with their sequence flow in lane p1:MyLane (a concrete instance of RDF type bpmn:Lane)

```
select ?source ?target
 where {
   ?flow bpmn:sourceFlowElement ?source .
   ?flow bpmn:targetFlowElement ?target .
   ?target bpmn:inLane p1:MyLane
 }
```

Select all activities in process p1:MyProcess that satisfy service level agreement (SLA) p1:MySLA

```
select ?activity ?activityInstance
 where {
   ?activity bpmn:inProcess p1:MyProcess .
   ?activityInstance obpm:instanceOf ?activity .
   ?activityInstance obpm:meetSLA p1:MySLA
 }
```

Representative examples[8] of the BPMN 2.0 ontology elements in OWL functional syntax[9] are listed below.[10]

[8] The complete BPMN 2.0 ontology comprises of about 100 OWL classes.

[9] OWL 2 Structural Spec. and Functional-Style Syntax, `http://www.w3.org/TR/owl2-syntax`

Start Event:

```
Declaration(Class(bpmn:StartEvent))
SubClassOf(bpmn:StartEvent bpmn:CatchEvent)
SubClassOf(bpmn:StartEvent DataMaxCardinality(1
bpmn:isInterrupting))
```

Data Association:

```
Declaration(Class(bpmn:DataAssociation))
SubClassOf(bpmn:DataAssociation bpmn:BaseElement)
SubClassOf(bpmn:DataAssociation DataMaxCardinality(1
bpmn:hasAssignment))
SubClassOf(bpmn:DataAssociation DataMaxCardinality(1
bpmn:dataAssociationSource))
SubClassOf(bpmn:DataAssociation DataMaxCardinality(1
bpmn:dataAssociationTarget))
SubClassOf(bpmn:DataAssociation DataMaxCardinality(1
bpmn:hasTransformation))
```

Task:

```
Declaration(Class(bpmn:Task))
SubClassOf(bpmn:Task bpmn:Activity)
DisjointClasses(bpmn:Task bpmn:CallActivity)
DisjointClasses(bpmn:Task bpmn:SubProcess)
```

ItemAwareElement:

```
Declaration(Class(bpmn:ItemAwareElement))
SubClassOf(bpmn:ItemAwareElement bpmn:BaseElement)
SubClassOf(bpmn:ItemAwareElement DataMaxCardinality(1
bpmn:hasItemDefinition))
```

5 An Ontology for SCA

The SCA composite model ontology represents the SCA assembly model. The Oracle BPM implementation comprises OWL classes for Composite, Component, Service, Reference and Wire, which form the major building blocks of the assembly model. A SCA ontology can be specified with OWL classes for concrete services specified by WSDL and data structures specified by XML-Schema. The transformation of the SCA assembly model to the SCA ontology includes creating finer grained WSDL and XML-Schema artifacts to capture the dependencies and relationships between concrete WSDL operations and messages to elements of some XML-Schema and their imported schemata.

[10] Generated from the source OWL via `http://owl.cs.manchester.ac.uk/converter/`

The SCA ontology is primarily created for the purpose of governance and to act as a bridge between the ontology for BPM and an ontology that would represent a concrete runtime infrastructure. This enables the important ability to perform impact analysis, identify dependencies and prevent unnecessary proliferation.

6 Applications

The ontology for BPM enables a rich set of applications based on a common model.

6.1 Continuous Process Refinement and Optimization

In modern enterprises, business processes are continuously refined and optimized for better business performance and to adapt to market changes. Reading from the bottom, the diagram in Figure 3 visualizes this refinement and optimization loop.

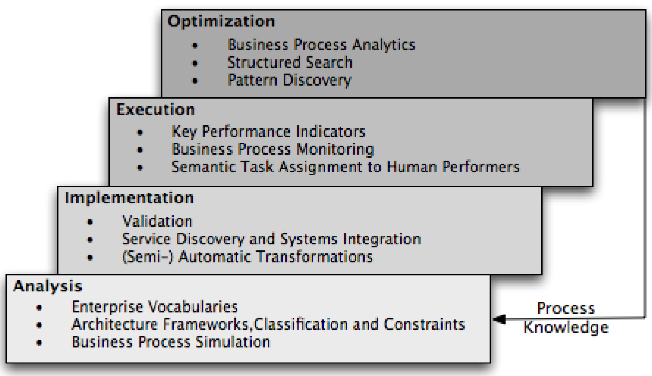

Fig. 3. Applications of Ontology for BPM

6.1.1 Process Analysis

In the process analysis stage, the ontology for BPM allows the process modeling toolset to constrain the palette of modeling elements depending on a chosen architecture framework (like DoDAF), and process tasks and data objects associated with concepts of the given enterprise vocabulary. The ontology can also help identify the people in the organization responsible for a specific task.

The ontology associates custom classification models, thesauri, and people's skill sets with process models and tasks, fosters re-use of existing tasks, constrains process models to appropriate modeling elements, and establishes a relationship between the process models and the broader concepts of the enterprise. It also enables impact and dependencies analysis for all artifacts (data, services, tasks, and people) involved in a business process. This can reduce the proliferation of services and processes and the risk of disrupting the business by introducing incompatible service contracts.

6.1.2 Implementation

An important aspect of implementing a business process for execution is validating the correctness of the process relative to the process blueprint from the analysis stage.

The ontology for BPM provides relationship metadata that the business process modeling toolset can use to validate semantic equivalence between the process model defined in the analysis stage and the process engine's execution. Validation includes ensuring the process and data flow conform with the organization's enterprise architectures.

6.1.3 Execution

During execution of a business process, the process engine generates events that can be used for business activity monitoring and identifies the human resources for the User Tasks in the business process.

The ontology for BPM can be used to provide a more precise identification of a human performer for the user tasks in the business process by relating specific attributes about an organization and skill sets. In addition, the events from activity monitoring can be investigated in the broader context by relating relevant information about business strategies and service level agreements.

6.1.4 Optimization

Optimization of a business process requires a holistic view of the process including the systems, services and people involved in the execution of business process.

The ontology for BPM can be referenced for semantically-enabled searches that provide a more complete perspective of the artifacts involved in a business process and discover patterns in a business process to foster re-use, reduce redundant work and identify bottlenecks.

6.2 Systems Integration

Modern business applications can expose thousands of SOA Services that have the potential to be re-used and integrated in custom business processes.

The ontology for BPM can be used as a source to search for services or composite applications that might be used in the implementation of a business process task. It can further support the developer in the modeling of data transformations by relating structural and semantic attributes of the source and target data objects. It is quite normal for the services exposed by commercial business applications to define hundreds of attributes for the service input and output. Semantic technology can help the systems integrator automate the process of mapping those attributes, a task that is time consuming and error prone if done manually.

7 Conclusion

An ontology for BPM, such as the one implemented for Oracle BPM, encompasses and expands the generic ontologies for BPMN 2.0 and the SOA composite model. It covers all artifacts of a BPM application from a potentially underspecified[11] process model in BPMN 2.0 down to the XML-Schema type level. The combination of BPM with RDF/OWL data storage, inferencing and SPARQL querying, as supported by Oracle Database Semantic Technologies, provides the ability to discover implicit relationships in the BPM models and find patterns beyond what is possible with the classical approaches of XML-Schema, XQuery and SQL. This promotes continuous process refinement, comprehensive impact analysis, and reuse to minimize unnecessary proliferation of processes and services.

Acknowledgements. We'd like to thank Sudeer Bhoja, Linus Chow, Xavier Lopez, Bhagat Nainani and Zhe Wu for their contributions to the paper and valuable comments.

References

[1] Business Process Model and Notation (BPMN) Version 2.0, `http://www.omg.org/spec/BPMN/2.0/`

[2] Ghidini, C., Rospocher M., Serafini L.: BPMN Ontology, `https://dkm.fbk.eu/index.php/BPMN_Ontology`

[3] Oracle Database Semantic Technologies, `http://www.oracle.com/technetwork/database/options/semantic- tech/`

[4] Service Component Architecture (SCA), `http://www.osoa.org`

[5] Kolovski, V., Wu, Z., Eadon, G.: Optimizing Enterprise-Scale OWL 2 RL Reasoning in a Relational Database System. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 436–452. Springer, Heidelberg (2010)

[6] "Primitives and Style: A Common Vocabulary for BPM across the Enterprise"; Dennis Wisnosky, Chief Architect & CTO ODCMO and Linus Chow Oracle; BPM Excellence in Practice 2010; Published by Future Strategies (2010)

[7] Web Ontology Language (OWL), `http://www.w3.org/owl`

[11] A BPMN 2.0 model element is considered underspecified, if its valid but not all attribute values relevant for execution are specified.