In Proceedings of the IEEE EDOC Workshop on Model-Driven Semantic Web (MDSW 04), *September 2004.*

An MDA-based Approach for Facilitating Adoption of Semantic Web Service Technology

Gerald C. Gannod[*][†]
Division of Computing Studies
Arizona State University - East Campus
7001 East Williams Field Road, Mesa AZ 85212
gannod@asu.edu

John T. E. Timm
Dept. of Computer Science & Engineering
Arizona State University - Tempe
Box 878809, Tempe, AZ 85287-8809
area51@asu.edu

Abstract

A semantic web service extends the capabilities of a web service by associating a semantic description of the web service in order to enable better search, discovery, selection, composition, and integration. Semantically-rich languages such as OWL-S have been created in order to provide a mechanism for describing the semantics of semantic web services. Unfortunately, for the common developer, the learning curve for such languages can be steep, providing a barrier for adoption and widespread use. Model Driven Architecture (MDA) is an approach to software development that is centered on the creation of models rather than program code, allowing for a separation of concerns between a specification and an implementation. We are developing an approach that allows an architect to focus on creation of composite web services by specifying a semantic web service using OWL-S specifications. These composite services are specified using standard UML model and generating specifications and applications using MDA concepts. As such, difficulties caused by a steep learning curve for OWL-S can be mitigated by using a language that has a wide user base, thus facilitating adoption.

1. Introduction

A *web service* is a loosely coupled component that exposes functionality to a client over the Internet (or an intranet) using web standards such as HTTP, XML, SOAP, WSDL, and UDDI. A *semantic web service* extends the capabilities of a web service by associating a semantic description of the web service in order to enable better search, discovery, selection, composition, and integration. Semantically-rich languages such as OWL-S [1] have been created in order to provide a mechanism for describing the semantics of web services. Unfortunately, for the common developer, the learning curve for such languages can be steep, providing a barrier for widespread adoption.

Model Driven Architecture (MDA) is an approach to software development that is centered on the creation of models rather than program code. The primary goals of MDA are portability, interoperability, and reusability through an architectural separation of concerns between the specification and implementation of software. In MDA-based approaches, the focus is upon creation of software via the development of models specified using standard and widely adopted languages such as UML.

Clements and Northrop define a software product line to be a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed way [5]. A product line approach is a disciplined methodology for strategic reuse of source code, requirement specifications, software architectures, design models, components, test cases, and the processes for using the aforementioned artifacts. Software product lines can provide large-scale productivity gains, shorter time-to-market, higher product quality, increased customer satisfaction, decreased development and maintenance cost.

In previous work, we developed an approach for facilitating the dynamic integration of software as services using the Jini interconnection technology [2]. In that work, the focus was upon the automatic generation of *glue code* that enabled a developer to focus on integration via specification of software architectures using ACME [3] rather than on the manual construction of the interconnection code. In this paper, we describe ongoing work that uses a strategy similar to our previous work. Specifically, we are developing an approach that allows a developer to focus on creation of semantic web services and associated OWL-S specifications via the development of a standard UML model. As such, difficulties caused by a steep learning curve for OWL-S can be mitigated by using a language that has a wide user base, thus facilitating adoption. To facilitate this approach, we use a product-line strategy to manage web services as assets within a product-line architecture and MDA to enable OWL-S specification generation and product development.

[*]This author supported by National Science Foundation CAREER grant No. CCR-0133956.
[†]Contact Author.

The remainder of this paper is organized as follows. Background material on MDA and Software Product Lines is presented in Section 2. Section 3 describes our proposed approach, including requirements, while Section 4 concludes and suggests future investigations.

2. Background

Model Driven Architecture (MDA) [4] is a standard produced by the Object Management Group (OMG). The goal of MDA is to separate the design of application or business logic from the implementation platform. Designs are specified in a platform-independent model (PIM) that can then be translated to a platform-specific model (PSM) by an MDA tool, which is then used to generate the implementation. MDA is dependent on and makes use of several other OMG standards including the Unified Modeling Language (UML), Meta-Object Facility (MOF), XML Meta-Data Interchange (XMI), and Common Warehouse Meta-model (CWM).

Software product line development involves three essential activities: core asset development, product development, and management. Core asset development involves the creation of common assets that will be used in individual products as well as the evolution of the assets in response to development user feedback, new market needs, etc. When combined with managerial strategy, core asset development embodies the domain engineering activity. Product development, or application engineering, creates individual products by reusing common assets, provides feedback to core asset development, and engineers product updates as core assets evolve and new requirements come. Management includes technical management and organization management, where technical management handles new requirements coming from inside and outside the organization, and coordinates core asset development and product development activities.

In our work we use both MDA and product-line concepts to enable management and generation of OWL-S specifications and code. Specifically, MDA is used to manage creation of product-line architectures in order to hide the details and knowledge necessary needed to create OWL-S specifications. We use product-line concepts as a means for managing reuse and product development.

3. Approach

In this section we describe the general approach that we are developing to support development of service composition using semantic web technology.

3.1. Overview

In order for many new techniques and technologies to gain entry into a market, many factors must be considered. While standardization can often be a major reason why adoption of a new technology succeeds, another factor is *ease of use*. That is, if a technology adds little new overhead or requires little in the way of new training, then it is more likely to be adopted. For instance, the benefits of formal methods for software development has been described numerous times [6]. However, an adoption barrier to formal methods by much of the software development community remains in the form of a lack of education. From this standpoint, one of the overarching philosophies of the research described in this paper is the following:

> *Development of applications that are based on the use of semantic web services should not require knowledge beyond the use of the UML modeling language.*

Specifically, it is our belief that in order for techniques that are based on the use of the semantic web and semantic web services, a bridge should be created using UML that facilitates adoption. In our work, this bridge is enabled via the use of model driven architecture.

Web services have gained a great deal of attention as a way for organizations to use information as a commodity. While many web services currently exist, very few have semantic descriptions associated with them. Furthermore, from the viewpoint of an "end-user", a given web service may or may not provide exactly the data that is needed for a given task. However, it may be the case that a group of web services, when properly composed and integrated, will provide the desired outcomes. Based on the above, we have identified two primary requirements that are being used to drive our approach:

- *R1* *The approach should be able to incorporate the use of both web services (e.g., services with WSDL specifications only) and semantic web services (e.g., services with semantic descriptions)*
- *R2* *The approach should facilitate composition of services to form applications or federations.*

The intent of requirement *R1* is to leverage existing web services regardless of the state of their specification. Specifically, the requirement is a recognition of the fact that adoption issues exist in the community and the expectation of widespread use of semantically-rich languages such as OWL-S is unreasonable. However, we do assume that at the very least, a WSDL specification does exist, which is reasonable given that frameworks such as the .NET platform provide facilities to automatically generate WSDL specifications.

Requirement *R2* along with the philosophy stated above embody the heart of our approach: a technique for integrating services into composite services by generating an OWL-S specification via the use of MDA.

The OWL-S language consists of three separate parts. The *Service Profile* is a description of what the service does. It is similar to, yet more powerful than WSDL. The *Service Model* is a description of how service works (e.g., the semantics of the service). Finally, The *Grounding Model* is a description of how to access the web service. From a WSDL description, an OWL-S description can be generated using XSLT. However, for semantic web service composition, other information is required to describe the operational behavior of the service (e.g., the service model). We are developing an approach for generating an OWL-S Service Model by taking an XML-based representation of a UML activity diagram and using XSLT to transform the diagram. In this approach, the software developer specifies the operation of a Web service as a composition of other services using UML. An automated design tool then takes the UML description of the service and converts it into the OWL-S Service Model description.

As part of our investigations, we are developing domain models for web services and web service composition as a way to facilitate translation via XSLT of WSDL specifications to OWL-S Grounding Models and UML Activity Diagrams to OWL-S Service Models, respectively. Furthermore, we are investigating the use of different service matchmaking techniques including lightweight signature-based techniques [7] and more computationally expensive semantic techniques [8].

3.2. Process

In our approach we use a product-line strategy to facilitate service composition. Specifically, our approach is based on the development of a product-line architecture that characterizes a family of products that exists in the product line. In a product-line approach, products share a common set of characteristics that are implemented by a set of core assets [5]. Individual products in a product line are characterized by variabilities that must be configured or developed to meet the requirements of the individual product. By reusing assets, which includes processes as well as components, organizations can achieve product differentiation and personalization while benefiting from reuse.

Figure 1 shows the context of the approach that we are currently developing. The diagram depicts the activities of the approach as existing on a spectrum of specifications that range from the semantically rich to the syntactically rich. In this context, a series of steps lead to the eventual creation of products that consist of a collection of services. Starting from the left, a domain expert and a software architect work together to develop a number of artifacts based on requirements for a product line including an ontology for the product line, a workflow depicting the series of operations for potential products in the domain, a product-line architecture that specifies the structure of the the family of products, and a list of services that meet the requirements of various parts of the product-line architecture. In this context, all components in the product-line architecture are implemented by services that are either local or distributed. Furthermore, MDA concepts are used to facilitate creation of the appropriate OWL-S specifications.

As an example, consider the domain of E-Learning (e.g., delivery and adminstration of course content in an on-line environment). Requirements for this domain focus on providing content, learning activities (e.g., homework and laboratories), collaboration, and assessment. Within this context, a domain expert and a software architect collaborate to characterize the domain using an ontology and by specifying a workflow. In addition, the architect creates a product-line architecture for the E-Learning domain while the domain expert identifies potential services that could be used to populate the product line.

Using the information provided by the domain expert and software architect, software developers create the software needed to enable a production capability for the product line by developing the software needed to map between different classes of services in the product-line architecture. In addition to creating software that enable the mappings, the software developers create the infrastructure necessary to control the flow of collaborations between classes of services. Since we are using a product-line approach, the mappings are dependent only upon the ontologies for the domain and the colloborations between services well-defined (e.g., content services always provide content). Using the E-Learning example, software developers are responsible for creating the product-line framework for the E-Learning product-line domain by providing software that supports operations like test and quiz management in general as well as the software that will facilitate mapping to specific test and quiz management services.

In our investigations, our primary focus is upon creating MDA tools to support creation of OWL-S specifications and for supporting the software development step. That is, we are developing tools and techniques for generating service-based product-line frameworks including using MDA to facilitate creation of OWL-S specifications of a product-line architecture and transformation of the product-line architecture at the PIM-level, to a PSM-level model suitable for generating a framework, where the service mappings provide information on realizing implementations. In addition, we are developing service search technology to support syntactic and semantic service match.

Once a product-line framework or generator is developed a domain developer creates new products. At the end of the diagram is the user that utilizes the end products to perform some task. In our approach the domain developer is considered a user of the framework in the sense that we do not

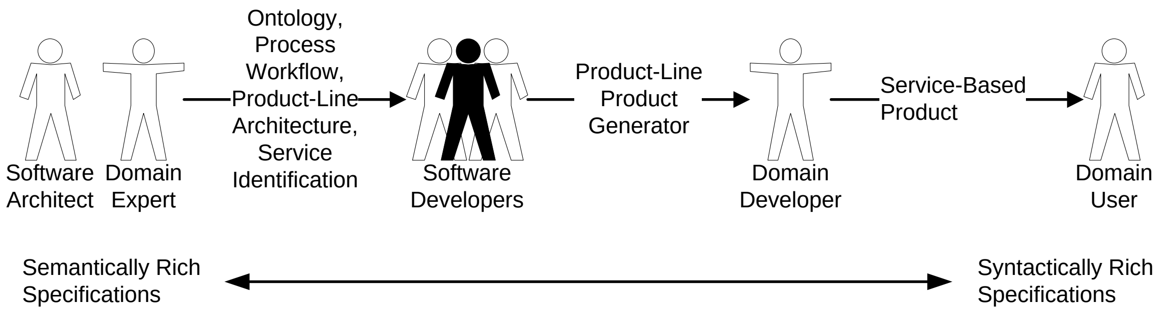

Figure 1. Context of Investigations

require programming knowledge in order to implement a product. That is, the act of product development is an activity of selecting services (e.g., configuration). For instance, in the E-Learning example, the domain developer could be an instructor that wishes to create a course while the domain user is a student.

3.3. Discussion

One of the technologies that allows for the work described above is the semantic web. Specifically, the use of ontologies to describe the domain for a product-line enables mapping and interchangability of services, without which the product-line generator cannot be achieved. However, in order to effectively use the semantic web, MDA technology is needed to make the activity of creating service-based product-line frameworks cost effective. Furthermore, for domain developers to be come fully involved in this approach, MDA must utilized to ensure that product development stays on the syntactic end of the spectrum of specifications.

4. Conclusions and Future Investigations

With the advent of semantic web services, enterprise applications have the ability of becoming more dynamic and better suited to specific users and customers. However, in order to allow a user to gain control of their application experience, enabling technologies such as model driven architecture are needed to provide the bridge between the technologies being widely used by practitioners and the advanced capabilities of the semantic web. Currently we are in the process of developing the tools that support the approach discussed in this paper. As part of our future investigations, we intend to perform a number of case studies on systems of varying size and domains. Furthermore, we intend to perform a user study involving software developers to determine the likelihood of adoption of the approach.

References

[1] The OWL Services Coalition. Owl-s: Semantic markup for web-services. [Online] Available http://www.daml.org/services/owls/1.0/owl-s.html, December 2003.

[2] Gerald C. Gannod, Sudhakiran V. Mudiam, and Timothy E. Lindquist. Automated Support for Service-Based Software Development and Integration. *Journal of Software and Systems Special Issue on Automated Component-Based Software Engineering*, 2004 (in press).

[3] David Garlan, Robert T. Monroe, and David Wile. Acme: An Architecture Description Interchange Language. In *Proceedings of CASCON'97*, pages 169–183, Toronto, Ontario, November 1997.

[4] Joaquin Miller and Jishnu Mukerji et al. MDA Guide Version 1.0.1. Technical Report omg/2003-06-01, Object Management Group, June 2003.

[5] P. Clements and L. M. Northrop. *Software Product Lines – Practices and Patterns*. Addison-Wesley, 2002.

[6] Edmund W. Clarke and Jeannette M. Wing. Formal Methods: State of the Art and Future Directions. Technical Report CMU-CS-96-178, Carnegie Mellon University, August 1996. Group report from the "Strategic Directions in Computing Research Formal Methods Working Group ACM Workshop".

[7] Gerald C. Gannod and Sushant Bhatia. Facilitating automated search for web services. In *Proceedings of the 2004 IEEE International Conference on Web Services*, July 2004.

[8] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. Automated discovery, interaction and composotion of semantic web services. *Journal of Web Semantics*, 1(1), 2003.