

WSMX: A Solution for B2B Mediation and Discovery Scenarios^{*}

Maciej Zaremba¹ and Tomas Vitvar²

¹ Digital Enterprise Research Institute (DERI),
National University of Ireland, Galway
maciej.zaremba@deri.org

² Semantic Technology Institute (STI2),
University of Innsbruck, Austria
tomas.vitvar@sti2.at

Abstract. We demonstrate Web Service Execution Environment (WSMX), a semantic middleware platform for runtime service discovery, mediation and execution, applied to SWS Challenge scenarios. We show the modelling principles as well as execution aspects of the WSMX semantic technology addressing the real-world requirements.

1 Introduction

Semantic Web Services (SWS) technologies offer promising potential to enable integration and discovery that is more flexible and adaptive to changes occurring over a software system's lifetime. However, there remains very few publicly available, realistic, implemented scenarios that showcase the benefits of semantics for services. In this respect we develop the WSMX¹ – a middleware system that operates on semantic description of services and facilitates automation in service integration and discovery. We demonstrate the value of the WSMX on real-world scenarios from SWS Challenge², a community-driven initiative that provides a set of scenarios with real Web services along with a methodology for the evaluation of solutions. We contribute to the SWS Challenge by implementing solutions based on the WSMX showing how this technology can be used to facilitate dynamic discovery and mediation in B2B integration. Users of our demonstration are able to learn how existing, non-semantic Web services can be semantically enabled and what are the benefits of semantics in the context of B2B integration and service discovery.

2 SWS Challenge Scenarios and WSMX

We build our solutions on the SWS framework developed in DERI including conceptual model for SWS (Web Service Modeling Ontology, WSMO[2]), language

^{*} This work is supported by the Science Foundation Ireland Grant No. SFI/02/CE1/I131, and the EU projects SUPER (FP6-026850), and SemanticGov (FP-027517).

¹ WSMX is an open-source project, see <http://sourceforge.net/projects/wsmx>

² <http://www.sws-challenge.org>

for service modeling (Web Service Modeling Language, WSML[2]), middleware system (Web Service Execution Environment, WSMX[4]), and modelling framework (Web Service Modelling Toolkit, WSMT³). In order to model the scenario, we use WSMO for modeling of services and goals (i.e. required and offered capabilities) as well as ontologies (i.e. information models on which services and goals are defined) all expressed in the WSML-Flight ontology language. WSML-Flight provides a Datalog expressivity extended with inequality and stratified negation that is sufficient for addressing requirements of SWS Challenge scenarios. We use KAON2 reasoner⁴ for the inference over WSML-Flight ontologies. In addition, we use Java SWING⁵ based monitoring facility to display WSMX components' progress of use-case executions.

Figure 1 depicts solution architectures for SWS Challenge discovery (part A) and mediation (part B) scenarios. **Discovery** (A) defines tasks for identifying and locating Business Services. **Selection** (A) selects most appropriate service according to user's preferences. **Orchestration** (A and B) executes the composite business process. **Mediation** (B) resolves data and process heterogeneities. **Reasoning** (A, B) performs logical reasoning over semantic descriptions of services.

2.1 Mediation Scenario

The mediation scenario (Figure 1, part B) describes a data and process mediation of a trading company called Moon. Moon uses two back-end systems to manage its order processing, namely a Customer Relationship Management system (CRM) and an Order Management System (OMS). The SWS Challenge provides access to both of these systems through public Web services. The scenario describes how Moon interacts with its partner company called Blue using RosettaNet PIP 3A4 purchase order specification⁶. Using the WSMT data mapping tool we map the Blue RosettaNet PIP 3A4 message to messages of the Moon back-end systems. We then apply the WSMX data and process mediation components to resolve incompatibilities of message exchanges defined by the RosettaNet PIP 3A4 process and those defined in the Moon back-end systems.

Our major contributions to the mediation scenario shows:

- how flat XML schema of RosettaNet purchase-order and other proprietary messaging schema used by different partners could be semantically enriched using the WSML ontology language as Listing 1.1 shows,
- how services provided by partners could be semantically described as WSMO services and built on top of existing systems,
- how conversation between partners and their services can be facilitated by the WSMX integration middleware enabling semantic integration, and

³ <http://sourceforge.net/projects/wsmt>

⁴ <http://kaon2.semanticweb.org>

⁵ <http://java.sun.com/docs/books/tutorial/uiswing>

⁶ RosettaNet is the B2B integration standard and PIP (Partner Interface Process) define various interactions patterns and vocabularies for business integration.

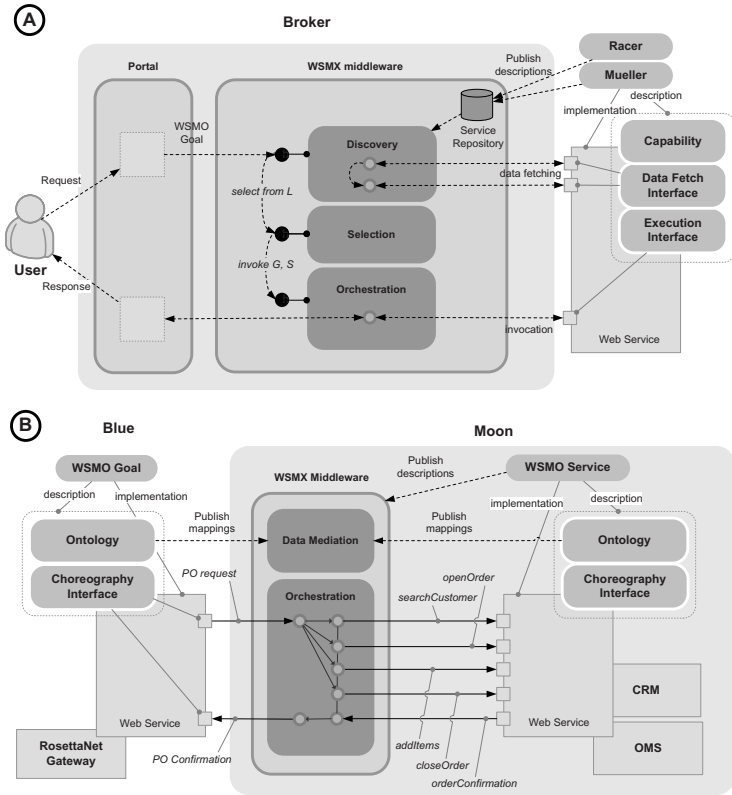


Fig. 1. Solution Architectures for SWS Challenge Scenarios

- how data and process mediation can be applied between heterogeneous services within the integration process.

```

1  /* XSLT Extract of lifting rules from XML message to WSML */
2  instance PurchaseOrderUID memberOf por#purchaseOrder
3  por#globalPurchaseOrderTypeCode hasValue " <xsl:value-of select=" dict:GlobalCode" />"
4  por#isDropShip hasValue IsDropShipPo
5  <xsl:for-each select=" po:ProductLinItem" >
6    por#productLinItem hasValue ProductLinItem<xsl:value-of select=" position()" />
7  </xsl:for-each>
8  ...
9  /* message in WSML after transformation */
10 instance PurchaseOrderUID memberOf por#purchaseOrder
11 por#globalPurchaseOrderTypeCode hasValue "Packaged product"
12 por#isDropShip hasValue IsDropShipPo
13 ...

```

Listing 1.1. Lifting in XSLT and resulting WSML message

Since the core WSMX functionality operates on semantic descriptions of messages, WSMX needs to also facilitate transformations between semantic and non-semantic messages through so called grounding descriptions (i.e. lifting and

lowering). For the modelling phase we demonstrate how WSMT toolkit can be applied to modeling of both semantic and grounding definitions for the Moon and Blue companies, that is, how we model service orchestrations, domain ontologies, lifting/lowering groundings. Data Mediation is based on declarative, rule-based mappings between source and target ontologies. We show how these mappings are created during the design-time and how they are executed on the messages (instance level) using KAON2 reasoner during the execution time. We demonstrate details of semantic B2B integration focusing on types of data and process heterogeneities that WSMX is able to handle. We also show how changes in utilized ontologies and processes are handled. In case of source or target ontology changes, adjustments of their declarative mappings using WSMT are required. Changes on the service public process level are handled automatically by Process Mediator without any additional design time steps.

2.2 Discovery Scenario

The discovery scenario (Figure 1, part A) describes a user who uses a third-party company (broker or e-hub) in order to buy certain products with shipment to certain location. A number of shippers allow to ship products with different shipment conditions (places of shipment, price, etc.). Our approach to discovery is to match a WSMO Goal with a WSMO Web service through their semantic descriptions as well as to use additional data not available in the semantic descriptions (e.g., shipment price). The WSMX fetches this information during runtime through a specific Web service data-fetching interface. In [3] we define a conceptual framework supporting integration of dynamically fetched data into the discovery context.

```

1  /* general abstract definition of the axiom in the common ontology */
2  relation isShipped(ofType sop#ShipmentOrderReq)
3
4  /* specification of the axiom in the Mueller ontology */
5  axiom isShippedDef definedBy
6    ?sOrder[sop#to hasValue ?temp, sop#package hasValue ?p] memberOf sop#SOrderReq and
7    ?temp[so#address hasValue ?to] and ?to[so#city hasValue ?city] and
8    isShippedContinents(?city, so#Europe, so#Asia, so#NorthAmerica, so#Africa) and
9    ( (?p [so#weight hasValue ?weight] memberOf so#Package) and (?weight =< 50) )
10 implies
11   sop#isShipped(?sOrder).

```

Listing 1.2. isShipped relation declared in the common and Mueller ontologies

We semantically describe shipment capabilities offered by different companies using common shipment ontology. We take the advantage of the shared ontology when defining “abstract” axioms and their specialization in the concrete shipment service ontology (e.g., *isShipped* axiom as Listing 1.2 shows). The axiom is shared by both the shipping services and the goals (representing service requester) and provides an interface-like mechanism⁷ to define a common

⁷ An analogy are interfaces in programming languages like Java. The interface declares some functionality but does not say how this should be implemented.

evaluation criteria for service discovery. Requestor does not need to know how *isShipped* is specified by the service, but it can use it in its request to check whether given service is able to ship for a specified input (i.e., source and target location, package weight, dimension, etc.). There is a context Knowledge Base (KB) created for every instance of matchmaking between goal and a service. Logic query provided in the goal is evaluated against this KB by KAON2 reasoner and depending on the evaluation and variable binding, a service is assigned to the given matching category and is further ranked for runtime selection purposes.

Our solution demonstrates how domain ontologies, shipment goals and services are semantically described (concepts, instances, relationships, rules) as well as how service discovery works. We demonstrate Web services and Goals modelling principles and we show how extra information (e.g., shipping price) can be dynamically provided into the discovery context by utilizing data-fetching service interface.

3 Related Work

Our work can be compared to other solutions of SWS Challenge scenarios like Diane or SWE-ET as described in [1]. Diane solution provides a solution based on the language supporting fuzzy sets modelling, although without rule support. On the other hand, SWE-ET is based on combination of software modelling workflow-based methods with Flora-2 utilized for semantic descriptions.

4 Conclusion

With our contribution to the SWS Challenge we proved the value of the WSMX semantic technology in the context of B2B integration. Our solutions have been evaluated, by peer-review, according to the evaluation methodology of the SWS Challenge⁸. The evaluation criteria targets the adaptivity of the solutions, that is, solutions should handle introduced changes by modification of declarative descriptions rather than code-changes. Success level 0 indicates a minimal satisfiability level, where messages between middleware and back-end systems are properly exchanged. Success level 1 is assigned when changes introduced in the scenario require code changes and recompilation. Success level 2 indicates that introduced changes did not entail any code modifications and only declarative parts had to be changed. Finally, success level 3 is assigned when the system is able to automatically adapt to the new conditions. WSMX proved to deliver a generic solution⁹ scoring level 2 as there were no changes required in WSMX code when addressing new scenarios but it sufficed to adapt or provide a new semantic descriptions of involved services and service requestors.

⁸ http://sws-challenge.org/wiki/index.php/SWS_Challenge_Levels

⁹ http://sws-challenge.org/wiki/index.php/Workshop_Innsbruck

References

1. Kuster, U., et al.: Service Discovery with SWE-ET and DIANE - A Comparative Evaluation By Means of Solutions to a Common Scenario. In: 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira-Portugal (June 2007)
2. Roman, D., et al.: Web Service Modeling Ontology. *Applied Ontologies* 1(1), 77–106 (2005)
3. Vitvar, T., Zaremba, M., Moran, M.: Dynamic Service Discovery through Meta-Interactions with Service Providers. In: Franconi, E., Kifer, M., May, W. (eds.) *ESWC 2007*. LNCS, vol. 4519, Springer, Heidelberg (2007)
4. Vitvar, T., et al.: Semantically-enabled service oriented architecture: Concepts, technology and application. *Service Oriented Computing and Applications* 1(2) (2007)