

Service-Oriented Computing - Introduction to the Special Theme

by Jana Koehler and Gustavo Alonso

Service-oriented computing is an emerging cross-disciplinary paradigm for distributed computing, which is changing the way software applications are designed, delivered and consumed. At the heart of service-oriented computing are services that provide autonomous, platform-independent, computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols to build networks of collaborating applications distributed within and across organizational boundaries.

Grid services and Web services are currently the most common forms of service for implementing service-oriented computing. Grid services provide the foundation for the distributed execution of long-running scientific computations over very large data sets using a standardized and stateful service interface. Web services provide the basis for the development and execution of business processes that are distributed over the network and available via standard interfaces and protocols. Technically, these two types of service have converged to a large extent in recent years, but the difference in application focus leads to a variety of complementing research questions.

The state of the art in service-oriented computing is characterized by the very early adoption of many forms of technology by the IT industry. In particular, in the area of Web services, large IT vendors are leading and driving many research activities. At the same time, many customers have begun to use Web services. However, it is a challenging task for many corporate IT organizations to migrate an enterprise towards a service-oriented architecture. Adopting service-oriented computing (or a service-oriented architecture as it is often called in the enterprise computing space) is a difficult endeavour given the fact that much of the required technology is at an early stage of maturity. Problems can be located in the usability of today's software tools for developing service-oriented applications, the incompleteness of the so-called Web services stack with its many standards, and the need to further mature many of those standards. Key standards are the Web Services Description Language (WSDL), the Business Process Execution Language (BPEL), and the Simple Object Access Protocol (SOAP). In addition, many WS-* standards (for example, WS-Addressing, WS-Policy, WS-Reliability, WS-Security) are addressing non-functional aspects of Web services. Semantic Web standards such as RDF and OWL have been adopted by the first commercial tools and used to extend Web services with annotations that describe more precisely what functionality a service provides.

Besides this, numerous problems relating to user-friendliness and the maintenance of the implemented IT solution artefacts must be overcome. For example, the problem of ensuring the correctness and completeness of business process solutions using service-oriented computing principles has been only partially solved. Similarly, it is very difficult to quantitatively define what constitutes a good service-oriented architecture and how to determine the right granularity

of the designed services in order to facilitate reuse. This is what service-oriented computing is about the next evolution in the reuse of software combined with principles of loose coupling and distributed computations. Yet initial experience indicates that even Web services may not be as loosely coupled as originally expected, and there is a strong trend towards Representational State Transfer (REST) and REST-oriented development with very thin intermediate layers (in addition to WSDL/SOAP).

In Grid computing, services and service-oriented computing plays a big role, and to a certain extent it was in Grid computing that some of the early ideas about service-oriented architectures were first explored. This early adopter strategy led to diverging and independent specifications for Web services in the enterprise and in Grid computing. This phase is now over and the convergence between both views of services and the related infrastructure is almost complete.

However, Grid computing poses different challenges and requirements than do enterprise applications. For instance, the amount and type of data exchanged is quite different than in enterprise applications, and the dependencies between tasks in a composition are typically stronger since often there is not just a data or control flow dependency, but also a location dependency imposed by the cost of moving data. As another example, the Web Services Resource Framework (WSRF) is a specification tailored to Grid computing to enable closer control of stateful resources (eg computations) through Web services. This specification can be used in many applications but it addresses problems that are found primarily in computing grids rather than on enterprise applications. These and other differences are substantial enough to require systems and approaches other than those encountered in enterprise computing, as the use of resources and the operations performed with and on services often involve a higher degree of control than conventional enterprise services.

Grid computing, however, still has to catch up with enterprise computing in terms of industrial strength tools and a better understanding of the development process. While a new form of software engineering is quickly evolving in the enterprise computing space, driven by the need to bring business and IT together, service composition and service architectures are still a matter of mostly ad-hoc development in Grid computing. Thus, what has been achieved so far is a leap forward over the state of the art from a few years ago. Still, much remains to be done to incorporate into Grid computing the discipline, techniques and lessons learned from software engineering. This is also true with regard to services, and will remain the main challenge in Grid computing for years to come.

The articles featured in this special edition reflect the recent trends in research on service-oriented computing, and address many of the aforementioned issues. Human collaboration is one of the strong drivers in the Grid and Web services area. It is therefore not surprising that non-functional aspects such as the quality of a service are at the heart of many research and development activities. Semantic technology and well-formalized analytical capabilities aim at improving tools to develop service-oriented applications. We are especially pleased to have several articles in this issue that illustrate the wide applicability of service-oriented computing including, for example, the shipping industry, customized shoe production, health care and satellite-based observation. This is a clear indication of the strengths and potential of service-oriented computing.

Please contact:

Gustavo Alonso

Computer Science Dept, ETH Zurich, Switzerland

E-mail: alonso inf.ethz.ch

Jana Koehler

IBM Research GmbH, Zurich Research Laboratory, Switzerland

E-mail: koe zurich.ibm.com