A Meta-Modeling Approach to Web Services

Fei Cao, Barrett R. Bryant, Wei Zhao, Carol C. Burt
University of Alabama at Birmingham
{caof, bryant, zhaow, cburt}@cis.uab.edu

Rajeev R. Raje, Andrew M. Olson
Indiana University-Purdue University-Indianapolis
{rraje, aolson}@cs.iupui.edu

Mikhail Auguston
Naval Postgraduate School
auguston@cs.nps.navy.mil

Abstract

Web Services (WS) technology is becoming pervasive in the development of distributed systems and is an appealing vehicle for service presentation and horizontal integration. On the other hand, Model Integrated Computing (MIC) offers a means of system integration in the vertical direction by using domain-specific modeling, and then synthesizing the software system from the high-level model using a model-specific generator. This paper presents a meta-modeling approach to WS to explore the application of MIC in WS development and its contribution.

1. Introduction

Web Services (WS) technology emerges as a Service Oriented Computing (SOC) ([8], [9]) paradigm to provide a platform-independent solution for system integration horizontally: WS is built upon open standard XML and HTML for service description and transportation, and software systems can be presented as WS so as to be exported and consumed by heterogeneous peers in the distributed environment.

For service description in Web Services Description Language (WSDL), though its XML-based representation is easy for machine processing using widely existent XML parsers, such specification is not straightforward for human comprehension, with service architecture lost in the pure textual form, and hand-crafting service description with WSDL is error-prone. To overcome this problem, there are tools on the horizon such as AXIS¹, and the Microsoft Net framework that provide the capacity of automatically generating WSDL by parsing implementation code (such as Java and C#), and vice versa. However, WSDL represents the design level knowledge, and the process of generating WSDL from implementation is in

¹ http://ws.apache.org/axis/
When modeling a WSDL for real business domain services implemented with a specific technology, we use the generalization relationship to extend those WSDL elements in Figure 2 rather than embedding the business domain service information as attributes to those WSDL elements. This avoids obfuscation of business and technology domain structure (actually meta-models of business/technology domain applications) with WSDL elements. The business domain information applies a generalization relationship to the operation entity, and technology domain information applies a generalization relationship to the binding entity. To exemplify, Figure 3 is a simple banking domain service specification.

Figure 2 shows the ER-based meta-model of this banking service WSDL. As can be seen from the figures, a typical business domain service represented as WSDL involves the extension of ER elements, which is associated to almost all the elements of WSDL. Nevertheless, by using the ER-based meta-model, such extension still keeps the original WSDL meta-model as shown in Figure 2 without being restructured, which helps generating WSDL from models with consistency.

### 2.2 The Mapping from ER based Meta-model to Other Forms of Meta-model

In GME, the containment relationship is represented by using a model element (tagged with $\langle\langle\text{model}\rangle\rangle$), which, in contrast to an atom element (tagged with $\langle\langle\text{atom}\rangle\rangle$), can contain other modeling elements. Also the contained elements can be promoted as ports of the model to have direct connections with external modeling elements. GME uses a root model as an entry point of access to all the modeling elements. Also, the relationship of ER is represented in GME as a first-class modeling element, connection (tagged with $\langle\langle\text{connection}\rangle\rangle$), with a connector in the form of a dot to associate this relationship with two modeling elements (entities).

The mapping from the ER-based meta-model to the counterpart in GME is based on the relationships in the ER representation. Three cases are involved as is shown in Figure 4. For the sake of limited space, below we only describe the mapping rules for case 3, i.e., B is specialized from A. In this case, A is rendered by an abstract FCO (First Class Object, tagged with $\langle\langle\text{FCO}\rangle\rangle$, represents an abstract generalization of other modeling constructs), a modeling element to be used as an abstract interface in GME, and B is represented as an inherited class of that FCO. Note there are two special treatments here: firstly, for the input/output elements of Figure 2, they are only used to tag the connection (named either “input” or “output”) between message entities and its interconnecting entities in GME; secondly, the generalization

---

2 http://www.omg.org/MDA/
3 http://www.omg.org/cgi-bin/doc?formal/00-04-03
A bank provides the service for users to set up accounts. Account information includes personal data including Name, SSN, phone number, address, and account data including Account Number, PIN, Transaction Record, Balance. There are two types of accounts: checking account and savings account.

For the bank side, it provides such services as: Account Verification, Account Query, Deposit, Withdraw, and Transfer.

The banking service implementation may use such technology as RMI, J2EE, and CORBA. Also it will enforce some Quality of Service (QoS) requirements such as Availability, Dependability, Capacity.

Figure 2: the ER-based Meta-model of Banking Service WSDL: the three parts enclosed with dashed line represent the extended part to the WSDL meta-model.
relationship between binding and portType is actually treated as an association when modeling in GME, because the binding entity actually attaches values of the chosen protocol to the portType in WSDL rather than in the real sense of inheritance.

Figure 5 shows the meta-model created by mapping from the WSDL meta-model of the banking domain with ER representation to that in the GME strictly observing the above mapping rules. Note the model WebService corresponds to the service entity in Figure 2. The lower part of the models in Figure 5 are attributes for the related models to be instantiated in the modeling phase as described in the next section.

Based on this meta-model, a WS modeling environment can be constructed, and a generator based on this meta-model can be created to interpret WS models to generate WSDL. The WS modeling environment as well as generated WSDL is described in [1].

3. Related Work

In [6], MDA is used together with workflow technology for modeling and composing WS. But the authors do not provide a guideline as to how to create the meta-models. Also the mapping from PIM to PSM is not detailed. In contrast, we focus on meta-modeling WSDL only, while the meta-modeling approach is more complete and general. In [7], an MDA approach is used for BPEL code generation from a UML design. This approach uses XML processing technology for UML model exchange. Comparatively the XML representation for the ER model is much simplified and easy to process in our approach. Code generation in [7] is based on the UML profile mapping, which is not as flexible as a generator-based approach in our case.

4. Conclusion

WS domain-specific modeling environment provides a user-friendly environment to build WS with underlying WS-specific details abstracted. This paper presents a general meta-modeling approach to WS, which is used for the construction of WS domain-specific modeling environment. In particular, we showed the merits of using the ER representation as an intermediate form for deriving and evolving meta-models to avoid the ad-hoc nature of constructing meta-models, which is an problem that is often not addressed (such as in [1]), particularly in constructing large-scale meta-models.

Meta-modeling of WSDL is a static structural modeling. Future work will include meta-modeling of WS behavior such as WS orchestration.

5. Acknowledgements

This research is supported by the U. S. Office of Naval Research under the award number N00014-01-1-0746.

6. References


---