Tutorial – Towards Dynamic Web Services

Luciano Baresi and Sam Guinea
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza L. da Vinci, 32 - 20133 Milano (Italy)
baresilguinea@elet.polimi.it

ABSTRACT
This tutorial introduces dynamic web services as a solution to cope with the dynamism and flexibility required by many modern software systems. Current technologies (WSDL, WS-BPEL, etc.) have proven insufficient in addressing these issues; however, they remain a good starting point for the analysis of the current situation and for building for the future.

The core part of the tutorial analyzes —by looking at available technologies and prominent research proposals— the deployment and execution of these applications within three separate phases: a composition phase, to discover available services and implement the desired behavior, a monitoring phase, to understand if a given service is behaving correctly (with respect to both functional and non-functional requirements), and a recovery phase, to react to anomalies by means of suitable replanning or recovery strategies.

In conclusion, the tutorial summarizes the main topics, presents a list of still-to-be-solved problems, and highlights possible directions for future research.

Categories and Subject Descriptors
H.1.0 [Models and Principles]: General; D.2.11 [Software Architectures]: Languages; D.2.12 [Interoperability]: Interface definition languages

General Terms
Design, Reliability, and Languages

1. INTRODUCTION
The service-oriented approach is becoming a popular paradigm for the integration of highly heterogeneous systems. The most visible advantage provided by service-oriented architectures (SoAs) is that they allow for the flexible coordination between business partners that wish to integrate their components (services) across business boundaries. However, the main novelty, with respect to component-based systems, is the new and unprecedented level of dynamism these systems can tackle. The contexts in which they can operate can be extremely unstable and evolving. Therefore, the number and quality of the services available for integration can be extremely unpredictable, and preclude —at times—the a-priori identification of all the components in a system. Moreover, the dynamic nature of SoAs requires that available functionality be sometimes discovered at run-time, and that their quality parameters be negotiated dynamically.

Such degrees of dynamism preclude us from analyzing if a system can be considered trustworthy before its deployment, and require that we consider methods and tools that can help enforce trust during run-time.

To this end, the deployment and execution of service-centric systems can be characterized by three separate phases: a composition phase, a monitoring phase, and a recovery phase. The first addresses the fact that available services must be composed to implement the desired process or meet the predefined goal. Non-functional requirements play an important role during this phase and impose that the execution environment be capable of dealing with them. The monitoring phase adds special-purpose probes to detect anomalous conditions. For example, the system must be able to understand if a given service does not answer (within a given time-frame), does not match a contract set in the process definition (be it functional or non-functional), or if it responds with an error message. In many cases, these systems cannot be simply shut down, since trust in the system would suffer as a consequence. In contrast, the systems must be able to react to anomalies by means of suitable replanning or recovery strategies that handle the deviations from the original plan, compensate anomalous situations, and continue the execution to meet the original goal.

To this end, the tutorial presents available technologies, and introduces the different solutions that can be used to build self-healing systems by analyzing (1) the dynamic composition of services, (2) the monitoring of their execution, and (3) the compensation of anomalous situations. Finally, the tutorial identifies future directions for applied research in the area.

2. GOALS AND OBJECTIVES
Besides a general introduction to the problem, the tutorial will provide participants with:

- A detailed definition of the domain, along with example applications and challenging problems;
- An organized and hierarchical presentation of the many initiatives and standards in the WS-* jungle;
- A survey of existing methodologies for the dynamic composition of Web services, their monitoring, and the resolution of anomalous situations;

Copyright is held by the author/owner.
ICSE ’06, May 20–28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.
• A proposal for a layered solution towards self-healing services;
• A thorough discussion of future research directions.

3. OUTLINE
Part I: Introduction, overview, and motivations. This part introduces the tutorial, presents the main faults that can arise in SoAs, and highlights the driving elements behind the proposed solutions.

Part II: Existing technologies. This part classifies the too many WS-* technologies and concentrates on the main ones (UDDI, WS-BPEL, WS-CDL, etc.). Its goal is to help the audience understand what is already available, what is missing, and how current technologies attempt to solve the problems previously identified.

Part III: Advanced research proposals. This part builds on the previous section and explains how the presented research proposals, methods, and tools provide solutions to the problems highlighted in Part I, and that are not currently solved by existing technologies.

Part IV: A complete approach. This part presents a complete and homogenous solution to the problem of deploying self-healing Web services.

Part V: Conclusions and future work. This last part summarizes the main elements presented during the tutorial, concludes the presentation with a list of still-to-be-solved problems, and highlights possible directions for future research by proposing a possible roadmap for their accomplishment.

4. INTENDED AUDIENCE
This tutorial is open to all software practitioners, students, managers, and researchers who want to learn more about web services in highly heterogeneous and dynamic contexts. Participants should have a basic background in computer science and be aware of what the Internet and Web technologies are (e.g., XML and SOAP), but the tutorial is designed in such a way that no special previous knowledge is required.

The tutorial will be given as a class, but active participation from the audience and discussions will be strongly encouraged.

5. REFERENCES