Provision and Integration of EDA Web-Services Using WSDL-based Markup

Heinz-Josef Eikerling, Wolfgang Thronicke, Siegfried Bublitz
Siemens Business Services C-LAB
Fürstenallee 11, D-33 102 Paderborn, Germany
[heijo,wolf,sb]@c-lab.de

1 Introduction

The rapid development of Internet related technologies nowadays significantly changes engineering practice. Remote, mobile, and decentralized working enable site-spanning teamwork and demand techniques for collaborative engineering. A major requirement is the integration of dispersed resources into homogeneous workflows ([Ap99], [GaPo99], [PaFo99]), particularly in the EDA domain with its complex engineering design, modeling, and verification.

A resource (human, hardware equipment, software service) is an entity that is active for a certain time during the process to achieve a certain (sub-)result. Whereas formerly [SvB93] the integration of tools in a rather centralized environment was stressed, this paper emphasizes on the integration of engineering services that are widely dispersed and exposed through web interfaces. This imposes new challenges, e.g. to deal with secured environments and intellectual property (IP) protection.

We propose a methodology based on Web Service Description Language (WSDL). WSDL characterizes services by meta-information and retrieves this information through a universal directory interface, enabling the re-use of integrations and permitting to encompass secured environments. The application of the technology is shown with a simple example, shortcomings and limitations are detailed in the conclusions.

2 Related Work

Concepts for deploying a web based design and simulation service were (among others) suggested by National Semiconductors. An approach utilizing Java for providing access to EDA tools like synthesis systems was presented in [LaKh97, LaBr00]. Similar research has been presented in [BrLa00, ChSp98, EDCS]. Most these concepts are rather limited in enhanced contexts, e.g. integrating PDM or ERP systems. The concepts' native, not standardized nature makes it difficult to even retrieve the interfaces of the integrated resources. For the EDA domain, tool integration was formerly tackled by CFI's TES standard [CFI TES] which is not widely used. Workflow management systems (WFMs) control complex processes from a higher level of abstraction [WfMC]. Workflows should also be recognized as integrable entities in the sense of a powerful and sophisticated tool. Yet, only little effort has been spent on a consistent, standardized and coherent approach for the integration of resources through web services, which is the major concern of this paper.

3 Requirements for Resource Integration

3.1 Integrable Resource Types

The to be integrated resources can be classified into 3 major groups:

1. *Physical or hardware resources* consist of special and general purpose computing equipment like printers and PCs. They should be seamlessly integrable into the workflow.

2. *Software resources*: software interfaces to tools or services permitting access to lower level (hardware drivers) or higher level (database access) software components.
3. *Human resources*: for collaborative engineering this means essentially supporting engineers concerning the distribution of work and awareness of the overall state of the workflow.

The coordination of these resources should be done by the workflow system. Thus appropriate language interfaces (C, C++, Java, Perl, ...) for implementing the resource access are needed.

3.2 Modularity/Extensibility/Scalability

Managing the change and customization of integrated resources is necessary to keep track of developments and changing technologies. Integration technology itself has to turn into an adaptive service that can assimilate the functionality in a structured way, leading to a hierarchical and thus scalable integration methodology.

3.3 Mechanisms for Distribution and Deployment

Integration descriptions shall be shareable on different platforms, e.g. as an inherited feature of integration. A practical way to do this is by means of a web server, i.e. integration services are deployed to a web server. Deployed services should also be self-explaining via meta-information on the integrated resources.

3.4 Internet Capability, Dissemination and Use

Integration must handle resources located on Internet hosts accessible via standard Internet protocols like plain TCP/IP, HTTP or SMTP.
Support for the integration concept in general purpose applications like design frameworks or workflow management systems and dedicated applications design tools (e.g., as plug-ins) is required. The user is should not be forced to use a web browser.

4 Service-Integration through WSDL

4.1 Application Scenario

The application scenario consists of a web service for simulating an electronic system. At the beginning, a customer completes a web form containing the model characteristics of the system. The content of this form is turned into a workflow which is implemented on the server. On the server side, a workflow engine processes the workflow and controls the execution of the involved activities, e.g. remote tool invocations and user interaction.

4.2 WSDL, UDDI and Engineering Services

The Web Services Description Language (WSDL) is an XML format for describing network services by identifying the services (and their operations) and providing format definitions for requesting them. Both, client and server demand properly formatted Simple Object Access Protocol (SOAP) requests and responses.
For publishing and finding services, the Universal Discovery, Description, and Integration protocol (UDDI) is used. It is composed of:

- *White pages*: name, address and other business contact information.
- *Yellow pages*: business classification according to existing (non-electronic) standards like NAICS or UNSPSC.
- *Green pages*: technical information about the (web) services provided by a given business.

4.3 Service Request Processing

Figure 1 illustrates the processing of a request (e.g. remote procedure call or object transmission) that has been previously discovered by UDDI. A client (written in Java, Perl,...) addresses the web service by its URL and submits URL, URN and the original request to the API. The SOAP API uses an XML Component to generate a SOAP request (XML encoding of the original request). It may contain additional information, e.g. how to generate a requested object. The generated request is transmitted via a TCP/IP connection (e.g. wrapped by HTTP or SMTP) to the server. The result takes the reverse way back. The client finally receives the decoded result or an error message. As can be seen, SOAP uses XML for transport of data and allows some interception to customize the processing of requests. Moreover, SOAP is also applicable to highly secured and heterogeneous/dispersed environments due to the support of HTTP or SMTP as a bearer for XML messages.

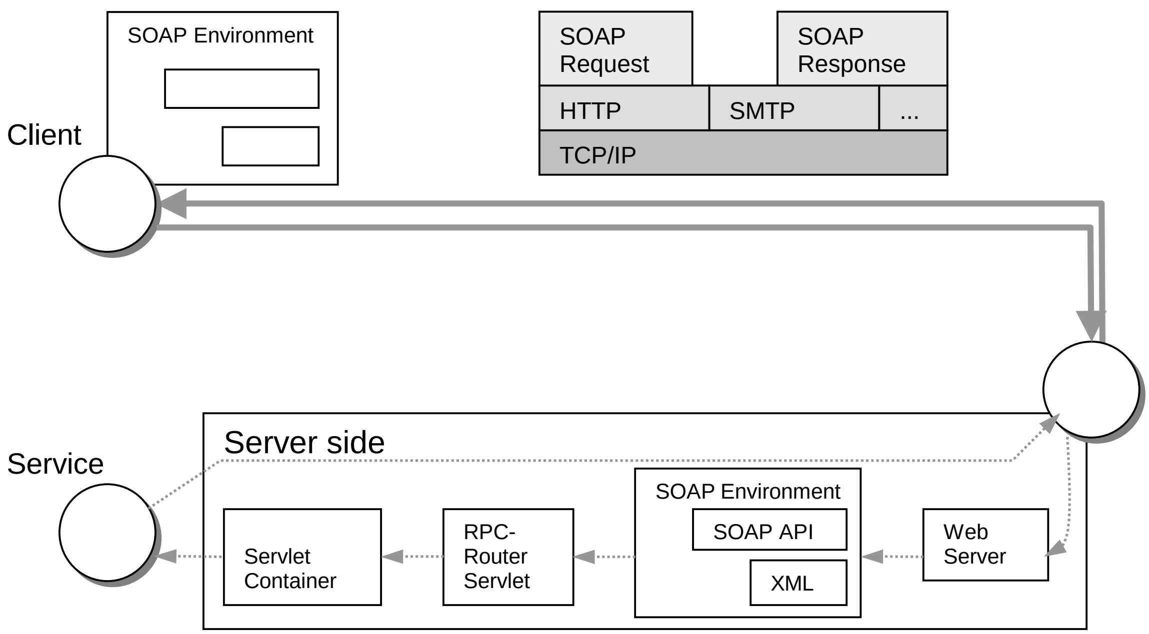

Figure 1: Service Request Processing.

5 Implementation

5.1 Architecture

Workflow functionality encapsulation and access to web services is implemented in a specific WSDL bridge attached to the ASTAI(R) system [ThFo96]. It can be used in either direction: on the one hand WSDL descriptions can embedded into workflows as activities accessing web-services, on the other hand workflows can be wrapped by WSDL descriptions. The former is realized via ASTAI(R)'s CFI's ITC [CFI ITC] compliant broadcasting bus which is mainly intended for communication between the internal services. For executing tools and applications, so called tool launchers are used. These tool launchers are triggered by events delivered through the broadcaster and may be also distributed over different types of nodes within the network.

5.2 Service Binding and Retrieval

Figure 2 shows the scenario starting with service retrieval using the UDDI registry, creating the workflow on the server, processing the workflow (e.g. response generated by the workflow system) and finally sending the automatic reply once the entire run has been completed.

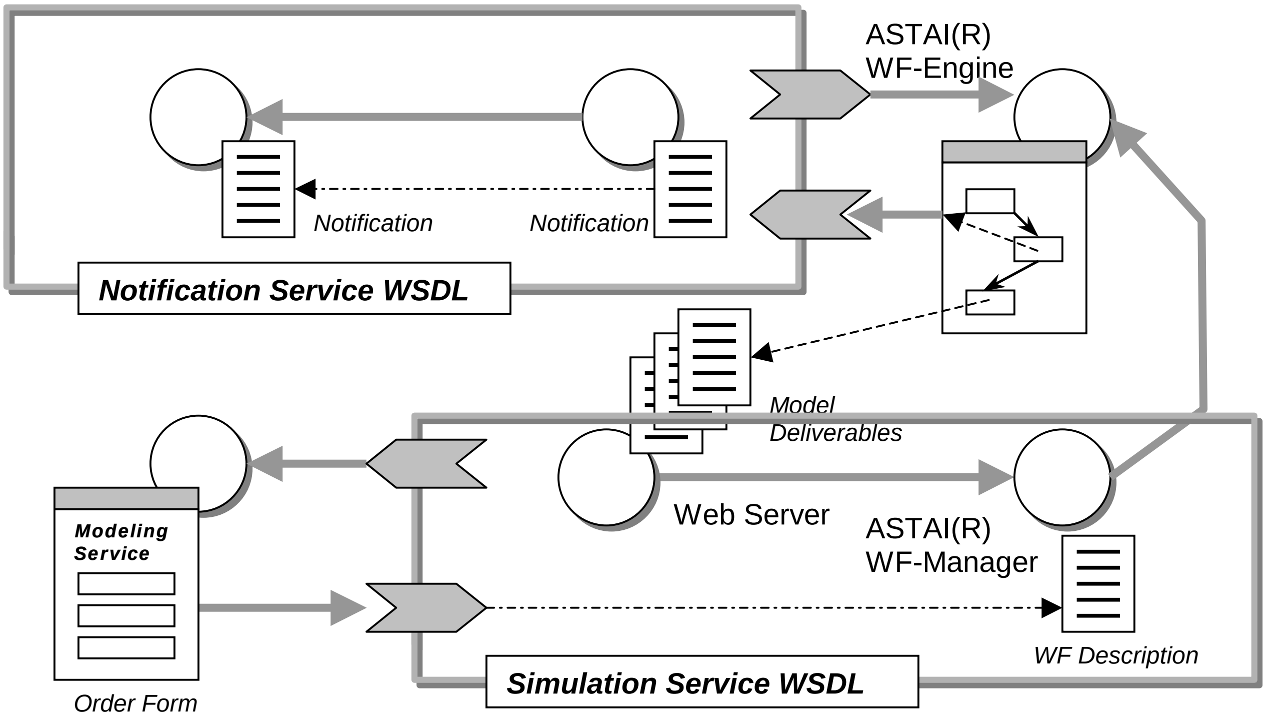

Figure 2: Simulation and Modeling Service as Web Service.

6 Comparison

6.1 Tool-Integration with TES

CFI's *tool encapsulation specification (TES)* supplies the building part for an integration. TES-files contain information about tool-invocation, data-dependencies, and command line structure. They even allow context-sensitive specification of tool I/O behavior and I/O transformations. TES is quite capable to handle tools even in complex integration situations where some input or output processing is required, but it does not focus on web integration.

6.2 Web-Integration with WSDL

WSDL provides a means for the web based integration of software resources thus enabling the set-up of highly distributed engineering environments. No aspects of re-usability, extensibility and versioning can be handled by TES so that native support for WSDL is an essential value-added for the integration logic of workflow systems. Finally, the XML approach is also viable for accessing services from and out of secured environments. There is no support for similar concepts within TES.

6.3 Comparison

The following table compares the TES-based and the XML/WSDL-based integration approaches according to the previously sketched criteria.

	WSDL	**TES**
Resources	Services, (humans through services)	Tools, (humans through tools)
Extensibility	Through dynamic port binding,	Embedded extensions

	type inheritance	(LISP)
Reuse	Reuse on different levels through schema definitions value, message, binding types	Reuse supported but limited for platform / application specific extensions
Distribution & Deployment	Meta-data based retrieval via UDDI	File name based retrieval
Versioning	Supported through Meta-data approach	Unsupported (only via encoding of tool name)
Internet Capability	Completely web-based (URL based naming and scoping for types and messages)	Provision through web possible, but not inherently built-in
Dissemination & Use	Significant development effort for wide-spread proliferation; yet, little support in EDA.	Rather limited support even in EDA business.

7 Summary and Future Work

We have described how to make XML-based markup available to an engineering workflow system in order to facilitate web-based collaboration. The integrable dispersed (web-)resources were described in WSDL and registered with and retrieved by a central UDDI directory. WSDL was shown to complement TES with respect to web-integration. The lack of standards for providing interfaces is a major obstacle for establishing collaborative engineering. Since WSDL receives increasing acceptance, it can help to fill this gap. Future work will focus on checking the applicability of this approach for a completely web-based system design as described in [Ra00] and to align this with the XML binding of WfMC [WfMC-XML].

8 References

[Ap99] Appelt, W. (1999) *WWW Based Collaboration with the BSCW System*. In Proceedings of SOFSEM'99, Springer Lecture Notes in Computer Science 1725, p.66-78; November 26 - December 4, Milovy (Czech Republic).

[BrLa00] F. Brglez, H. Lavana, et. al: *OpenExperiment: A Configurable Environment for Collaborative Experimental Design and Execution on the Internet*. Technical Report, NCSU, March 2000, Raleigh, NC, USA.

[ChSp98] Francis L. Chan, Mark D. Spiller, A. Richard Newton: *WELD - An Environment for Web-based Electronic Design*. 35. DAC 1998, 146-151, San Francisco, California, USA

[EDCS] EDCS Design Management Cluster, *http://www.sei.cmu.edu/community/edcs*

[CFI ITC] CFI; Inter-Tool Communication Programming Interface; CFI document no. dis-92-S-3, CAD Framework Initiative Inc., 1992, Austin, USA

[CFI TES] CFI; Tool Encapsulation Specification; Version 1.0.0, CAD Framework Initiative Inc., 1992, Austin, USA

[GaPo99] Gammack, J., and Poon, S. (1999) *Communication Media for Supporting Engineering*. In Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, Maui, HI, 1999.

[LaBr00] H. Lavana, F. Brglez, R. Reese, G. Konduri, and A Chandrakasan. OpenDesign: An Open User-Configurable Project Environment for Collaborative Design and Execution on the Internet, ICCD, 2000.

[LaKh97] Lavana H., Khetawat A., Brglez F., Kozminski K.: Executable Workflows: A Paradigm for Collaborative Design on the Internet, 34th Design Automation Conference, 9-13 June 1997, CA, US.

[PaFo99] Patrikalis, N.M., Fortier, P.J., Ioannidis, Y.E., Nikolaou, C.N., Robinson, A.R., Rossignac, J.R., and Abrams, S. (1999) *Distributed Information and Computation in Scientific and Engineering Environments.* In D-LIB Magazine, Vol. 5, No. 4, April 1999, ISSN 1082-9873.

[Ra00] Rammig F.: Web-based System Design with Components Off The Shelf (COTS), Forum on Design Languages, Tuebingen, Sept. 2000.

[SvB93] D. Schefström and G. van den Broek (Editors). *Tool Integration - Environments and Frameworks.* Wiley & Sons. 1993. ISBN 0-471-93554-9.

[ThFo96] W. Thronicke, W. Fox, et al.. From Tool Integration to Workflow Management - A Lean Integration Solution. In *Proc. 2nd World Conference on Integrated Design and Process Technolog*y, Austin, TX, December 1996.

[WfMC] The Workflow Management Coalition. *http://www.wfmc.org.* 2001.

[WfMC-XML] The Workflow Management Coalition. *Workflow Standard - Interoperability Wf-XML Binding.* Document Number WFMC-TC-1023. Version 1.0. 2000.