Editorial

Esperanza Marcos

Kybele Research Group
Department of Software and Computing Systems
Rey Juan Carlos University
Tulipan S/N Street, 28933, Móstoles, Madrid, Spain
E-mail: esperanza.marcos@urjc.es

Klaus Dittrich

Database Technology Research Group
University of Zurich
Department of Informatics
Winterthurerstrasse 190
CH-8057 Zurich, Switzerland
E-mail: dittrich@ifi.unizh.ch

Biographical notes: Esperanza Marcos is Associate Professor at the Department of Software and Computing Systems, Rey Juan Carlos University, Madrid. She is Chair of the Kybele Research Group which researches on Information System Engineering and focuses on topics like Model Driven Development, Web Information Systems, Philosophical Principles of Information Systems, *etc.* She received her PhD in Computer Science and MSc from the Polytechnic University of Madrid. She has lectured in various high-level courses and masters. Currently, she is collaborating in the 'Software Engineering Master' of the Polytechnic University of Madrid and the 'Decision Support Engineering Master' of the Rey Juan Carlos University. She is the coauthor of numerous book chapters and papers in national and international publications. She has directed and participated in many research projects.

Klaus R. Dittrich is a Professor and Director at the Department of Informatics at the University of Zurich, Switzerland, and heads the database technology research group at this place. His main areas of interest include architectural issues in database management systems and their environment, system integration issues and advanced DBMS functionality. He has lectured and published widely and has been a visiting scholar at places like IBM Research, HP Labs, Boeing and the universities of Stanford, Milano, Aalborg and Nice-Sophia Antipolis. He has been and still is engaged in various national and international cooperation projects and regularly consults with industries and government agencies. Among other professional activities, he is the former President of SI (Swiss Informatics Society), board member of ICTswitzerland and coeditor-in-chief of the *Very Large Data Base Journal*, the leading scientific journal in its area. His doctoral degree is from the University of Karlsruhe, Germany.

Web services have emerged as a new technology expected to have a tremendous impact on distributed applications and their integration. In recent years, many conferences, books and special journal issues (like this one) related to web services have appeared.

Various definitions of what a web service is can be found in the literature; for example:

- 'Self-contained, modular business applications that have open, internet-oriented, standards-based interfaces' (UDDI Consortium, 2001).
- 'A software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artefacts. A web service supports direct interactions with other software agents using XML-based messages exchanged via internet-based protocols' (W3C, 2002).
- 'Services are self-describing, open components that support rapid, low-cost composition of distributed applications' (Papazoglou and Georgapoulos, 2003).
- 'A web service is an interface that describes a collection of operations that are network-accessible through standardised XML messaging. A web service performs a specific task or a set of tasks' (Gottschalk *et al.*, 2002).
- 'Web services are applications that can be published, located, and invoked across the internet. Web services may use other services in order to perform their task' (Gunzer, 2002).

Despite their differences, these definitions have many similarities and common terms, like *open, internet-based, distributed applications, standardisation, self-contained and XML messages*. This is because, in spite of their different points of view, these definitions fortunately share the same underlying concepts and technologies.

While it has been difficult to agree on a single unified and accepted definition for a web service, there is no doubt that this new technology could mean changes for distributed applications and integration. Although web services can be seen as another type of middleware (like CORBA, RMI or DCOM), this middleware seems to be more suitable for the internet and the web (Gunzer, 2002). The web is a heterogeneous environment for both the server and the client. It is not possible to know in advance what kind of middleware each side of the connection uses. A new approach to middleware for this kind of distributed applications is needed, and many experts believe web services could be a solution.

Moreover, conventionally distributed middleware usually relies on just one company. Centralised middleware occupies an intermediate place in the application integration process and mediates interactions (Alonso *et al.*, 2004). This kind of middleware also poses some limitations for *Business-to-Business* (B2B) applications, where the customer and the supplier have to agree to a common middleware platform. However, this is not always possible because each company wants to preserve autonomy. Web services allow a key change, moving the integration scenario from pre-defined architectures to application integration.

As a result, web services technology and underlying architectures (*e.g.*, Service Oriented Architecture or SOA) have the following advantages:

- Distribution – They allow service execution at any point in the network.
- Reusing – They are deployed in the network and thus can be used by any application.

- Integration – The human factor is removed from the integration process because web services are accessed by programs. Moreover, web services enable dynamic and scalable cooperation between independently developed systems and organisations. Thus, web services technology appears to be the solution for B2B integration and Enterprise Application Integration (EAI).

The potential benefits of web services have led to the establishment of an important class of research activities, both in industry and academia. This research draws on a variety of fields such as automated software engineering, process modelling, semantic web and workflow. The variety of research topics can be divided into two groups: those related to *technology aspects* (*e.g.*, infrastructures, languages, architectures) and those related to *engineering ones* (*e.g.*, automatic development of web services, automatic application development with web services including orchestration and choreography). There are also some orthogonal topics like web services security and semantics.

This special issue broaches recent challenges and advances in Web Services Engineering and Technology (WSET). It contains theoretical articles as well as some which present more practical approaches. Of special interest are three papers that were judged (by an independent jury) to be the best of the *Spanish National Workshop on Web Services* (JSWEB)[1], held in 2005. The main objective of this conference was for academia and industry to share practical experiences in web services. The rest of the papers present results of several well-known researchers in web services. We thank all of them for kindly accepting our invitation to contribute to this special issue.

The first paper, by Gutiérrez *et al.*, constitutes an application of the research results of the Alarcos group (*Castilla-La Mancha University*) on web services security to a company application (*i.e.*, Spanish State Lottery). The paper, 'Development of web services security systems', presents the problems found when applying a Process for Web Services Security (called PWSSec) developed for the migration of a desk application to one that is web based. Through this case study, based on a real application, the authors explain the problems and the solutions for each case.

Ortiz *et al.*, from *Extremadura University* (Spain), are the authors of the second paper, 'Preparing web services for choreographs'. In contrast to most research on web services composition which focuses mainly on orchestration, this paper deals with the problem of implementing choreographies. Whereas in orchestration, a service manages interaction flow, in choreography, the control of the composition is distributed. This paper approaches the problem by integrating the SOAP header information into aspect-oriented techniques.

In contrast, the third paper, 'Service composition modelling: a case study', by de Castro *et al.*, from *Rey Juan Carlos University* (Spain), offers an engineering approach to web services modelling and composition. The authors present their experience in applying a model-driven method for service composition modelling in a real case: a Web Information System (WIS) for medical image management and processing. The method starts by identifying the services required by the user of the WIS (*e.g.*, getting diagnostic information on a human brain). Afterwards, it proposes some guidelines to be able to map these services into web services. The method also allows obtaining a service composition model that facilitates the mapping to a specific web services composition language. Perhaps, the main contribution of this work is that it proposes a service-oriented approach from the first steps in the development process all the way to the implementation by using web services.

Another engineering approach to web services application development is presented by Fraternali *et al.*, from the *Politecnico Milano* (Italy) in their paper 'A CASE tool for modelling and automatically generating web service-enabled applications'. The CASE tool is based on WebML, a modelling language for web applications. This paper describes the new primitives added to WebML to allow web services modelling, the architecture of the CAISE tool, as well as the architecture of the generated applications. The CASE tool allows the automatic generation of the code of a web application, including web services calls and the exposed web services. An important part of this work is the use of the CASE tool for code generation in some industrial applications.

In the fifth article, 'Message mediation in composite web services', current work of Bussler, of the Digital Enterprise Research Institute (DERI), Galway (Ireland), is presented. Web services communicate with each other by message passing according to the input and output schema defined in the interface. Each web service has its own message definition schema; this poses a semantic problem that complicates understanding between different web services. The authors examine the problem of message mediation in the context of composite web services. They propose a methodology for including message mediation in composite web services definitions.

Biörnstad *et al.*, ETH Zurich (Suiza), are the authors of the sixth paper, 'Enforcing web services business protocols at run-time: a process-driven approach'. Business processes provide abstractions for modelling business protocols that define the correct interactions between two or more web services. In this paper, the authors devise a more technical approach to web service interaction and propose employing process-based tools to ensure that the messages exchanged between different web services comply with a given business protocol, both in terms of sequencing constraints and data flow characteristics. They also show, with experimental results, how their solution simplifies the implementation of web services because it helps separate the concern of business protocol compliance from the actual service implementation.

This special issue ends with a paper by Papazoglou and van den Heuvel, from Tilburg University (The Netherlands), 'Service-oriented design and development methodology'. According to these authors, since SOA is based on open standards and is frequently realised using web services, developing meaningful web service and business process specifications is an important requirement for SOA applications that leverage web services. Therefore, developers need methodologies that help them manage complex service-oriented development projects. The paper provides an overview of the methods and techniques used in service-oriented design and development. Moreover, it examines their own service development methodology from the point of view of both service producers and requesters and reviews the range of elements in this development methodology available to them.

In the first paragraphs of this editorial, we spoke about the different definitions of web services. Keep these differences in mind when reading this issue. In our opinion, they are mainly based on the point of view of each individual author writing about web services (implementation, languages, architectures, *etc.*).

As you will recall, we grouped the research topics on web services into two categories: *technological problems* and *engineering problems*. The second and the sixth paper fall into the former category, and the third, fourth and seventh into the latter. The first paper addresses the security problem from an engineering point of view and the fifth deals with a semantic problem from a technical point of view.

Enjoy!

References

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

Gottschalk, K., Graham, S., Kreger, H. and Snell, J. (2002) 'Introduction to web services architecture', *IBM Systems Journal*, Vol. 41, No. 2, pp.170–176.

Gunzer, H. (2002) 'Introduction to web services', *Borland White paper*, http://www.borland.com (retrieved on 20 October 2005).

Papazoglou, M.P. and Georgapoulos (2003) 'Service-oriented computing', *Communications of the ACM*, October, Vol. 46, No. 10, pp.25–28.

UDDI Consortium (2001) *UDDI Executive with Paper*, http://uddi.org/pubs/UDDI_Executive: With_Paper.pdf (retrieved on 20 October 2005).

W3C (2002) *Web Services Architectures Requirements*, http://www.w3.org/TR/wsa-reqs (retrieved on 20 October 2005).

Note

1 Spanish national workshop on Web Services (Jornadas Científico-Técnicas de Servicios Web, JSWEB). Organised by W3C, Andalusia Government and Rey Juan Carlos University. Granada (Spain), 13–14 September 2005, http://www.w3c.es/Eventos/ServiciosWeb/.