Innovative Solution to Service Discovery and Integration using WSMOLX

Innovative Solution to Service Discovery and Integration using WSMOLX *

Maciej Zaremba[1], Raluca Zaharia[1], and Tomas Vitvar[2]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
firstname.lastname@deri.org
[2] Semantic Technology Institute, University of Innsbruck, Austria
tomas.vitvar@sti2.at

Abstract WSMX is a Semantic Web services broker enabling dynamic discovery and agile B2B integration of enterprise services. Our framework enables on-demand service discovery encompassing frequently changing aspects like fine-grained client's search requests and intricacies of service functionality. WSMX has a strong notion of mediation that overcomes heterogeneities on the data and protocol level, allowing services to be seamlessly executed. The main premise of using WSMX from a business perspective is its problem-oriented approach; tasks are specified at design time in terms of "what", not "how", and the business partners providing the optimal solutions are automatically discovered, selected and used at execution time.

1 Business Value Proposition

Over the last years we have observed a considerable shift from monolithic and proprietary IT systems built to last towards Service Oriented Architectures (SOA) built to change. Web services ease the integration of business functionality across a network, but there is still significant human labour required to discover and execute Web services. Semantics and ontologies are seen as key enablers of further automation in the SOA context, helping service requesters and providers to explicitly describe their needs or offered functionality. Ontologies provide the foundation for semi-automatic data and process mediation, simplifying service integration and execution. Logic reasoning over semantic descriptions achieves much higher precision in the discovery process.

We propose the WSMX[5] semantic middleware system as a solution that facilitates automation in service discovery and integration. WSMX is an open-source implementation of the Semantic Web services framework WSMO (Web Service Modeling Ontology [3]) and all the elements it operates upon are unambiguously described in WSML (Web Service Modeling Language [3]).

The WSMX framework materializes two of the major benefits envisioned by applying SOA in the business domain: (1) to respond quickly and cost-effectively to market changes and (2) to simplify reuse and interconnection of existing functionality by providing a uniform access manner. The first advantage means to be able to identify new providers for the required functionality or to easily switch

* *This work is supported by the Science Foundation Ireland Grant No. SFI/02/CE1/I131, and the EU project SUPER (FP6-026850).*

service providers when necessary. The discovery mechanism of WSMX finds the best business opportunities out of a large number of choices, while the selection ranks the matching services according to the user's input and the capabilities of the services. The second benefit means information exchange is achieved even between heterogeneous domain understandings. In general two systems are able to (automatically) communicate if they use the same format to transfer the information, understand the exchanged data and receive the messages that they expect (i.e. in a certain order). WSMX supports integration of the functionality provided by different systems through its mediation mechanism, which handles heterogeneities at both the data and process level.

The users are required to provide their requests or offerings in semantic descriptions, but the gained value of our approach consists of: (1) data understanding semantics (i.e. machine and human understandable data) that is much easier to use for non-technical users, (2) shortened time-to-market and business process adaptivity through flexibility in switching business partners whenever new opportunities arise where new products (services) can be easily built or extended, increasing reusability of existing systems, (3) seamless interoperability (providers register service descriptions with WSMX and clients use WSMX to handle discovery and interactions with the registered services). The registered services can be connected and exposed as a new added-value service since WSMOLX supports service orchestration on the semantic, declarative level.

In the following sections we will look at existing solutions, both semantic and non-semantic approaches to service integration and discovery, we will present our solution and two particular use cases where it has been successfully applied. Finally, we will discuss shortly projects that use and develop our framework and present an evaluation in context of the SWS-Challenge[3].

2 Position on the Market

Currently it is a time-consuming process to adapt IT solutions to the requirements arising from new business opportunities and service time-to-market is rather high. The typical IT system of a small or medium enterprise consists of monolithic legacy subsystems which are often insufficiently documented. Such a system addresses the requirements of the time when it was built, but is not able to effectively respond to changing demands and a time-consuming, manual process is applied to adjust the system to the new requirements. It is estimated that about 40% of companies' IT budget is allocated to the system integration[4], therefore further automation of this process is highly desirable.

SOA emerged from the companies' need for integration and lack of agility in existing IT systems. SOA promotes loose coupling and defines standardized means of communication, facilitating faster integration and reuse of the available functionality. Services can be used in contexts that have not been thought of when created, but they have particular data structures and message exchange

[3] `http://sws-challenge.org`
[4] Gartner report: `http://www.gartner.com`

patterns, resulting in significant human labour involved in connecting services. Current solutions consist of point-to-point system integration where both data and process mismatches are reconciled by a central mediation process. Software integration solutions like IBM WebSphere[5] or Microsoft BizTalk[6] tightly couple data and process mediation, whereas our approach provides separate solutions to data and process mediation using semantic descriptions. Separation of concerns gains N:M integration in case of WSMOLX, instead of 1:1 in case of tightly coupled solutions. Semantic SOA uses ontology based descriptions understandable both by machines and by humans for describing services and goals to be achieved. WSMX is an example of Semantic SOA providing number of platform services facilitating complex tasks of service discovery, selection, mediation and execution. Semantic SOA makes businesses more responsive to changing requirements and arising opportunities by facilitating service reuse and integration. Some business opportunities arise only for a short time and are gone or become insufficiently profitable if not addressed quickly enough.

In our solution, services complying with existing integration standards like RosettaNet are semantically augmented and therefore can be used with non-RosettaNet services as well. Once the provider's Web service has been identified, the WSMOLX solution can automatically handle aspects of the integration, reducing the time and human labour required before. One of the main advantages of our integration approach is *strong partner de-coupling*: when changes occur in the back-end system of one partner, consequent updates in the service descriptions do not affect the integration, which automatically adapts to the updated service descriptions as there is no central integration workflow. Manual effort is required in updating the semantic descriptions (i.e. ontologies and mapping rules) to reflect the changes in the back-end system, but our SWS technology allows for semi-automated approaches in modelling and mapping definitions, where a human user must adjust and approve the results.

There is currently no widely accepted approach for describing and discovering services. Approaches built around UDDI and ebXML have not gained substantial support on the market and publicly available UDDI repositories have been quietly shut down around year 2006 [2]. UDDI and ebXML repositories provide keyword based facilities for service description and discovery, supporting only coarse-grained, category level search and lacking any mechanism to determine details of service functionality for the customer at hand. Some of the relevant service information cannot be placed into a static service description and stored in some repository as expected by UDDI, ebXML and most of semantic approaches [1], as they are often customer dependent and may frequently change. WSMOLX provides a dynamic discovery mechanism that evaluates service suitability considering the *user's input* and *current service capability*. Changes to the fine grained parameters of a service search (e.g. service pricing) can result in different sets of matching services. These fine grained parameters are usually missing from the static part of service descriptions, meaning that service discov-

[5] `http://www.ibm.com/websphere`
[6] `http://www.microsoft.com/biztalk`

ery often remains a manual design-time process, while it could be automated in many cases.

For example, a service request "order 10 lbs package shipping from Galway to London in three days, overall cost below 30 Euro" usually has to be abstracted to a search for shipment services. Once such a service is located, it must be determined if the service is actually able to ship the given package considering the time, price and location requirements. It is clearly unfeasible to list all combinations of source and target locations with prices and delivery time in the service description. The WSMOLX approach can automate evaluation of such a request by checking the actual price of the service during the discovery process and providing service ranking following the specified criteria (e.g. the sooner the better). Price or delivery time information has a dynamic character and therefore can only be fetched from the service when needed. Information like the trading history with the customer, current service capability (e.g. workload, discounts), capability of a third-party company (e.g. lead time of contracted manufactures) influences the actual offering of the service. WSMOLX *operates and ranks actual service offerings* instead of generic, static and high level service description considered by most of the discovery approaches.

Feature	WebSphere/Biztalk UDDI/ebXML	WSMOLX
Service descriptions	Keywords, WSDL, UDDI - white, yellow, green pages,	Semantic capability, service data-fetching interface
Search (goal) description	Keywords	Explicit semantic descriptions
Data model	XML Schema	Ontologies created from analysing domain, services and messages
Discovery	Keyword matching	Rule evaluation - Logic Programming; hard and soft constraints
Service integration	Central workflow both for data and process mismatches, 1:1 integration	Separation of concerns for data and process mismatches, N:M integration

Table 1. Current SOA Implementations and WSMOLX Comparison

We summarize our approach in comparison Table 1. Without moving towards a more automated (Semantic) SOA, the existing IT systems will become more and more complex, up to the point where they are unmanageable. WSMOLX is used in business processes (workflows) where the most suitable service providers can be individually chosen for the business process tasks. The semantic middleware provides a communication platform between business partners, handling their communication even if they do not know each other in advance (discovery), speak different languages or come from different domains (integration).

3 Technical Solution Provided by WSMOLX

We present a technical description of the WSMX framework in context of two scenarios of the SWS-Challenge: discovery (part A of Figure 1) and mediation

(part B). SWS-Challenge is a community-driven initiative that provides a set of increasingly difficult problems with real Web services and an evaluation methodology. We augmented the services with semantic descriptions for the WSMOLX framework to operate upon, demonstrating the benefits of semantics in the context of service integration and discovery.

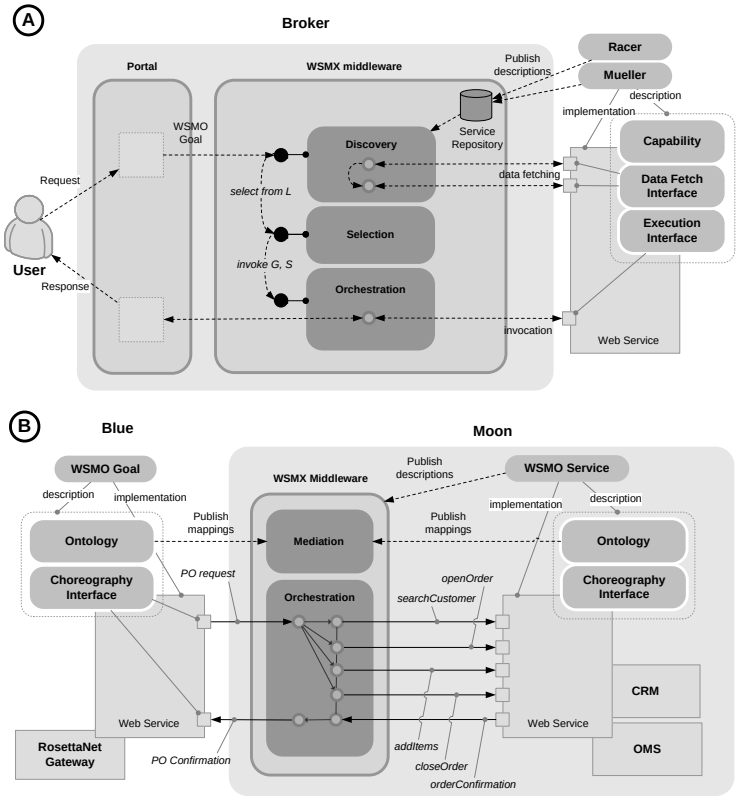

Figure 1. Solution Architectures for SWS-Challenge Scenarios

The core of both solutions is the WSMX middleware located between service requestors and providers (Figure 1). The functionality of the following WSMX components was required by the scenarios: **Discovery** (A) defines tasks for identifying and locating Web services, **Selection** (A) selects the most appropriate service according to user's preferences, **Orchestration** (A and B) executes the composite Web service business process, **Mediation** (B) resolves the heterogeneity issues at the data and process levels, **Reasoning** (A, B) performs logical reasoning over semantic descriptions of services.

The actors of the first scenario are a user wishing to buy some products with shipment to a particular location and a number of third-party companies (brokers or e-hubs) capable of providing such a service under certain conditions (places of shipment, price etc.). Our approach to discovery is to match a WSMO Goal (i.e. a semantic description of what the user wants) with a WSMO Web service

(i.e. what a company offers, semantically described), as well as to use additional data not available in the semantic descriptions (e.g. shipment price). WSMX can dynamically fetch the additional information during runtime discovery through a specific Web service data-fetching interface [4] and use it for service ranking.

The mediation scenario describes a common integration problem: data and process level mismatches between two companies. Moon, a trading company, uses two back-end systems to manage its order processing: a Customer Relationship Management system (CRM) and an Order Management System (OMS). The scenario describes how Moon has to interact with its partner company using the RosettaNet PIP 3A4 purchase order specification. Using the WSMT data mapping tool we map the RosettaNet PIP 3A4 message to messages of the Moon back-end systems. The WSMX Data Mediation component resolves mismatches on the message content level using mappings provided during design time, while Process Mediation handles message incompatibilities during the execution time following service public process analysis.

4 Evaluation and Conclusions

WSMOLX is a key SWS technology utilized and developed in projects like SUPER[7] (applying SWS technologies to business process management), SemGov[8] (SWS in the e-government domain), SWING[9] (SWS in the geospatial domain), SOA4ALL[10] (lightweight semantics for mash-ups) and others.

We have presented our service discovery and integration solutions in context of SWS-Challenge problems, showing how augmenting services with semantics increases the automation in service use. Evaluation has been performed by peer-review and the criteria targeted the solutions' adaptivity (i.e. solutions should handle introduced changes by modification of declarative descriptions rather than code changes). In both scenarios the WSMX proved to be a generic and flexible solution where the adaptation to the new scenario required changes in the service and goal descriptions and no code related changes were necessary.

References

1. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic Matching of Web Services Capabilities. In *ISWC*. 2002.
2. C. Pautasso, O. Zimmermann, and F. Leymann. Restful web services vs. "big"' web services: making the right architectural decision. In *WWW*. 2008.
3. D. Roman, *et al.* Web Service Modeling Ontology. *Applied Ontologies*, 1(1):77 – 106, 2005.
4. T. Vitvar, M. Zaremba, and M. Moran. Dynamic Service Discovery through Meta-Interactions with Service Providers. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007)*. 2007.
5. T. Vitvar, *et al.* Semantically-enabled Service Oriented Architecture: Concepts, Technology and Application. *In SOCA, Springer London*, 1(2), 2007.

[7] `http://www.ip-super.org`
[8] `http://www.semantic-gov.org`
[9] `http://www.swing-project.org`
[10] `http://soa4all.org`