SECOND SPECIAL ISSUE ON SERVICE-ORIENTED COMPUTING

GUEST EDITORS' INTRODUCTION

VINCENZO D'ANDREA* and MARCO AIELLO†‡
Department of Information and Communication Technologies
University of Trento, Via Sommarive 14, 38100 Trento, Italy
†Presently at the Distributed Systems Group, TU Wien
Argentinierstraße 8, A-1040 Wien, Austria
*dandrea@dit.unitn.it
‡aiellom@dit.unitn.it

Service-Oriented Computing is consolidating as the solution of choice for enterprise application integration, for realizing message oriented middleware, and — more generally — for building cooperative information systems acting in open environments such as the Web. At the same time, grid-based architectures are establishing themselves as the reference model for computationally intensive problems. Today, the Service-Oriented Computing and Grid communities are converging because the ultimate goals of having platform independent, loosely-coupled, computationally powerful, networked information infrastructures are evermore shared by researchers in the two fields. One of the points of contact of the two fields is the Web Service technology. Web Services are, on the one hand, the best-known incarnation of the Service-Oriented Computing paradigm and, on the other hand, are the enabling technology of choice for many Grid architectures. Furthermore, the adoption of Web Service is growing at a steady rate, guaranteeing that services will be out there to be invoked when designing an information system or for autonomous agents living on the Web.

The optimism about the future of Service-Oriented Computing and related architectures is to focus important research efforts on the topic and open a number of passionate debates. The most appealing idea behind having a vast number of services readily available is that of composing them in order to build added valued services or information systems. Then one needs to commit to which information is provided by the individual services to facilitate the composition. Is this information purely syntactic, i.e. a list of operation names with the type definition of inputs and outputs, or is the information rather semantic, i.e. carrying general descriptions of which operations are possible and what they mean referring to some ontology and formal semantics? Another related debate is to determine when the composition does occur in the lifecycle of a cooperative information system. Is it a static process...
happening at design-time, or rather a dynamic process taking place at run-time simply by considering the requirements of the application to be executed? As it is often the case, the golden bullet may very well lie somewhere in the middle of these extreme perspectives.

Research is currently tackling such issues of service’s semantics, service composition, and more. The problem of service description concerns, besides the syntactic vs semantic debate, also the issue of what needs to be described. Service operations are of course essential, but there is more to a service than its functional properties. Quality of Service parameters define the actual usability, while policies and aspects do qualify behavioral characteristics of services which are, in turn, essential for engineering distributed applications fulfilling requester’s desiderata. Service Level Agreements need to be established among services that have recurring interactions. What is the basis for an agreement and how to negotiate them is the object of initial proposals. Finally, how to take advantage of the Grid in order to have powerful service-oriented architectures and, dually, how to provide general Grids using Service-Oriented Computing principles are topics of current investigation.

The present special issue on Service-Oriented Computing collects the extended versions of the six best papers coming from the Second International Conference on Service-Oriented Computing, held in New York at the premises of the Columbia University in November 2004, and follows the first special issue on the topic hosted in this same journal. ICSOC is an annual happening which has established itself as the reference event in the area of Service-Oriented Computing coming after the success of the inaugural conference held in Trento in 2003, and followed by the 2005 edition in Amsterdam and the 2006 edition in Chicago. The New York conference attracted around 200 researchers and various professionals from diverse fields including distributed systems, data bases, software engineering and artificial intelligence. The papers published in the present volume were updated from the conference version and went through an additional peer-review process before being granted publication. The papers provide a meaningful overview of the current main topics in Service-Oriented Computing covering service composition based on semantic descriptions, runtime monitoring of service invocations and of process execution, service capability descriptions and policies and, finally, grid services.

Fujii and Suda tackle the problem of dynamic service composition based on service descriptions which include functional, semantic and rule-based aspects. Most notably, the semantics of a service is provided in terms of conceptual graphs. The authors present the architecture and describe an implementation based on the best-known Web Service standards.

Run-time monitoring of service’s compliance with composer provided requirements is the topic of the article by Spanoudakis and Mahbub. The authors consider events and state-based information available from the system composition manager and translate it into expressions of the event calculus. Then, five possible forms of behavioral deviation of the system can be recognized, namely, inconsistencies recorded from the system behavior, inconsistencies from the expected behavior,
cases of unjustified system behavior, possible inconsistencies from the expected behavior, and possible cases of unjustified system behavior. The monitoring process is performed by comparing templates which represent different instantiations of the event calculus formulas specifying behavioral properties. Interestingly, the behavioral properties are extracted directly from a Web Service standard for expressing state-based processes (BPEL) and translated into the event calculus expressions.

A different take on service-oriented business process monitoring is considered in the paper by Aiello and Lazovik. In this approach, the requirements of service requesters and providers are translated into a plan to be executed against a state-based business process. While monitoring the execution of the process, unexpected events can occur or specific return values may need intervention to meet the original requirements. This is achieved by interleaving planning and execution. Additionally, the authors provide a classification of business rules which may be expressed by service providers. The classification considers two dimensions: the expressive power used in the assertion, on the one hand, and the role of the owner of the assertion in the execution of the business process, on the other hand.

In their paper, Dan, Dumitrescu, Ranganganthan, and Ripeanu present SLM (Service Level Manager): a framework for service management, with the goal of matching service client’s goals and a set of providers’ resources. The SLM maps client objectives to resource requirements, acquires and aggregates resources, and arbitrates the dynamic allocation of resources. The authors identify three abstraction levels for resources: the raw resource level, the abstract resource level, and the service level. The SLM is then separated into three functional layers, matching the level of abstractions of resources. In the beginning of the paper, a motivating scenario, used throughout the paper in order to provide concrete examples, is described illustrating SLM in various environments. In the final part, the authors discuss how SLM functions and systems can be combined to address new scenarios.

The application of Aspect Oriented Programming (AOP) to Web Services is the core idea of Ferraz Tomaz, Himda, and Monfort. The main goal is to address non-functional concerns of services by means of Aspects. After describing the AOP paradigm, they discuss two approaches to the use of AOP in services. The first approach is based on the notion of an intermediary, an additional message handler, extracting from messages the relevant non-functional data and then directing messages to services. The author then identify the drawbacks of this approach and motivate a second proposal, based on intercepting messages, and redirecting them to a service weaver, dealing with Aspect related operations. A detailed description of the technological issues completes the paper.

A concrete application is the subject of the last paper, where services are used to manage the allocation of resources in a large computational grid. Sandholm et al. describe how, in the Swedish Grid Network, the proposed system collects data on resources usage to allow the enforcement of resource usage policies. The authors describe the SweGrid computational resource, which provides the motivation and the test bed for the proposed system. A prominent feature of the proposed system is
the extensibility and flexibility of the architecture, allowing the adoption of several management policies. The implementation is based on the composition of primitives expressed using existing standards for grid management. The description of two simulation of the application of the system completes the paper: one of fair resource requests and one of an unfair flow of requests. After discussing the simulation, the authors present a practical implementation of the system.

This volume of the *International Journal of Cooperative Information Systems* provides a meaningful picture of the state of the art in Service-Oriented Computing and we hope you will enjoy its coverage of different key aspects of the field. We take this opportunity to express our gratitude to the authors and the reviewers who contributed to the realization of the present volume.

References
