Interoperability testing of Web Services for e-Learning[*]

Dr. Andrea Polini
Istituto di Scienza e Tecnologie dell'Informazione – "Alessandro Faedo"
Consiglio Nazionale delle Ricerche (ISTI/CNR),
Via Moruzzi 1, I-56124 Pisa

Tel. +39 050 3153467

Fax +39 050 3142924

e-mail: andrea.polini@isti.cnr.it

[*] This work has been funded by the European Project "Technology Enhanced Learning Conformance - European Requirements and Testing (TELCERT – FP6 STREP 507128)".

Interoperability testing of Web Services for e-Learning

Keywords: Web Services, Interoperability Testing,

Zusammenfassung

Web-Services-Technologien eröffnen neue, interessante Szenarien im e-Learning Bereich. Im Besonderen ist die Möglichkeit der Definition und Standardisierung von Services attraktiv, welche implementiert und komponiert eine komplexe, integrierte e-Learning Applikation ergeben.
Dieser Artikel präsentiert einen Ansatz für die Konformitätsverifikation von Services, welche das Risiko von Interoperabilitätsproblemen während der Laufzeit reduziert.

Summary

Web Services technologies seem to open interesting unexplored scenarios in the e-Learning domain. Particularly appealing is the possibility of defining and standardizing Services that when implemented and composed, will result in complex e-learning applications. However an agreed standard specification per se is not sufficient, rather methodologies for validating conformance against it must also be released.
This paper discusses current approaches standardise Web Service specification and implementation and outlines an approach to conformance verification of services through testing, aiming at reducing the risk of interoperability issues at execution time.

Introduction

Web Services is today one of the hottest technologies putting in practice the Service Oriented Architecture (SOA) vision. Following such paradigm, complex applications are developed from the integration and cooperation of smaller software units exposing specific interfaces and providing specific services. Integration of services can also happen at run-time, thanks to a specific service (foreseen in the SOA definition) acting as a directory that stores and returns references to registered services. The necessity of such a distributed technology became evident when, given the high connectivity acquired even by small enterprises, people started to think and imagine to applications that could emerge from the dynamic integration and cooperation of software units running on different machines and generally belonging to different organizations. The best answer to such need has been the development and the standardization of a middleware layer mainly based on Web related technologies. In that context the World Wide Web Consortium (W3C) has naturally assumed the role of a *super-partes* standardisation body and released, in the last years, several specifications defining the different technologies necessary to set up a Web Service environment. In Booth et al. (2004) the W3C provides the following definition for Web Service:

> *A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web Service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialisation in conjunction with other Web-related standards.*

The above definition lists which are the building blocks of this new technology based on the eXtensible Mark-Up Language (XML). In fact, both the Web Service Definition Language (WSDL), that it is the language used to provide a syntactic description of a service, and the Simple Object Access Protocol (SOAP), that is the protocol used for message exchange between two services, are based on a XML format. Finally, the possibility of dynamically binding services at run time is realised thanks to the introduction of the already metioned directory service. For such purpose, a standard definition is developed and maintained by the OASIS Consortium that recently released the version 3.0 of the Universal Description, Discovery and Integration (UDDI) service (Clement et al. (2004)). The latter acts as a repository in which the provider of a service can register, among other information, a WSDL definition, or a reference to it, that later can be retrieved by a requesting service. Many other related specifications are currently under their way to reach the status of a standard. Among them, particularly relevant are languages permitting to express the cooperation and coordination that should take place at run-time among integrated services (see in particular specifications such as WS-CDL and BPEL4WS). Summarizing, the technology just described, thanks to the widely acceptance of standards and the use of web related protocols, greatly reduces the complexity of setting a distributed application in a heterogeneous and "multi-owner" environment.

Given its inherent distributed nature, the e-Learning community is strongly receptive to the advent of technologies making the cooperation of remotely connected applications easier. It is not difficult to imagine the influence and benefits that the Web Service technology can bring to this domain. Consider for instance a set of cooperating services that collectively provide the functionality of a distributed Learning Management System (LMS), for instance permitting to combine learning objects stored on different repositories and owned by different organizations; or imagine an authoring tool that interacts with a repository storing and retrieving learning objects using a web service interface, drastically reducing the complexity of setting and managing a remote interaction (see Vossen, G., Westerkamp, P. (2003) or Pankratius, V., Sandel, O., Stucky, W. (2004) for a wider discussion on the application of WSs in the e-Learning domain). The relevance of such an approach is also testified by the emergence of different European projects with a specific focus on Web Services for e-Learning (see for instance the TELCERT project, Riley et al. (2004)).

The following section presents new opportunities and relative implications, in particular for what concerns interoperability issues, that will derive from the introduction of Web Services in the e-Learning domain. Successively the paper discusses a possible approach to mitigate the risk of run-time failures due to different interpretations of standards, assumed by different developers of cooperating services. Finally a conclusions section discusses the approach and what remains to be done.

New Scenarios and new issues

The e-Learning process is certainly characterized by the necessity of exchanging many different kinds of information among many different stakeholders. So far, the complexity resulting from such a scenario at the software level has been generally managed by a single organization, or sometimes a group of connected organisations, each one developing a different platform for the delivery of e-learning contents, resulting in a plethora of huge software systems. In each platform many different tools can be identified but really seldom a tool developed within a platform is able to interact with another tool developed within a different platform. Obviously the consequence of this scenario has been detrimental for real take off of the e-learning domain, often making the production of an e-learning content product not affordable, and generally requiring a drastic reduction of the ambitions on the kind and complexity of the resulting artefact. Consider for instance a content developer; following the above depicted scenario he/she was generally obliged to produce a different version of a course for each different targeted platform, with obvious consequences for what concerns the resulting costs. Vice versa, developing the contents only considering one single platform would result in a drastic reduction on the number of possible customers and so on the budget that can be devoted to the project. Summarizing the lack of interoperability among e-Learning tools and artefacts strongly hinders the creation of an open market in which each stakeholder does not have to be extremely tightened to another (as for instance the provider of the platform for a content developer).

Potential interaction problems among different tools can be characterised at two different levels: data and message level. The first level refers to the necessity of standardizing the data structure used to deliver an e-learning content. The second level, instead, requires an agreement on the interfaces and messages that must be used to correctly interact.

Different standard bodies operating in the e-learning domain have already started to release standard specifications for the first level (data) solving related interaction issues. Instead, Web Service technologies promise to be the right key that will permit to enter an ideal domain in which interactions, considering the message level, are not an issue anymore, with substantial benefits in terms of costs and capability to handle complexity. Specifications for the message level are currently under their way, in particular by the IMS global consortium (IMS (2005)).

However, it is important to note that the release of a standard *per se* does not solve the problem. Contextually with the release of the standard we need a way to assess that an implementation actually conforms to what is specified in the standard, otherwise problems could arise even among two implementations, each one claiming to be conformant. In any case it is highly recommended that the verification of conformance is carried on by a third trustable party, as already happens in different domains.

Currently a Web Service is described trough the definition of a WSDL file. Such description is rather little informative, only defining at a syntactic level the service offered. Obviously such way of describing a service leaves little room to the automatic verification that an implementation of a service is actually conform to the specification defined within a standard. In fact conformance can only be ascertained with regard to a syntactic verification, made by a sort of compiler, that assesses that the service actually implements the specified interface. Obviously this cannot give any guarantee on the interoperability of the resulting software, since different assumptions made by the interacting services could lead to incompatible behaviour. In alternative, behaviour conformance could be addressed on the base of a set of tests defined by experts of the particular domain and obtained after a deep analysis of a particular specification. However, this approach suffers of two main drawbacks. The first is obviously inherent to the nature of any human process, that could produce errors in the different phases of the derivation of the tests. As a result the assessment of conformance on such bases cannot provide enough guarantees on the interactions that will happen among two conformant software at run time. The second drawback, probably the hardest, is the extreme high cost that a human based process will absorb. Moreover, considering the high number of possible services that can be identified in the e-learning domain and should then go through a standardisation process, the emergence of an automatic approach to the verification of conformance becomes even clearer.

Summarizing we need a way to produce standard specifications for Web Services which include some kind of semantic definition of the service behaviour, to allow for the automatic derivation and execution of test cases for conformance verification. An approach that is currently under development aiming a solving these issues is described in the following section.

Interoperability testing from Augmented WS specifications

Currently there exist many different approaches with awkward names such as WSCI (Arkin, A. et al. (2002)), or BPEL4WS (Andrews, T. et al. (2003)) to augment the definition of a WS and of admissible interactions. The target of these approaches has mainly been to identify which kind of information could be added to a Web Service interface specification to increase the semantic description of the service, as just discussed. A primary constraint in such identification has been the necessity of leaving out of the description all the information that would directly relate to a specific implementation. Moreover another important choice to take has been the instrument to use in order to express such additional information.

A first idea in this direction has been to reuse concepts and methodologies developed in the area of Software Engineering, in particular for component-based software, a research area that shares many similarities with the WS domain. Hence one natural choice would be to augment the definition of each method in the WS interface with a list of pre- and post-conditions, i.e., predicates that define what must be guaranteed before invoking a service and what a client should expect when the invocation returns. Another particularly important kind of information is the specification of the ordering of invocations that a service accepts as correct, and assumes will be followed by a client. For instance, in an airline reservation service a confirmation for a reservation can be correctly accepted only after a reservation request has been issued, providing information such as passenger personal data. Once the information to be provided with a service and aiming at augmenting the semantic description of it, have been chosen, the second necessary step is to identify the instruments to use in order to express such information. Taking this decision two factors have to be considered fundamental. First of all we need a language that can be "unambiguously" and easily interpreted by a human reader that possibly wants to implement the specification. The second factor instead concern the opportunity of adopting a widely accepted language, possibly increasing the chance of reusing already developed tools for the automatic derivation of test cases from the specifications. Given that, the most attractive choice seemed to be the Unified Modelling Language (UML). This language in fact has attracted in the last years a lot of interest from all the different software communities. The graphical basis of this language, that providing different kind of diagrams permits to express structural and behavioural features of a software artefacts, is certainly the characteristic that has guaranteed to UML a really wide acceptance. As a consequence the usage of UML will guarantee that the provided specifications will be easily accessible and largely understandable. Finally pre- and post-conditions can be expressed, directly within UML, using the Object Constraint Language (OCL) that is a first order logic language that permits to directly refer within the conditions to elements specified within UML diagrams.

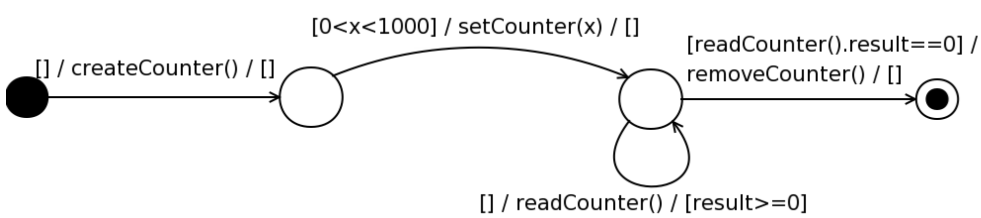

Figure 1: A simple Abstract Machine, annotated with pre and post conditions, for a simple counter

Another important advantage is that the availability of a UML model permits in principle the automatic derivation of a set of test cases. In the literature a lot of different approaches for the derivation of test cases from models, can be found. However in the general case they are applied starting from formal specifications languages that result to be rather simple with respect to real systems. Currently the research has been focused on how to adapt the theory of conformance testing (see Tretmans, J. (1996)) to the case of PSM (UML (2004)), that is the particular kind of UML diagram that has been chosen to augment WS descriptions. In Figure 1 there is an example of an abstract machine that provides an idea on the kind of model that can be used to automatically derive test cases. The very general idea is that test cases will be derived from paths on nodes and arcs of a defined machine. Moreover the selection and generation of the tests will be influenced by the expressions found on pre- and post-conditions that also express constraints of the range of variability of the input parameters and that in the example are used as labels on the arcs. The final goal of this research is to encapsulate the derived theory in a tool that will permit the derivation of the necessary test cases. The availability of such a tool will also permit the derivation and execution of test cases on the fly, i.e. concurrently with system execution. Such a tool could be inserted in the general framework for Web Services testing described in Bertolino and Polini (2005).

Conclusions

After outlining the emergence of Web Service technology, the potential relevance of such a technology in the e-Learning domain has been highlighted. In particular it has been argued the possibility of having standard specifications increasing the interoperability between tools, even when they are produced by different vendors. However to have adequate guarantees of correct interactions among different services, standards are not enough, instead it is also necessary to test the implementation of the standard against the specification. Currently the scarce documentation model defined for Web Services description hinders the real possibility of deriving meaningful test cases to be used for conformance verification purposes. The paper proposes a way to increase the semantic information attached to a WS description, suggesting the usage of a particular UML diagram called PSM, and describes a methodology for the automatic derivation of test cases from the defined specification diagrams. A tool that applies the proposed methodology is currently under development.

References

Alonso, G., Casati, F., Kuno, H., Machiraju, V. (2004). *Web Services – Concepts, Architectures and Applications*, Springer-Verlag.

Andrews, T. et al. (2003) *Business Process Execution Language for Web Services – version 1.1*, retrieved September 29, 2005 from http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

Arkin, A. et al. (2002) *Web Service Choreography Interface (WSCI) 1.0*, retrieved September 29, 2005 from http://www.w3.org/TR/wsci/

Bertolino, A., Polini, A. (2005). *The Audition Framework for testing Web Service Interoperability*. Proceeding of 31st Euromicro Conference, Porto, Portugal. IEEE Computer Society. pp. 134-142

Booth, D. et al. (2004) *Web Services Architecture*, retrieved September 29, 2005 from http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

Clement, L. et al. (2004) *Universal Description Discovery & Integration (UDDI) ver. 3.0* retrieved September 29, 2005 from http://uddi.org/pubs/uddi v3.htm

IMS (2005) *IMS global Learning Consortium*, retrieved September 29, 2005 from http://www.imsglobal.org/

Pankratius, V., Sandel, O., Stucky, W. (2004) *Retrieving Content with Agents in Web Service E-Learning Systems*. Proceeding of the Symposium on Professional Practice in AI, 1st IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), Toulouse, France, August 2004, pp. 91-100.

Riley et al. (2004) *TELCERT: Technology Enhanced Learning Conformance - European Requirements and Testing*, retrieved September 29, 2005 from http://www.opengroup.org/telcert/

Tretmans, J. (1996) *Test generation with inputs, outputs and repetitive quiescence*. In Software| Concepts and Tools , 17(3):103-120.

UML (2004). *UML 2.0 Superstructure Specification,* ptc/04-10-02 October 2, 2004. retrieved September 29, 2005 from http://www.omg.org/

Vossen, G., Westerkamp, P. (2003). E-Learning as a Web Service. *7th International Database Engineering and Applications Symposium* (IDEAS 2003), Hong Kong, China. IEEE Computer Society, pp. 242-249.