

Exploring the combined potential of Web sites and Web services¹

Ioana Manolescu

INRIA Futurs

78153 Le Chesnay, France

ioana.Manolescu@inria.fr

Stefano Ceri

Marco Brambilla

Piero Fraternali

Sara Comai

Dipartimento di Elettronica e Informazione, Politecnico di Milano

Via Ponzio 34/5, 20133 Milano, Italy

ceri|mbra|brambill|fraterna|comai@elet.polimi.it

¹This research was developed while the first author was visiting Politecnico di Milano, and is part of the WebSI (Web Service Integration) project, funded by the EC in the Fifth Framework.

ABSTRACT

Recent *web service* standards, based on WSDL, have been adopted as a process-to-process communication paradigm, enabling large-scale, inter-organization application integration. In this paper, we extend a declarative model and language for specifying data-intensive Web applications, in order to model complex interactions between the application (driven by the user) and remote processes (represented as services). Thus, we demonstrate how a declarative web site specification method can be effectively turned into a tool for web service usage and composition, particularly suited for interactions between human users and web services.

Categories and Subject Descriptors

H.3.5 [Online Information Services] – commercial services, data sharing, web-based services.

General Terms

Design, Human Factors, Standardization, Languages.

Keywords

Web site design, web site management, web services, application integration.

1 Introduction

The current focus for achieving interoperability on the Web is *application integration*: businesses with related interests aim at automatic interaction between their respective business processes. *Web services* [WSDL] are very popular as a means of interaction between such processes; languages like BPEL4WS [BPE] have been proposed for Web service composition, and are currently under study.

The implementation of a business process relies on automatic, computerized components, but also on input from *human users*. The paradigm of *Web applications*, with which human interact by activating links and/or providing input, has gained wide

popularity. To fully support user participation in a process driven by web services, we need a way of prototyping *fast a friendly, low-weight, low-cost, still correct* interface for users.

Our solution to this need is based on the existing experience on *formal models and languages for declarative specification of Web applications*. We identify a minimal set of extensions required by WebML [CF+02], a data-intensive web site specification language, in order to specify web applications tightly integrated with web services.

Our *high-level, declarative methodology* allows to design, specify, and automatically generate web applications for several purposes, as summarized in Figure 1.

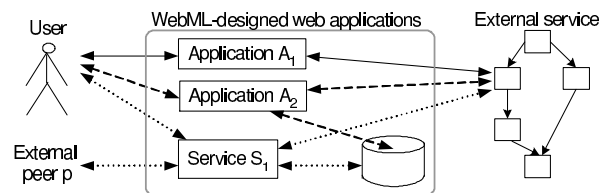


Figure 1. Interaction patterns for using the target applications and services.

(1) *Interacting with web services*: an application like A_1 in Figure 1 is an HTML-based shell around the XML syntax specific to the external web service. A_1 enables a user to conduct an interaction (continuous lines) with such services.

(2) *Integrating databases and web services*: the application A_2 in Figure 1 allows a user to exploit, together, database content, and the remotely developed service (dashed lines).

(3) The same methodology allows for *specifying new web services by composing existing services, local resources, and human user interaction*, as illustrated by service S_1 . S_1 combines database information, access to external services, and input from human users, into a new, added-value component, that an external peer p may in turn use (dotted lines).

2 The WebML model and architecture

This section explains the components present in the WebML architecture before the present work, shown in Figure 2 in regular lines and fonts. A WebML site definition consists of a *data schema* structured in Entities and Relationships, and a *hypertext* describing the content of the site in terms of the underlying data model. In the hypertext model, web pages are defined as sets of interconnected *units*. A unit is an elementary piece of content corresponding to a parameterised query over the underlying E-R data model. Units, pages and their

correlations are specified using WebML's library of graphical primitives.

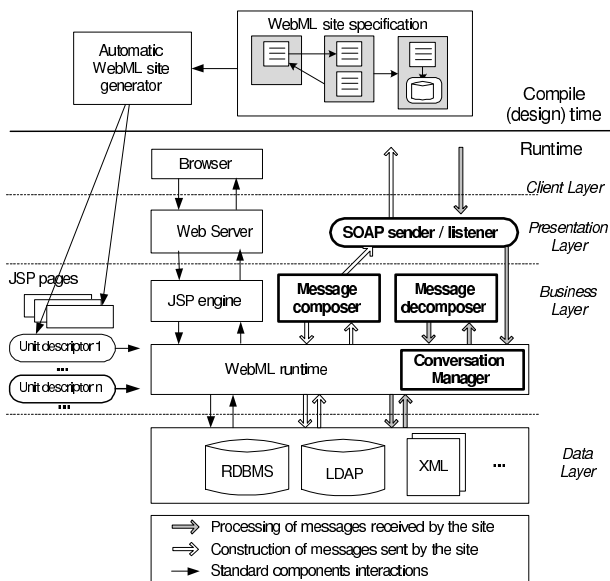


Figure 2. Design and execution environments of WebML-specified Web applications.

At compile time (Figure 2, top), from the graphical site definition are generated: a set of JSP pages, and for each unit, link and page included in the site design, an XML *descriptor* including the information needed to instantiate it in the functional web site.

At runtime (lower part of Figure 2) the site is instantiated and "run". Clients issue requests via a browser to the Web server, that in turn calls the JSP engine; the generated pages include customized JSP tags that define the placement of the WebML units into the JSP pages. The content required by the user is assembled by the WebML runtime component, using the descriptors and the data sources in the Data Layer.

This architecture is currently being implemented into the commercial tool WebRatio [WebRatio], which supports the WebML model.

3 Extensions for supporting Web services

We make the following extensions to WebML for using and constructing web services.

Generic entities for managing users' interaction with web services

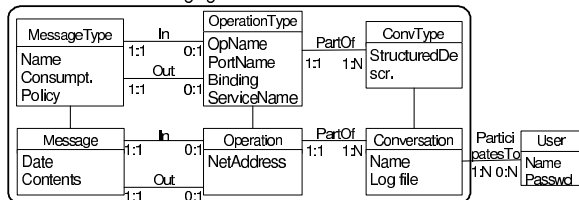


Figure 3. Default E-R schema for a Web application supporting Web services.

At *compile time*, we enrich the data schema of a WebML-specified web application with a few standard entities for recording the interactions with web services (Figure 3). The Message and Operation entities correspond to the WSDL model [WSDL]. We also use a Conversation entity to model a sequence of web service operations, belonging to one or several web services, involving the interaction of zero or more users

with the web application. For a more abstract description level, we use MessageType and OperationType entities [WSDL], as well as a ConversationType entity.

To the basic set of WebML graphical primitives [CF+02] we add six new ones (Figure 4), derived from the types of WSDL web service operations, and from the particular aspects of the interaction between a Web site and web services.

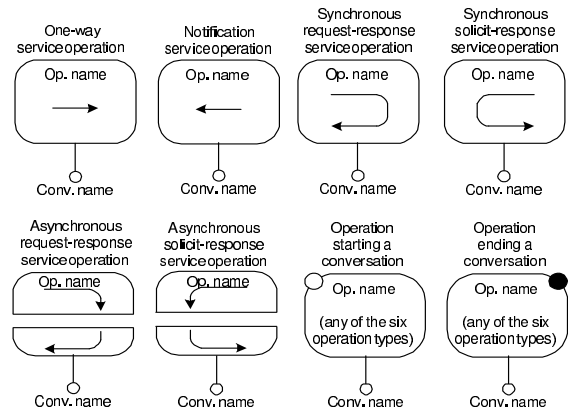


Figure 4. New WebML primitives for Web services.

To reflect the WSDL specification, we introduced icons for one-way, notification, request-response, and solicit-response operations. In the case of *two-message operations* (request-response and solicit-response), there are two ways in which the web site designer may choose to use them in the site. The difference is in whether or not, he wishes to *impose some actions to take place in the Web application in the time interval between sending a request and receiving a response* (respectively, between receiving a solicitation and sending a response). If no action can take place in-between, from the web application's perspective, we term the operation *synchronous*. If some action is taken by the user and/or automatically performed by the application between the two messages, the operation is *asynchronous*. This difference leads us to the six, instead of four, WebML operations for interacting with web services.

Conversation creation and ending are designated by special markers attached to any Web service operation, as shown in Figure 4.

WebML runtime extensions. These extensions are depicted in bold lines and fonts in Figure 2. We extend the runtime to support exchanging web service messages: we add a SOAP sender/listener, and a Conversation Manager for conducting long-running interactions with a service. Finally, we extend the site generation algorithm to produce descriptors for each of the new primitives, which will be used in the generated JSP pages.

A complete example of WebML design for a data- and service-intensive Web application is presented on the full-sized poster.

4 References

[BPE] Business Process Execution Language for Web Services Available at: <http://www.ibm.com/developerworks/webservices>
 [CFB00] S. Ceri, P. Fraternali, A. Bongio: Web Modeling Language (WebML): a modeling language for designing Web sites. WWW9/Computer Networks 33(1-6): 137-157 (2000)
 [CF+02] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, M. Matera: Designing Data-Intensive Web Applications, Morgan-Kaufmann, to appear (2002).
 [SOAP], [WSDL]: Available at <http://www.w3.org/TR>
 [WebML] The WebML Project: <http://www.webml.org>.
 [WebRatio] The WebRatio Tool Suite: <http://www.webratio.com>