Proceedings of the 2\textsuperscript{nd} Workshop on

\textbf{Middleware for Service Oriented Computing (MW4SOC 2007)}

held at the

ACM/IFIP/USENIX International Middleware Conference

November 26 – November 30, 2007

Newport Beach, CA, USA

\textbf{Workshop Chairs}

Karl M. Göschka (Vienna Univ. of Technology, Austria)
Schahram Dustdar (Vienna Univ. of Technology, Austria)
Frank Leymann (Univ. of Stuttgart, Germany)
Vladimir Tosic (NICTA, Australia)

\textbf{Sponsors:}

ACM International Conference Proceedings Series
ACM Press
# Table of Contents

**2nd Workshop on Middleware for Service Oriented Computing**

Mobile Web Services Mediation Framework  ………………………………………………………p.6  
_Satish Narayana Srirama, Matthias Jarke, and Wolfgang Prinz_

CAESAR: Middleware for Complex Service-Oriented Peer-to-Peer Applications………p.12  
_Lipo Chan, Shanika Karunasekera, Aaron Harwood, and Egemen Tanin_

Ontology based Algorithm Modeling: obtaining adaptation for SOA environment……… p.18  
_Simone Grassi, Stephen Barrett, and Francesco Sordillo_

A Platform to Support Decentralized and Dynamically Distributed P2P Composite OWL-S Service Execution………………………………………………………………………………………………p.24  
_Thorsten Möller and Heiko Schuldt_

Flexible Matching and Ranking of Web Service Advertisements………………………p.30  
_Navid Ahmadi and Walter Binder_
1. WORKSHOP OVERVIEW

Service Oriented Computing (SOC) is a computing paradigm broadly pushed by vendors, utilizing services to support the rapid development of distributed applications in heterogeneous environments. The visionary promise of SOC is a world of cooperating services being loosely coupled to flexibly create dynamic business processes and agile applications that may span organizations and computing platforms and can nevertheless adapt quickly and autonomously to changes of requirements or context. Consequently, the subject of Service Oriented Computing is vast and enormously complex, spanning many concepts and technologies that find their origins in diverse disciplines like Workflow Management Systems, Component Based Computing, "classical" Web applications, and Enterprise Application Integration (EAI) including Message Oriented Middleware. In addition, there is a strong need to merge technology with an understanding of business processes and organizational structures, a combination of recognizing an enterprise's pain points and the potential solutions that can be applied to correct them.

Middleware, on the other hand, is defined by the ObjectWeb consortium as the software layer in a distributed computing system that lies between the operating system and the applications on each site of the system. Middleware is the enabling technology of system and enterprise application integration (EAI) and therefore it clearly and evidently plays a key role for SOC.

While the immediate need of middleware support for Service Oriented Architectures (SOA) is evident, current approaches and solutions mostly fall short by primarily providing support for the EAI aspect of SOC only and do not sufficiently address composition support, service management and monitoring. Moreover, quality properties (in particular dependability and security) need to be addressed not only by interfacing and communication standards, but also in terms of integrated middleware support. But what makes these issues so different in a SOA setting? Why - for instance - is traditional middleware support for transaction processing different to transaction processing in SOA, reflecting different types of atomicity needs? One answer lies in the administrative heterogeneity, the loose coupling between coarse-grained operations and long-running interactions, high dynamicity, and the required flexibility during run-time. Recently, massive-scale and mobility were added to the challenges for Middleware for SOC.
However, loose coupling is not always the best approach to solve a particular problem. In order to temporarily allow for a stronger (traditional) form of coupling (like group membership agreement or atomic transaction), the middleware also has to provide explicit and configurable means to change between different strengths of coupling and various communication paradigms. This will enable service-based applications to take the best from both worlds by dynamically adjusting the interactions between the composed services.

The highly dynamic modularity and need for flexible integration of services (e.g. Web service implementations) may require new middleware architectures, protocols, and services. These considerations also lead to the question to what extent service-orientation at the middleware layer itself is beneficial (or not). Recently emerging "Middleware as service" offerings, from providers like Amazon or from the open source community, support this trend towards "infrastructure services" that can be purchased and consumed over the Internet. However, this model may not be suitable for all kinds of middleware functions, including those addressing dependability (availability, reliability, integrity, safety, maintainability). Providing end-to-end properties and addressing cross-cutting concerns in a cross-organizational SOA is a particular challenge and the limits and benefits thereof have still to be investigated.

2. WORKSHOP CONTRIBUTION

First, in an invited keynote, P. M. Melliar-Smith reminds us to keep Service Oriented Computing simple: Service Oriented Computing allows programs to interact with each other on demand without being tightly coupled together. Unfortunately, the Web Services standard is moving rapidly towards the slippery slope of increasing complexity, elaborate standards and difficult programming, the slippery slope that engulfed CORBA and EJB. We need to find a new paradigm that can achieve the promise of Service Oriented Computing without the complexity of Web Services.

Second, the challenges of mobility and scale in service oriented systems are addressed at the middleware level, and solutions are presented in the form of mediation frameworks and middleware architectures.

Third, it is investigated how the use of semantic annotations and ontologies can improve the otherwise generic middleware solutions and consequently achieve better results with respect to adaptivity, scalability, and discovery.

To summarize: If service-oriented computing shall fulfil its promise, particular attention has to be paid to heterogeneity, scale, and dynamism of service-oriented systems, in order not to fall short by providing just "yet another technology wrapper" for existing systems. Our workshop addresses precisely these challenges on the middleware level, with a particular focus on quality of service, adaptivity, scale, and dynamics. We also address the problem of increasing complexity of middleware solutions for service-oriented systems and investigate alternate approaches for future systems.

3. STATISTICAL DATA

This is the second year for the MW4SOC workshop and 15 submitted papers show the importance of the topic. 33 reviewers performed our double-blind review process. Finally, 5 papers have been accepted resulting in an acceptance rate of 33%.

4. ACKNOWLEDGEMENTS

We would like to thank our reviewers as well as our authors for their work and support for making this an interesting workshop for research and industry. Special thanks go to our organizational chair, Johannes Osrael at the Vienna University of Technology, who did a great job during the whole paper submission and review process.