International Journal of Cooperative Information Systems
Vol. 14, No. 4 (2005) 329–331

World Scientific
www.worldscientific.com

SPECIAL ISSUE ON SERVICE-ORIENTED COMPUTING GUEST EDITORS' INTRODUCTION

MARCO AIELLO* and VINCENZO D'ANDREA†
Department of Information and Communication Technologies, University of Trento
Via Sommarive 14, 38100 Trento, Italy
**aiellom@dit.unitn.it*
†*dandrea@dit.unitn.it*

Service-Oriented Computing is an emerging computing paradigm for building cooperative information system in which the concepts of distribution, openness, asynchronous messaging and lack of centralized control take a leading role. In this context, applications are built out of individual services that expose functionalities by publishing in appropriate repositories their interfaces, and abstracting entirely from the underlying implementation. Published interfaces may be searched by other services or users and then be invoked. The interest in Service-Oriented Computing is a consequence of the shift from a vision of a web based on the presentation of information to a vision of the web as computational infrastructure, where systems and services can interact in order to fulfill users' requests. Web services, the best-known example of this shift, are the realization of service-oriented systems based on open standards and infrastructures, thus extending the XML syntax.

There is a wide landscape of opportunities offered by Service-Oriented Computing: it is possible to create added value information systems simply by aggregating individual services; it is possible to create inter- and intra-company information systems with ease by "servicizing" existing systems; it is possible to achieve true interoperability and transparency at run-time. But Service-Oriented Computing is far from being a solution for all distributed systems problems. A number of open issues make the research in the field interesting and challenging at the same time. Devising ways to automatically or semi-automatically compose services at design time or, even better, at run-time would change the way we look at software engineering and radically increase the speed of application development. How to discover existing services in smart ways and how to consider equivalences among services is an open problem. All considerations of non-functional requirements are paramount for the user's satisfaction when interacting with services and need general and feasible solutions. Finally, there are a number of problems on how to manage the lifecycle of services, how to model individual and compositions of services.

On the one hand, the adoption of the basic web service technology is blooming; and on the other hand, the Service-Oriented community needs to prove that higher level functionalities are feasible. For instance, having a user express his/her needs in the form of a high-level request and abstracting away from any implementation, any issue of availability of a service, or even of which services are necessary to complete the request is the arrival point of SOC architectures. To achieve this, we need ways to automatically compose services, ways to choose and compare automatically services based on their capabilities and equivalences, ways to handle quality of service and user preferences, and be able to create and manage processes involving different services and let them execute distributedly in a peer-to-peer fashion. Proposals to address the above issues are the object of the current issue of the *International Journal of Cooperative Information Systems*.

In December 2003, the first International Conference on Service-Oriented Computing (ICSOC 2003)[1] was held in Trento, Italy, attracting more than 200 researchers from diverse fields such as distributed systems, data bases, software engineering and artificial intelligence. The conference had a relevant scientific impact and was followed by a second edition in New York (November 2004) and Amsterdam (December 2005), establishing itself as a reference event in the area of service-oriented computing. This special issue collects the extended and improved versions of the best five papers of the first ICSOC conference, which went through an additional peer-review process. Each one of the five papers addresses five hot issues in the field of Service-Oriented Computing: automatic service composition, distributed peer-to-peer process execution, matching of process interfaces, quality of service and service customization.

Berardi *et al.* tackle the problem of automatic service composition. They provide a technique based on finite state transition systems to check whether a given service composition exists and, in the positive case, a composition is provided. The technique is proved to be correct and implementable. In fact, it is a sound, complete and terminating procedure to automatically build service compositions.

Schuler *et al.* consider the issue of decentralized process execution. Running a process is delegated to a number of hosts which are organized in a peer-to-peer fashion. This scalable approach, which is the most reasonable one in the context of autonomous independent services, proves also to be more efficient than a centralized approach. This is shown by the experimentations ran with the prototype Osiris described in the article.

Stroulia and Wang consider the problem of matching different service descriptions. The work is important when considering issues of service substitutability or service subsumption when interacting, for instance, with service registries. The authors provide a set of techniques based on information retrieval and related fields to compare structurally and semantically services. The proposed techniques are evaluated experimentally to assess their performances.

Constraint satisfaction techniques are used by Ruiz-Cortéz *et al.* to automate the procurement of services. In this way, users may select services based on

non-functional requirements and select an optimal service. Furthermore, the consistency and conformance of procured services can be guaranteed. The authors also provide a proof of concept implementation and some experimental results.

Finally, Hull *et al.* transform the vision of a personalized web into a concrete framework based on rule-based customization. The strength of the approach lies in having rules, rather than simply values, to drive the customization of service provisioning. In this manner, the core logic of a service can be customized to the user needs, rather than simply some values of the service. The framework is implemented in the Houdini prototype. The implementation shows the feasibility of the approach and treats with great care the end-user interaction by allowing the expression of preferences via form-based interfaces.

We hope you will enjoy this special issue and will share with us the enthusiasm for the emerging field of Service-Oriented Computing whose success is bound to the solution of a number of open challenges such as those addressed in the current issue of the *International Journal of Cooperative Information Systems*. We also take this opportunity to thank all the contributors and reviewers whose work has been essential for the existence and quality of this present volume.

Reference

1. M. Orlowska, S. Weerawarana, M. Papazoglou and J. Yang (eds.), in *Proc. First Int. Conf. Service-Oriented Computing — ICSOC 2003*, Lecture Notes in Computer Science, Vol. 2910 (Springer, 2003).