Web Modeling-based Approach to Automating Web Services Mediation, Choreography and Discovery

Marco Brambilla[1], Stefano Ceri[1], Dario Cerizza[2], Emanuele Della Valle[2], Federico Facca[1], Piero Fraternali[1], Christina Tziviskou[1]

[1] Dipartimento di Elettronica e Informazione, Politecnico, 20133 Milano, Italy
{mbrambil, ceri, facca, fraterna, tzivisko} @ elet.polimi.it
[2] CEFRIEL, 20133 Milano, Italy
{cerizza, dellava}@cefriel.it

Abstract. We propose an approach for the discovery and integration of Semantic Web Services that exploits modern Web engineering methods, including visual declarative modeling (based on the WebML language), visual data marshalling, and automatic code generation. The approach to automatic interaction between services is based on a relational representation of the WSMO ontology model and on a set of visual primitives for defining the interaction between services and the choreography structure. Choreography, front-end, and services implementations are completely modeled by WebML schemas. Service discovery is based on a research prototype engine called Glue.

1 Introduction

The development of applications based on Semantic Web and Web services is currently lacking a set of high level abstraction for design of applications. We propose an approach to the design that exploits modern Web engineering methods for the specification of all the aspects of the application:

- Storage and management of WSMO-based ontology: ER diagrams representing relational storage of the ontology, diagrams for ontology content management [2];
- Front-end of the application: visual diagrams representing Web sites [2];
- Web services interfaces and implementation: visual diagrams including Web service specific primitives for invocation and publishing[5];
- Choreography of Web services interactions: visual diagrams including conditional navigation together with possible explicit representation of workflow primitives within the hypertext [1];
- Discovery services: invocation of services of a prototype discovery engine called Glue, which exposes Web services [4].

In the context of the Challenge, our approach provides the following contributions:

- **Phase I:** WebML modeling of the Web services needed for the specified interaction; automatic generation of the implementation; data mismatch solving through appropriate primitives; demo of the application implementing the requirements.

- **Phase II:** for *Scenario 1* (Internal System Upgrade): dynamic update of the Ontology through content management Web site (modeled in WebML and automatically generated); automatic invocation of the new services based on the new results of the query on the WSMO ontology. For *Scenario 2* (New RosettaNet PIP): just-in-time redesign of the WebML diagrams and redeploy of the new application, thus demonstrating the power of a model-based approach wrt brute-force hand coding. For *Scenario 3* (Discovery of new Business Partner): navigational querying of the ontology, and recursive refinement of the query; submission of the query to the Glue Discovery Engine, for extracting the matching services wrt the user needs.

2 Technologies employed

Web sites, Web Services, and Data Marshalling Primitives. The specification of a *WebML application* [3] consists of a conceptual model (E-R model) describing application data, and of one or more site views (e.g., for different types of users), expressing the Web interfaces. A site view is a graph of pages; units, representing publishing of atomic pieces of information; and operations on the underlying data or operations performing arbitrary business actions (e.g. sending e-mails). Units are connected by links, to allow navigation, parameter passing and computation of the hypertext.
WebML meta-model has been extended with a set of *Web service units* [2, 5], corresponding to the WSDL classes of Web service operations. These units are exemplified in Fig. 1.
Request-response and *one-way* (Fig. 1 (a)) operations model services invocation and are triggered by an input link; an XML message is composed, and then sent to a remote service as a request. *Notification* and *solicit-response* (Fig. 1 (b)) are instead triggered by the reception of a message, thus they represent the publishing of a Web service, which is exposed and can be invoked by third party applications. The WebML diagrams dictate how to build the response to the invoker. Publishing of Web services is specified using three concepts: Service view, Service, and Port.

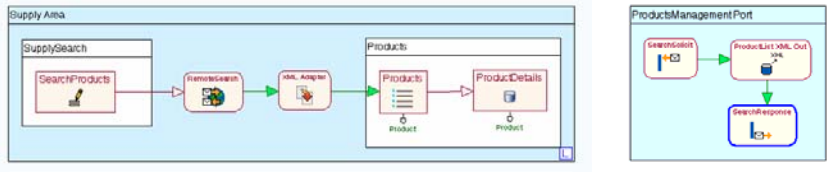

Fig. 1. WebML Web service units: Service invocation (a) and Service publishing (b)

The capability is needed of *grounding* Web services modeled in WebML to the XML format of Web service messages and of *data-mediation* in order to perform and maintain XML-to-XML mapping between different message formats. For dealing with the grounding of WebML, we introduce the so called “canonical XML format”, an intermediate format between the E-R representation and the arbitrary XML data representation of the messages. The data-mediation problem is addressed by the means of three additional WebML units: *XML-in*, that transforms XML fragments into rela-

tional data; *XML-out*, performing the opposite transformation; and *XSL Adapter* unit, that applies arbitrary XSLT transformations to XML messages. The transformations to be applied can change dynamically and can be specified using a visual XSLT generator tool.

Representation and Management of WSMO Ontology. The data model for describing ontologies that conform to WSMO includes metadata describing the ontology semantic, and ontology data definition together with ontology instances information.
Different site views and service views can be defined on this data schema: ontology browsing hypertexts for final users that need to navigate the stored knowledge; ontology editing and maintenance hypertexts for content managers; service views for publishing Web services for remote and automatized access to the knowledge. New WebML primitives for permitting semantic querying and reasoning, are under investigation.

Glue, a WSMO compliant discovery engine. Glue aims at developing an efficient system for the discovery of semantically described Web services following a mediator centric approach [4]. Glue refines the conceptual model of WSMO discovery and its related architecture. It is designed and built as a system suited for medium scale deployment (up to some tens of classes of Web Service Description and of classes of goals, using ontologies of a couple of thousand concepts each, but with some hundreds of instances of Web Service Description in each class) and it is developed as a lightweight stand-alone application accessible via standard Web services.
.
Architecture Implementation. The concepts previously illustrated have been implemented inside a CASE tool for Web application development, called WebRatio (www.webratio.com), an integrated environment for visual specification automatic code generation of Web applications.
The architecture of WebRatio, shown in Fig. 3 (a), consists of two layers: a design layer, providing functions for the visual editing of WebML schemas, and a runtime layer, implementing a Model View Controller Web application framework [2]. These layers are connected by the WebRatio code generator, which exploits XML transformations to map the visual specifications edited in the design layer (which are stored as XML) into application code executable within the runtime layer. Layers are extensible and customizable through plugins.
WebRatio generates service-oriented applications compliant to MVC-based organization, in which the components produced by the code generators fit into a well-established framework. The various hypertext primitives of WebML map into the layers of the MVC2 architecture (see Fig. 3 (b)). Web services invocation, deployment, and publishing required to extend the framework, by means of a SOAP listener and a set of Web service-specific WebML conceptual primitives.

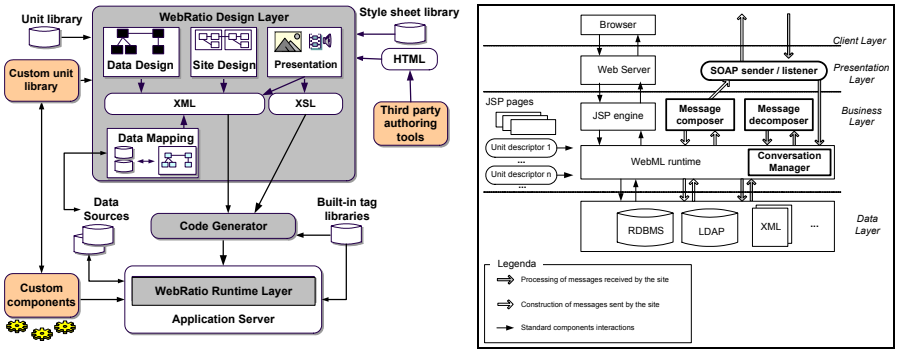

Fig. 3. WebRatio Compile-time (a) and Run-time (b) Architecture

References

1 Brambilla, M., Ceri, S., Comai, S., Fraternali, P., Manolescu, I., Specification and design of workflow-driven hypertexts, Journal of Web Engineering, 1(2) April, 2003.

2 Brambilla, M., Ceri, S., Fraternali, P., Acerbis, R., Bongio, A.: Model-driven Design of Service-enabled Web Applications, SIGMOD 2005, Industrial Track.

3 Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications, Morgan-Kaufmann, December 2002.

4 Della Valle, E. and Cerizza, D.. The mediators centric approach to automatic webservice discovery of Glue. In Martin Hepp, Axel Polleres, Frank van Harmelen, and Michael R. Genesereth, editors, MEDIATE2005, volume 168 of CEURWorkshop Proceedings, pages 35–50. CEUR-WS.org, 2005..

5 Manolescu, I., Brambilla, M., Ceri, S., Comai, S., Fraternali, P.: Model-Driven Design and Deployment of Service-Enabled Web Applications, TOIT, Volume 5, number 3 (August 2005).