Object Orientation and Web Services⋆

Christian Zirpins[1], Giacomo Piccinelli[2], Winfried Lamersdorf[1],
and Anthony Finkelstein[2]

[1] Department of Computer Science, University of Hamburg, Germany
{Zirpins, Lamersdorf}@informatik.uni-hamburg.de
[2] Department of Computer Science, University College London, United Kingdom
{A.Finkelstein, G.Piccinelli}@cs.ucl.ac.uk

Abstract. The annual European workshop on Object Orientation and
Web Services focusses challenges and potentials of service-oriented com-
puting in relation to object-oriented technologies and methodologies. In
particular it brings together the academic and the industrial perspective
on Web Services. This year's issue was characterised by the competency
and motivation of the participants, both workshop activists who pre-
sented their specific results as well as organisers and invited speaker,
who contributed their broad experience. This report outlines the contri-
butions and discussions of the event, as well as the conclusions reached
by the participants.

1   Structure of the Workshop

The second annual European workshop on Object Orientation and Web Services
(EOOWS'04) brought together researchers and practitioners interested in the
relation of service-oriented- and object-oriented computing paradigms (section
2). With this year's event, EOOWS entered its second round since initiation at
ECOOP 2003 in Darmstadt and proved its role as a constant workshop series.
The workshop was lead by Giacomo Piccinelli and Winfried Lamersdorf, who
compiled an attractive mix that included general perspectives of an invited talk,
a brought variety of specific contributions and lively discussions spanning the
whole event. Around fifteen people where participating, including presenting and
non-presenting attendees.

Initially, the invited talk added an interdisciplinary perspective to the work-
shop and extended the discussions beyond usual Web Service scenarios. The
talk was given by *Daragh Byrne* from the Edinburgh Parallel Computing Cen-
ter (EPCC) [1]. Daragh Byrne works at EPCC as an Applications Consultant.
At the time of this writing, he is engaged in the OGSA-DAI project, which in-
volves generalizing data access for the Grid. The talk provided an architectural
blueprint of Grid systems with special focus on the Open Grid Service Architec-

---

⋆ The title of this report should be referenced as "Report from the ECOOP 2004
Workshop on Object Orientation and Web Services".

ture [2, 3] of the Global Grid Forum [4]. In particular, it detailed the relationship and interplay of Grid- and Web Service technologies.

Presentations of individual contributions made up the main part of the workshop. From the response to the call for papers, nine papers where selected for presentation. The contributions ranged from genuine and upcoming research to industrial experience (section 3). Presentations where given in relaxed 20 minute intervals and where always followed by intense discussion that resulted in further insights for listeners as well as contributors and lead to quite some network building. The workshop was concluded by a plenary discussion (section 4).

2   Themes and Objectives

Web Services are evolving beyond their SOAP, WSDL, and UDDI roots toward being able to solve significant real-world integration problems. Developers of Web Services systems are currently working on new generations of systems that incorporate security, transactions, orchestration and choreography, grid computing capabilities, business documents and processes, and simplified integration with existing middleware systems. Current economic issues continue to force consolidation and reduction in enterprise computing resources, which is resulting in developers discovering that Web Services can provide the foundation engineering and realisation of complex computing systems.

The question of how Web Services could and should change system and solution development is very much open. Are Web Services just about standards, or do they imply a new conceptual framework for engineering and development? Similarly open is the question of how requirements coming from system and solution development could and should make Web Services evolve. In particular, methodologies as well as technologies based on the object-oriented conceptual framework are an established reality. How do Web Services and object-orientation relate? How can Web Services leverage the experience built into current object-oriented practices?

The overall theme of the workshop is the relation between Web Services and object orientation. Such relation can be explored from different perspectives, ranging from system modelling and engineering to system development, management, maintenance, and evolution. Aspects of particular interest are the modularisation of a system into components and the (possibly cross-domain) composition and orchestration of different modules. Components and composition are closely connected with the issue of reuse, and an important thread of discussion within the workshop will address the way in which Web Services impact reuse.

The objective of the workshop is twofold: assessing the current work on Web Services, and discussing lines of development and possible cooperation. Current work includes research activities as well as practical experiences. The assessment covers an analysis of driving factors and a retrospective on lessons learned. The identification and prioritisation of new lines of research and activity is a key

outcome of the workshop. In particular, the intention is to foster future cooperation among the participants.

3   Contributions

The workshop contributions can be classified into four categories. Three contributions revolve around *software development for and with Web Services*. Another two contributions where related to *interaction processes for the composition of Web Services*. The topic of *Web Service management* with emphasis on context awareness and autonomy was the main line of two contributions. Finally, two contributions where concerned about *architectural patterns and general modelling of Web Service based systems*. In the rest of this section, the categories and contributions will be outlined.

3.1   Software Development of and with Web Services

Web Services border on internal software systems at two points. First, the functionality of a Web Service has to be realised by an internal software system. Second, the functionality of a Web Service is to be used by an internal software system. The relation to object-orientation is obvious in the case that these software systems are designed and/or implemented in an object-oriented way. There where workshop contributions for each of these points.

The first contribution came from *Guadalupe Ortiz* from the Quercus Software Engineering Group, University of Extremadura, Spain. During presentation of her work on *"Decoupling Non-Functional Properties in Web Services: An Aspect-Oriented Approach"*, she stressed the point that Web Service technologies offer a new and successful way for interoperability among web applications. However, current approaches do not propose an acceptable method to decouple non-functional properties from Web Service implementations, having a large amount of code scattered and tangled all over the application, thus raising problems at design. implementation, maintenance and evolution. She argued that Aspect-oriented techniques allow these properties to be easily modularised and reused. Furthermore, she showed how information about properties can be added in the WSDL file, in order to keep clients informed about the characteristics of the service they are going to use.

A contribution concerning the second aspect came from *Takashi Koshida* from the Department of Information Engineering, Matsue National College of Technology, Japan. He presented an *"Automated Dynamic Invocation System for Web Service with a User-defined Data Type"*. In particular, he claimed that this system can automatically discover the WSDL file describing a Web Service from a UDDI registry. After analysing the WSDL file, the system extracts and automatically sets up the parameters needed for dynamic invocation. Furthermore, the system dynamically generates and compiles the Java code for a user-defined data type. Finally, he presented experimental results and demonstrated the validity of this system by invoking published Web Services as practical examples.

Finally, *Fabien Baligand* and *Valrie Monfort* from IBM , MDTVision, France contributed their work on *"A pragmatic Use of Contracts and Aspects to gain in Adaptability and Reusability"* as an industrial paper that offers a practical perspective on the integration of software application systems based on Web Services. They argue that buying and selling of company departments involves huge changes in their information systems (IS). Often, the technical solution to connect distant IS is based on Web Services. This requires to model organisational structures and business processes, which are used and shared to deploy service-oriented infrastructure and Web Services. However, practice shows that distant IS are supported by different and heterogeneous technical infrastructures. The authors claim experience with SOA and Web Services for industrial projects in heterogeneous contexts. They encountered different problems like the lack of methodology to develop Web Services w.r.t. new emerging standards. Faced with industrial realities, they defined such a methodology for Enterprise Application Integration (EAI) and SOA. Moreover, they propose solutions to lacks of new WS-I standards as Policies and Meta data. They based their technical solution on contracts and aspects to gain in flexibility and reusability.

3.2   Interaction Processes for Web Service Composition

Interactions are significant for Web Service concepts. The interaction between different roles of the Web Service Model is eminent from the basic invocation of a single operation up to the choreography of a complex composite e-Service. In the later case, interactions form the building blocks for collaboration procedures between organisations [5]. Such interaction processes that *represent* an e-Service are complemented by meta interaction processes that implement a methodology leading to the *enactment* of e-Services. These complementary perspectives where each represented by one workshop contribution.

Christian Zirpins from the Distributed Systems and Information Systems Group (VSIS) at Hamburg University, Germany based his contribution about *"Service Cooperation Patterns and their Customised Coordination"* on the first perspective. He argued that service-oriented computing is meant to support loose relationships between organisations. Such relationships constitute cooperation procedures that translates to interaction processes via Web Services. Service composition deals with the specification and automated enforcement of such interaction processes and its predominant approach is orchestration, where a workflow management system (WFMS) is pro-actively coordinating the interaction activities. In most cases, the orchestration process is regarded as an implicit result of cooperation logic (actually, they are often the same) but the reverse impact of operational coordination on cooperation logic are often neglected. Based on that, he claimed that the choice of coordination alternatives impacts the quality of service and has to be customised to actual service cases and their individual participants. He proposed a potential solution approach that revolves around *service cooperation patterns*. Therein, paradigms of patterns/idioms that are well known from object-oriented design/development are applied to cooperation procedures and orchestration processes. This approach allows studying a)

reusable cooperation patterns typical for service relation-ships and b) for each pattern a range of possible coordination idioms. Finally, he sketched a technique that is intended to refine a composition process based on an analysis of its *cooperation patterns* and the application of suitable *coordination idioms* selected by rules in terms of the service context.

The second perspective is taken by *Muhammad F. Kaleem* from the Technical University Hamburg, Germany who was not present at the workshop. In his contribution about *"Interactions for Composition as a Means for Enacting Robust Composite Services"*, he focuses on the importance of the composite service enactment process for robust composite services. He discusses why this focus is necessary, and describes a methodology currently being developed which allows autonomous service providers to interact with each other for composite service enactment.

3.3 Management of and with Web Services

One of the current trends in Web Service research and – in particularly – in practice is the internal and cross-organisational operations management of platforms for atomic and composite Web Services [6]. In particular, Web Services allow for Service Level Management (SLM) of metrics that are close to the application domain (e.g. control of order throughput). Complementary, Web Service technology can act as an instrument for the management of any heterogeneous systems and build a basis for the development of new management paradigms. Two of the workshop contributions where revolving around this topics.

Concerning SLM, *Clemens Kerer* from the Distributed Systems Group, Information Systems Institute, Vienna University of Technology, Austria presented his work on *"Presence-Aware Infrastructure using Web services and RFID technologies"*. He explained that Web services have the potential to serve as a key enabling technology for environments facilitating collaborative scenarios. His group analysed a same room/same time collaborative scenario, a program committee (PC) meeting. They focus on design and implementation of a service-oriented approach for PC meetings. Their proposed solution includes various Web services for the actual support of the work, Web applications combining those services, and presence-aware technology, in this case RFID (Radio Frequency Identification) tags. The essence of this contribution was to show how a presence-aware infrastructure comprising Web services and RFID technology can be built and how they provide support for the envisaged scenario.

In contrast, *Amir Zeid* from the Department of Computer Science at the American University in Cairo, Egypt presented his efforts *"Towards Autonomic Web Services"*, that presents management *with* Web Services. He explained autonomic computing that was introduced with the promise of achieving self management; however, most of the current work does not address the problem of designing autonomic computing systems so that computing systems can be evolved towards self-management paradigm. Regarding this problem, his research ties autonomic computing with Web services to achieve a new approach, which is autonomic Web services, where each entity is a Web service behav-

ing autonomically. Adopting Web services supports the dynamic and seamless integration among computational units. He claimed that this research aims at proving that: with autonomic Web services, computing systems will be able to manage themselves as well as their relationships with each other. To achieve this objective, the research proposes a system that implements the concept of autonomic Web services; a proof-of-concept prototype of this system is currently under development and testing.

3.4   Modelling and Patterns of Web Service-Based Systems

The benefit of modelling in the development cycle of software systems is commonly agreed and Web Service-based systems are no exception to this rule. Many powerful modelling techniques emerged from the field of object-oriented development. Their application to Web Services is promising but has to be verified and customised in order to meet disparities of the paradigms. One of the most prominent approaches is the Unified Modelling Language (UML) [7] that provides fundamental modelling capabilities. A family of more specific approach revolves around capturing mature knowledge in common design- or implementation *patterns* [8, 9] that can be reused by system engineers as proven solutions for building new systems. The mapping of these approaches to Web Services was the main line of the two concluding workshop contributions.

Concerning the utilisation of UML for Web Service modelling, *Rafik Amir* from the Department of Computer Science at the American University in Cairo, Egypt presented his contribution about a *"UML Profile for Service Oriented Architectures based on W3C Web Service Architectures"*. He pointed out again that Service Oriented Computing (SOC) is the new emerging paradigm for Distributed computing and e-business processing that is changing the way software applications are designed, architected, delivered and consumed. Services are autonomous platform-independent computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols for the purpose of building agile networks of collaborating business applications distributed within and across organizational boundaries. He came to the conclusion that engineering and modelling service-oriented architectures need extensions to existing modelling techniques and methodologies. Subsequently, he propose a UML profile for service-oriented architectures.

As an example for service-oriented design patterns, *Simon Martin* from the Space Science and Technology Department, Rutherford Appleton Laboratory, UK presented *"The Service Cache Pattern and Grid Services in the European Grid of Solar Observations (EGSO)"*. He introduced the EGSO project [10] that has developed a reusable messaging infrastructure for communication between Grid applications, whereby message-relay services transparently leverage directory, storage and logging services in order to provide enhanced functionality to consumer applications. He continued to describe the proposed Service Cache pattern, which has been applied to the engineering of these services. Finally he claimed that the main potential benefits derived from the application of this

pattern include improved manageability and predictability of service delivery, and faster response times for consumers.

4   Main Lines of Discussion

The workshop closed with a plenary discussion. It mainly revolved around three topics. First, the invited talk gave reason to question the relationship between Web Services and Grids. Second, the number of contributions that combined aspect oriented techniques with Web Services promoted to fathom the potential of this constellation. Then, after addressing just a few facets of all new trends and standards, the discussion turned to the vast proliferation in the Web Service arena and raised general questions of rationale and expedience of this evolution. Finally, the thread returned to the foundation of it all and evaluated the basic Web Service standards after nearly half a decade since the emergence of WSDL and SOAP.

Web Services and Grids. Motivated by the talk of Daragh Byrne and his brought experience, an initial thread of discussion was about the relation of Web Services and the Grid. Especially the Open Grid Service Architecture (OGSA) [2] and Infrastructure (OGSI) where regarded. In fact, the situation is manifold: On the one hand, it is Web Service technology that facilitates many of the OGSA capabilities. Actually, OGSA builds on an extension of WSDL (GWSDL) that enables inheritance of port types and defines additional such port types for vital features like stable references, state maintenance and lifecycle management. That is, Grid Services are by far more sophisticated then traditional Web Services. On the other hand, Grid Services are intended for much finer grained entities then Web Services. That is, Grid Services are tailored to represent low-level resources while Web Service are normally positioned very close to application-level (business) objects. As a result, the technical relationship must not be confused with the usage of these entities (e.g. in a business-to-business interaction scenario): Web Services (if anything) are the choice for representing organizational endpoints for the cross-organizational interactions while Grid Services offer means to implement them beneath. This answers one of the main questions that emerged: could and should Web Services and Grid Services be integrated? The answer is no, because they reside on a different level of abstraction.

Web Services and AOP. A lively discussion revolved around the relation of Web Services and separation of concerns with aspect oriented programming [11]. The later is around in object-oriented development for quite a while and its application to Web Services should not come as a surprise. However, there are several alternatives for this combination. A first one is to utilize an aspect oriented language to implement Web Services and factor out service properties as aspects. This was the approach presented by Guadalupe Ortiz (see section 3.1) who could report on positive results with using AspectJ [12] for this purpose. More positive experiences in this direction where told by co-attendees from the

AOP workshop. One question here was about the dynamics of such approaches. Certainly, the possibility to use aspects for the static development of Web Services is already a benefit but it would be even better to allow the adaptation of service properties in running systems (e.g. to be used by service management systems). A different idea that was discussed revolved around separation of concerns in service composition. One possibility to do this is to code service composition logic using aspect oriented languages and prevent "hard" coding by factoring out composition aspects. Also aspect oriented concepts have been applied to workflow evolution [13]. As workflow is the underlying principle of many service composition approaches, it might be worthwhile to investigate in the separation of concerns within collaborative service processes.

Evolution of Web Service Standards. In the end of the session, the discussion shifted towards the evolution and future development of Web Services themselves. The ongoing process of organisations including diverse standard bodies (e.g. OASIS, W3C, WSI...) and companies (IBM, Microsoft...) flooding the Web Service scene with an ever-growing number of standard-like specifications was discussed very critically. This is by no means a coordinated process and the fact that numerous of such approaches overlap or contradict each other (sometimes completely) is a clear indicator for that. Not to mention that some of them are obviously not consistent or even complete, the vast majority of new Web Service specifications keeps quite aloof from implementation issues and just *state* properties of Web Services.

As regards the foundational and by now established core Web Service standards (i.e. WSDL, SOAP), there where little worries and subsequently no pressing need to evolve them was felt by the participants. On the contrary the mind within the plenum was more in the direction of applying and better understanding a stable core. We will return to this point when eventually more complex application scenarios and applications are reached and the Web Service core comes across its conceptual limits dating back to its middleware roots.

5   Conclusion

In the end, the workshop was perceived as a very positive experience by everybody. Thanks to competency and motivation of the participants, there was a concentrated and productive atmosphere in terms of both, interesting presentations of single contributions and fruitful discussions.

As a result, it can be concluded that Web Service research enters a more advanced but still very early stage that is yet far from the goals of business-to-business interaction aimed by the general service-oriented computing paradigm. In fact, the centre of gravity is still close to middleware and system integration. Meanwhile, the basic principles of this setting seem to be well understood and start to become settled knowledge that is represented, for example, as design patterns. On this basis, attention shifts to more sophisticated issues. Concepts that

only very recently entered the object-oriented arena are already being aligned with and transferred to Web Services.

An example is the obvious trend towards separation of concerns in Web Services that can be witnessed for service implementation (utilisation of aspect oriented technology) and service models (e.g. aspects of service composition logic). While in the former case Web Services benefit from the experience of the object-oriented community, the later case shows that the relationship is not one-sided: the innovative power of the (still) young Web Service discipline pays back in the currency of new concepts for issues that are out of scope for others. This is one step further towards the final goal: dynamic and flexible mega-programming for open application-level interactions.

References

1. EPCC: Computing for business and science. http://www.epcc.ed.ac.uk/ accessed at:1.8.2004 (2004)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: An open grid services architecture for distributed systems integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum (2002)
3. The Globus Project: Towards open grid services architecture. Published on the Globus website: http://www.globus.org/ogsa/ accessed at:1.7.2004 (2004)
4. GGF: Global grid forum. http://www.ggf.org/ accessed at:15.8.2004 (2004)
5. Zirpins, C., Piccinelli, G.: Interaction-driven definition of e-business processes. In: Proc. 26th International Computer Software and Applications Conference (COMPSAC 2002), Prolonging Software Life: Development and Redevelopment, 26-29 August 2002, Oxford, England. IEEE Computer Society (2002) 738–740
6. Casati, F., Shan, E., Dayal, U., Shan, M.C.: Business-oriented management of web services. Commun. ACM **46** (2003) 55–60
7. OMG: Unified modeling language, v1.5. Technical Report formal/03-03-01, Object Management Group (Mar 2003)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Professional Computing Series. Addison-Wesley, USA (1994)
9. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P.: Pattern-Oriented Software Architecture - A System of Patterns. Wiley and Sons Ltd. (1996)
10. EGSO: European grid of solar observations. http://www.mssl.ucl.ac.uk/grid/egso/ accessed at:15.8.2004 (2004)
11. Czarnecki, K., Eisenecker, U.W.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, Boston (2000)
12. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.G.: An overview of aspectj. In Knudsen, J.L., ed.: Proc. ECOOP 2001 - Object-Oriented Programming, 15th European Conference, Budapest, Hungary. LNCS 2072. Springer (2001)
13. Bachmendo, B., Unland, R.: Aspect-based workflow evolution. In Rashid, A., ed.: Workshop on Aspect-Oriented Programming and Separation of Concerns (Lancaster). (2001)