

# ASMs in Service Oriented Architectures

Michael Altenhofen\*

SAP Research, CEC Karlsruhe, 76131 Karlsruhe, Germany,  
michael.altenhofen@sap.com

**Abstract.** In this presentation we will report on a number of projects where we have successfully applied the ASM methodology to provide abstract models for a number of problems that are commonly found in Service Oriented Architectures (SOAs). In particular, we will provide solutions for the following topics: (1) service behaviour mediation, (2) service discovery, and (3) service composition.

*Introduction.* Service Oriented Architecture (SOA) serve as an architectural blueprint for designing flexible information systems. SOAs are built from autonomous functional entities, *services*, which inter-operate through well-defined, message-based interfaces that abstract from both programming languages and implementation platforms.

The most prominent incarnation of this architectural blueprint is based on the technical specification set defined by the World Wide Web Consortium (W3C) where services expose their functionality as so-called *Web<sup>1</sup> services*. The corresponding W3C technical report on the Web Services Architecture [7] describes a basic engagement model that identifies, among others, the following three fundamental entities: the service *requester* invoking functionality from a service *provider* through message exchanges. The definition of these message exchanges has been published and thus can be retrieved through a public service *discovery service*.

Although the underlying principles are not radically new, the SOA style has given raise to a lot of research in the recent years which tries to address the following problems:

– *Service Mediation.*

What if requester and provider do not perfectly agree on the interaction protocol? Is there a way to *mediate* [6] in heterogeneous scenarios where mismatches may occur both on the message content and the message flow level?

---

\* Thanks to my colleagues A. Friesen and J. Lemcke and to Prof. E. Börger who significantly contributed to the work presented here.

<sup>1</sup> The technical specifications in this area make heavily use of other specifications issued by the W3C that are related to the World Wide Web, e.g. using HTTP as one possible transport protocol for Web service messages, hence the name *Web services*.

- *Service Discovery.*  
Given a number of advertized service descriptions, which one fits best the requesters needs?
- *Service Composition.*  
How can new service functionality be exposed by composing it from available services?

*Service Behaviour Mediation.* The idea to use the ASM modelling framework for providing a high-level specification of service mediators has been stimulated by work carried out in the EU-funded research project DIP <sup>2</sup>. There, ASMs had been used to describe both *WSMO choreographies* and *WSMO orchestrations*, but the technical report [5] actually did not specify how such a choreography would be linked up to a corresponding orchestration other than by sharing some states. Although perfectly legal from a conceptual point of view, we felt that this coupling was actually *too loose* and that a more rigid model would be needed, especially when it comes to (formally) proving system properties. Further investigation of the problem revealed that we were actually looking for a model that could be used to describe *service behaviour mediators*, i.e. mediation on the message flow level. We eventually managed to come up with a simple basic model, which we called *Virtual Provider (VP)*, that allows us to construct arbitrary complex service mediator scenarios starting from simple hierarchical processing structures in one VP (so-called *seq/par-trees*) that can be extended via VP composition [2].

*Service Discovery.* It actually turned out that the underlying abstract processing model of the VP could be easily extended to serve different application scenarios. It was again in the context of the DIP project, where we had the task to provide a specification for a semantic web service discovery component. The term *semantic* is used here to indicate that the service discovery should not be based on syntactical, e.g., keyword based search, but rather on matching semantically described *goals* (goal queries) with semantic annotations of web services (capability descriptions). The overall solution is specified as a distributed architecture, where a *service discovery provider* will either distribute goal queries to peer entities or will forward a query to a local *discovery engine*. For both entities in this architecture, the specifications could be directly derived from the VP model by fairly simple refinements [1].

*Service Composition.* Our work on protocol mediation with the VP has eventually led to a new approach for *service composition*. We interpret multi-party communications as an *orchestration problem*, i.e. finding a mediator that could steer the interactions among those parties, where one party is considered the initiator of the conversation defining the *required interface*. For this, we have developed a mathematical model of (Web) services based on ASMs and a composition algorithm that is able to construct an *transactional* execution plan that

---

<sup>2</sup> See <http://dip.semanticweb.org>

guarantees that any possible execution trace in that plan will reach an expected end state for all parties, i.e. either a success or recovery state [4]. Since all artefacts in that model are specified as ASMs, we can actually construct and simulate the execution plan in the CoreASM execution engine [3].

## References

1. Altenhofen, M., Börger, E., Friesen, A., Lemcke, J.: A high-level specification for virtual providers. *International Journal of Business Process Integration and Management*. Special Issue 2006. Inderscience Publishers.
2. Altenhofen, M., Börger, E., Lemcke, J.: An Abstract Model for Process Mediation. In: Kung-Kiu Lau, Richard Banach (Eds.): *Formal Methods and Software Engineering*, 7th International Conference on Formal Engineering Methods, ICFEM 2005
3. Farahbod, R., Gervasi, V., Glässer, U.: CoreASM: An extensible ASM execution engine. In *Proc. of the 12th Intl Workshop on Abstract State Machines*. Paris, France. March 2005.
4. Friesen, A., Lemcke, J.: Composing Web-service-like abstract state machines (ASMs) Workshop on Web Service Composition and Adaptation (WSCA07). 2007 IEEE International Conference on Web Services, ICWS 2007.
5. Scicluna, J., Polleres, A., Roman, D: Ontology-based Choreography and Orchestration of WSMO Services. WSMO Technical Report 2006. <http://www.wsmo.org/TR/d14/v0.2/>
6. Wiederhold, G.: Mediators in the future architecture of information systems. In: M.N. Huhns and M.P. Singh (Eds). *Readings in Agents*, San Francisco, CA, 1997.
7. W3C, Web Services Architecture. W3C Working Group Note 11 February 2004. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, 2004.