

Web Services and Service-Oriented Architectures

Gustavo Alonso¹, Fabio Casati²

¹*ETH Zurich, Switzerland*

Email: alonso@inf.ethz.ch

²*Hewlett-Packard, Palo Alto, CA, USA*

Email: fabio.casati@hp.com

Web services, and more in general service-oriented architectures (SOAs), are emerging as the technologies and architectures of choice for implementing distributed systems and performing application integration within and across companies' boundaries. In this tutorial we describe Web services from an evolutionary perspective, with an emphasis on their utilization for enterprise application integration and service-oriented architectures. The tutorial covers basic middleware problems and shows how the solutions to these problems have finally evolved into what we call today Web services.

The first part of the tutorial is intended to put Web services in the right perspective, particularly in terms of what can be and what cannot be done today. This is an important aspect of the tutorial as the almost daily appearance of new specifications and self-proclaimed standards has led to a very confusing set of ideas around Web services. Some of this confusion affects the very basic model behind Web services and it is therefore important to dispel these myths. For instance, SOAP and WSDL have been designed under the assumption that service interfaces will describe the service themselves. In practice, this is almost never the case. On the one hand, WSDL may lack some important features (as witnessed by the apparition of the G-WSDL specification for grid services) and SOAP might be used more with proprietary bindings than with standard bindings. On the other hand, binary attachments and the need for complex business protocols and conversation controllers indicate that Web service descriptions might be more geared towards describing low level access to the service than to describing the service and its usage.

The tutorial then focuses on two of the most important and innovative aspects of Web services: business protocols and service composition.

Business protocols define which message exchange sequences are supported by the service. They complement the traditional IDL-like (or, in Web services, WSDL-like) interface definition by specifying constraints on the order in which service operations should be invoked.

Business protocols are an important aspect of a service specification, as it rarely happens that service operations can be invoked at will independently from one another. On the contrary, the interaction between clients and

services is always structured in terms of a set of operation invocations, whose order typically has to obey certain constraints for clients to be able to obtain the service they need. The tutorial discusses the need for business protocols and the opportunities they bring, and presents different approaches to protocol modeling in terms of languages, formalisms, and expressive power. It also stresses the need for a protocol algebra and shows how protocol modeling and protocol algebras can together form the basis for supporting and automating many aspects of service development and execution. Due to its importance, protocol models and languages have been the subject of standardization efforts. While presenting different approaches to protocol modeling, the tutorial also discusses and compares standardization proposals.

The second aspect detailed in the tutorial is that of service composition. When services are described and interact in a standardized manner, the task of developing complex services by composing other (basic or composite) services is considerably simplified with respect to conventional middleware, where composition technologies (such as workflow technology) were available but failed to become widely adopted. Indeed, as SOA-related technologies mature, service composition is expected to play a bigger and bigger role in service development. Mirroring the discussion about business protocols, the tutorial motivates the need and opportunity for service composition models and technologies, compares different approaches, and analyzes standardization proposals.

Finally, we discuss how protocols and composition, respectively representing the external specification and the internal implementation of a service, are related to each other and how this relationship can be used to further support service development.

The basic principles underlying the concepts presented in this tutorial are derived from a book co-authored by the presenters [1].

References

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. Web services: concepts, architectures, and applications. Springer Verlag, 2004.