# Comprehensive tool support for iterative SOA evolution

Thomas Schuster, Christoph Rathfelder*, Nelly Schuster, and Jens Nimis

FZI Forschungszentrum Informatik an der Universität Karlsruhe,
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe
{schuster|rathfelder|nschust|nimis}@fzi.de,
http://www.fzi.de/

**Abstract.** In recent years continuously changing market situations required IT systems that are flexible and highly responsive to changes of the underlying business processes. The transformation to service-oriented architecture (SOA) concepts, mainly services and loose coupling, promises to meet these demands. However, elevated complexity in management and evolution processes is required for the migration of existing systems towards SOA. Studies in this area of research have revealed a gap between in continuous and actual tool support of development teams throughout the process phases of evolution processes. Thus, in this article we introduce a method that fosters evolution by an iterative approach and illustrate how each phase of this method can be tool-supported.

## 1 Introduction

The use of a SOA promises organizations to adapt their software more rapidly to changing business needs. A successful implementation of a SOA is not limited to IT systems and requires changes throughout the whole enterprise [1, 2]. In order to handle this complexity, it is appropriate to implement an enterprise-wide SOA step by step using evolutional approaches [3]. These approaches can also aid enterprises upgrading an already implemented SOA.

As SOA introduction and evolution influences the whole enterprise, complex knowledge from experts of different fields is required in order to make the right decisions or changes. To support the decision making a SOA evolution process needs to be assisted by an end-to-end tool chain. The tools should be specialized to certain activities but should also be able to use information resulting from the previous activity.

In this paper we i) present an iterative method, that supports SOA introduction as well as SOA evolution and ii) give an overview of our vision of a comprehensive tool chain supporting this method. During the last years, several tools focused on certain aspects within the domain of SOA analysis, implementation and maintenance, have been developed at our research institute. We present these tools and how they support the phases in our method. The development

* Work was partially supported by the European Commission (grant No. FP7-216556)

of the method is, in addition to our own experiences, based on several published SOA case studies (e.g., Deutsche Post [4, 5], Credit Suisse [5, 6], ABB [7], and Sparkassen Informatik [8]). Furthermore, we evaluated a variety of articles documenting best practices, success factors, and experiences related to SOA implementations, as well as publications which present new research results and open research questions (e.g., [5, 9–11]). Several methods and approaches already exist to support SOA development and evolution [12, 13]. Partially these are supported by a tool chain of a certain vendor. Our method is vendor-independent.

The remainder of this article is organized as follows: Section 2 presents our SOA introduction and evolution method. Section 3 describes the tool chain that supports the different activities within the method. Section 4 concludes the paper and provides an outlook to future work.

## 2 SOA Evolution Method

Since SOA implementations, as well as legacy systems, are based on a multitude of technologies (e.g. CORBA, .Net, J2EE) a framework that supports evolution of their architecture has to be independent of the underlying technologies and products as far as possible. Thus, a generic, vendor-independent method that enables identification of assets and determination of an adequate target architecture is required. This method has to be driven by analysis of the business processes and the required IT-alignment.

Figure 1 shows the outline of such a method.


**Fig. 1.** SOA Evolution Method

The depicted method comprises five phases, from target planning, process modeling and analyis, service identification, quality prediction and assurance to service enactment. As systems are required to adapt to changing business needs, the method can and should be executed multiple times, i.e., the target planning phase can be started with the running system from the service enactment phase as input. In the following sections details of each phase will be unveiled.

### 2.1 Target Planning

The target planning phase provides an overview of the current system and gives a first insight into a possible target architecture. General, technical and business conditions need to be clarified before the actual evolution of the system can be conducted. In order to support this planning and controlling, maturity models can be adopted. Such models can be used to assess the maturity level of an existing system and assist in improving the maturity as they provide the possibility to deduce a roadmap to a successful SOA implementation. The selection of the most adequate maturity level is an important part of developing a roadmap. The highest maturity level is not always the most suitable one for each enterprise. The benefits promised by a level have to be weighed against the costs to reach and maintain that level. In order to aid this examination we suggest accomplishing the maturity level identification and setting up project conditions in this phase to allow for decisions about a necessary evolution in general.

### 2.2 Process Modeling and Analysis

In the process modeling and analysis phase existing as well as future business processes that need to be supported by the system are gathered and subject to analysis. We suggest a top down modeling approach that includes several levels of abstraction in which activities of one process may call other activities of more detailed processes (such as sub-processes in BPMN [14]). Once hierarchical models have been created services or orchestrations need to be mapped to business process activities. The modeled activities are evaluated on their suitability as service candidates. Once candidates have been identified their interfaces can be derived from in- and output data of the according business process objects. The phase is completed by an analysis that might lead to a further refinement of the modeled business processes, if necessary.

### 2.3 Service Identification

The service identification phase comprises not only the identification of new services described by the business processes but also the identification of components of the current system which are already deployed as services or that might be reused as services. Thereby, services that need to be implemented from scratch are pointed out. The implementation of new services by replacing internal components with calls of external service interfaces, is challenging and requires access to source code and detailed knowledge about the systems. Apart from the newly demanded and actually provided functionality, depending on already known metrics and documentation of the existing systems, analysis of the code quality can be necessary to choose the right components. In [15] it is stated that the existence of certain design patterns can be an indicator to reuse components as services. On the other hand identified anti patterns [16] can also be an indication that a component should not be reused but reimplemented or at least needs to be refactored before exposing it as a service. Therefore, the analysis for service identification has to cover the identification of costs and general efforts associated with deployment or modification of an existing component as a service.

### 2.4 Quality Prediction

The quality of services becomes a business critical factor as services are either offered and sold to customers or used within the enterprise to support business critical processes like the production or sales process. In these cases, the unavailability or a reduced response time of a service might have serious impact on the business results of an enterprise. Therefore, service providers should guarantee specific quality of service properties for the services they offer. In order to specify the guaranteed quality attributes like response time or throughput of a service, often strict contracts (so called Service Level Agreements (SLAs)) between service provider and customer are used.

For these reasons, this phase focuses on the quality analysis and prediction of services and service compositions. Service providers have to analyze in advance which infrastructure resources and external services are required in order to fulfil a certain quality level. This knowledge allows them to estimate costs and to acquire necessary resources. Furthermore, service consumers need detailed information on the quality of a service, as they might again build and offer a composed service based on these services. This information helps them, for example, to identify the optimal trade-off between costs and offered quality.

### 2.5 Service Enactment

The service enactment phase combines mechanisms and actions of service development, deployment, wiring and decision of concrete platforms. If not already evaluated one of the first questions before implementation of services is to choose a target platform and an adequate infrastructure to host the services later on. In terms of cost efficiency the usage of virtualization concepts has developed interesting synergetic effects. The proper choice of the target platform is often not easy since in most areas many vendors compete with similar services. Hence a decision can be driven by cost calculation, security services and dependence on certain vendors. Often enterprises try to avoid being dependent on certain vendors or products, however good service and support offers can also be preferred over scarce documented or supported open source projects.

## 3 Tool Support for SOA Evolution

The method introduced in the previous section enables a sound evolution of IT systems. However, without tooling the involved stakeholders will not be able to cope with the overall complexity. Thus, we match adaquate tools with each method phase, shown in Figure 2. The tool-support is depicted in the following subsections.

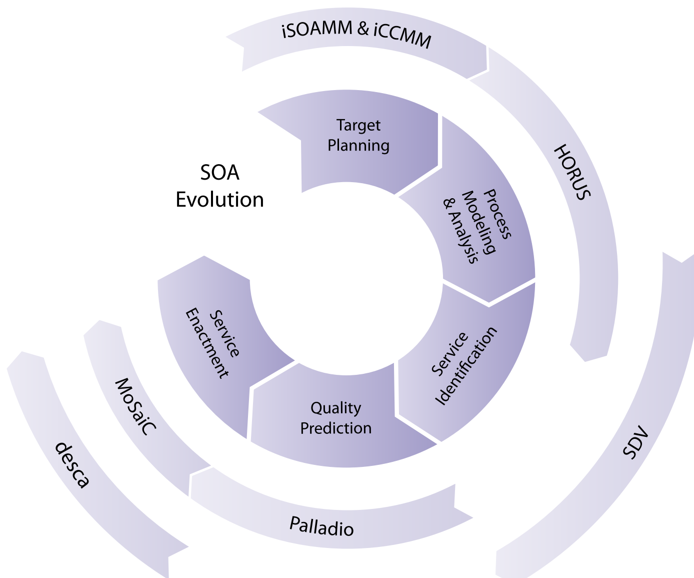


**Fig. 2.** SOA Evolution Process and Tooling

### 3.1 Target Planning with SOA and Cloud Maturity Models

As SOA adoption in an organization is a gradual process a maturity model is an adequate tool to assist this procedure. Existing SOA maturity models (e.g., [17] or [18]) were in most cases developed by SOA vendors (e.g., IBM, BEA, HP, or Oracle) and cannot deny a dependency on the respective products. Additionally, the vendors take the desire to reach the highest maturity level for granted. Therefore, they often neglect supporting an enterprise in the selection of the most *appropriate* maturity level. For this reason, we developed the independent SOA Maturity Model (iSOAMM) [19]. The iSOAMM is product- and technology-independent and considers technical as well as organizational aspects. It covers the five evaluation viewpoints *Service Architecture*, *Infrastructure*, *Enterprise Structure*, *Service Development*, and *Governance*. In addition to the evaluation of the current maturity, it eases the selection of the most adequate maturity level by pointing out the challenges, benefits, and risks associated with each level. Based on the iSOAMM, we developed a Web-based questionnaire tool[1]

[1] http://soa.fzi.de

that evaluates the current maturity level and proposes next steps to reach a higher maturity level.

Recently, more and more enterprises tend to complement their SOA evolution with the emerging technology Cloud Computing, which promises a flexible and highly efficient use of hard- and software on-demand. Like SOA evolution Cloud Computing adoption should be a gradual process. In order to support this process in the target planning phase, we extended the iSOAMM with an independent Cloud Computing Maturity Model (iCCMM) [20], which is also available as Web-based tool. The iCCMM consists of five maturity levels and six criteria groups. iSOAMM and iCCMM and their respective tools can be used independently but also in combination. Combined adoption of both models allows to show links between SOA and Cloud maturity levels and to propose a recommended Cloud maturity level for each SOA maturity level.

### 3.2 Process Modeling and Analysis with Horus Business Modeler

Once an adequate maturity level has been determined, the identification, analysis and refinement of business processes is the next step according to the SOA evolution method. Hereby, the business analyst is typically supported by a modeling platform (such as ARIS, IBM WebSphere Business Modeler, Borland Together, Signavio or YAWL). Since it offers a vendor independent and formal modeling technique based on Petri nets, mechanisms for simulation and enterprise wide collaboration we suggest the usage of Horus Business Modeler (HBM[2]) for tooling in this phase. Business processes modeled in HBM can be propagated to other users and stakeholders, thereby accessing and integrating necessary information, knowledge and requirements that are related to the identified processes. In order to enable business process enactment, tasks of business processes modeled with HBM might be associated with actors, stakeholders and service descriptions (given in WSDL [21]). If the process has the structure of a so called webservice-net (ws-net [22]) the generation of a BPEL [23] orchestration is possible. However, the choice of appropriate services requires further knowledge about already offered services by the business analyst. In demand for an enhanced tool support we are investigating on the integration of query mechanisms that operate on semantically enriched service descriptions, such as the ones used by SDV.

### 3.3 Service Identification using SDV

SDV (Semantisches Diensteverzeichnis, *Ger. for Semantic Service Registry*) [24] is a tool that allows to capture and process the semantics of services' business aspects in order to make services more accessible to business analysts than common technically oriented service registries. To keep the technical and business aspects of the service descriptions separate, UDDI [25] is used as the basis for the technical parts of the descriptions while Semantic MediaWiki (SMW) [26] provides a collaborative front end to the registry's business oriented model which

[2] http://horus.biz

is based on OWL-Lite [27]. This combination of technologies and tools allows manual interaction with the registry for human users as well as automated processing of the descriptions and integration into an existing or newly established SOA.

### 3.4 Quality Prediction with Palladio

After the identification of the required services, it is necessary to evaluate their influences on the overall quality of service compositions. With PCM-Bench, we provide an Eclipse-based workbench[3] to evaluate the performance and resource usage of services and service compositions based on the specification of functional and extra functional properties of the basic services. PCM-Bench is based on the Palladio Component Model (PCM) [28] which is a domain-specific modeling language for modeling component-based and service-oriented software systems. The tool includes automatic transformation of design-oriented architectural models to analysis-oriented performance models including layered queueing networks [29], stochastic process algebras and simulation models [28]. In PCM, architectural models are parameterized over the system usage profile, the execution environment and external services. This allows to reuse models in different contexts for different usage scenarios and execution environments. As Palladio works on the model level, services do not need to be implemented for the performance analysis. This eases to evaluate the impact of different external and integrated services as well as the used hardware or cloud resources on the performance of the composed service. The applicability and usability of Palladio was investigated in theoretical and industrial context (e.g. [30, 31], which were mainly located in the domain of business information systems.

Automated SLA negotiation requires an automated evaluation of different alternatives regarding the used hardware resources offered by a service-oriented infrastructure and the guaranteed quality of integrated services. Palladio allows the integration into SLA management frameworks as sketched in [32]. Thus, Palladio can, in addition to the analysis of services and service compositions performed during design time, also be used in fully automated SLA negotiation frameworks.

### 3.5 Using desca to assist Service Enactment

desca – the demonstrator for elastically scalable Cloud applications – provides insights to the changing landscape of enterprise architectures and thus helps to support architecture decisions in the service enactment phase. It is motivated by the observation that current Cloud offerings in the area of IaaS (Infrastructure as a Service)[33] are not 1:1 replacements for traditional IT infrastructure components, but lead to completely new application architectures for enterprise systems. The demonstration is based on a well-known Cloud Computing sucess story [34]: in 2007, the New York Times faced a challenge, when it wanted to

[3] http://www.palladio-approach.net

make available its archive consisting of 11 million articles, dating back to 1851 over the web. NYT had already scanned all the articles in TIFF format and now needed to translate a four TerraByte picture pile into PDF files. Our simple implementation of this scenario shows an end-user oriented version of this scenario. It depicts the use of Amazon's Cloud offerings (i.e. AWS, the Amazon Web Services EC2, S3, SQS and SimpleDB) [35] to implement a simple Web-Based graphics to PDF file converter, which allows for the users to upload plain picture files via a Web interface and receive PDF-files which have been generated in the Cloud. Thus, it can easily be shown how standard AWS features like CloudWatch, AutoScaling and LoadBalancing can be used to implement entreprise-grade SOAs running in the Cloud.

### 3.6 MoSaiC – Mashups for Situational Collaboration

The models, business processes, rules, SLAs, and other data required in the SOA evolution phases are often represented in enterprise documents. These enterprise documents facilitate collaboration and communication between the stakeholders in a visual format. While IT-supported document collaboration is well established for structured recurring business processes, creative processes that emerge and evolve instantaneously lack an explicit document abstraction at the architectural level and thus cannot be easily integrated with business logic or instrumented for organizational purposes. As to this, we investigate a novel approach to represent documents as mashups of services and put these service-oriented documents in the center of an open collaboration process [36]. Document mashups offer an interactive, intuitive and dynamic way to indicate the structure and behavior of document fragments provided by human collaborators or IT systems as services.

In our approach, a RESTful Web architecture and prototype infrastructure of a lightweight document service bus represent document fragments as stateless software services. A simple composition model built on events and rules coordinates document content, editing and publishing services. The infrastructure enforces these regulations in a non-invasive way by mapping and controlling interaction between the participants. Thus, MoSaiC presents a complementary approach in order to support creative, situational and collaborative business processes based on documents.

## 4 Conclusion and Outlook

Evolution of SOA systems can be a challenging task. In order to tackle the complexity of this task we introduced an iterative evolution method comprising five phases. A sound evolution is based on the identification of an adequate target architecture, while the details are captured throughout phases of business process modeling, analysis and identification of according services. Thereafter, the method envisions the usage of mechanisms for quality prediction of the designed solution, this might also include changes to the modeled architecture. As a last step service enactment puts the upgraded solution in an operation mode. In order to support this method at the best we furthermore suggested the usage of tools that cover all phases.

The suggested tools that support the phases of the evolution method are currently developed independently. In our future work, we plan the investigation of further integration and the definition of interfaces that synergize the results.

## References


1. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Systems Journal **44**(4) (2005) 691–708
2. Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G.: Impact of service orientation at the business level. IBM Systems Journal **44**(4) (2005) 653–668
3. Heffner, R., Fulton, L., Stone, J.: Key SOA Success Factors: A Starter Kit For SOA. Forrester Research (2006)
4. Helbig, J.: SOA Serie: Teil 1-5. CIO (2007) http://www.cio.de/schwerpunkt/d/Deutsche-Post-Brief.html.
5. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA. reprint. edn. Prentice Hall PTR (2006)
6. Legner, C., Heutschi, R.: SOA Adoption in Practice - Findings from Early SOA Implementations. In: Proc. of European Conference on Information Systems (ECIS 2007). (2007)
7. Gizanis, D., Heutschi, R., Solberg, T.: Global Order Management Services Support Businesses at ABB (2005) ttp://www.alexandria.unisg.ch/Publikationen/23667.
8. Zimmermann, O., Milinski, S., Craes, M., Oellermann, F.: Second generation web services-oriented architecture in production in the finance industry. In: OOPSLA '04. (2004)
9. Baer, T.: SOA: BUILDING THE ROADMAP. zapthink white paper (June 2007)
10. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal **16**(3) (2007) 389–415
11. Crawford, C.H., Bate, G.P., Cherbakov, L., Holley, K., Tsocanos, C.: Toward an on demand service-oriented architecture. IBM Systems Journal **44**(1) (2005) 81–107
12. Schepers, T.G.J., Iacob, M.E., Van Eck, P.A.T.: A lifecycle approach to soa governance. In: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, New York, NY, USA, ACM (2008) 1055–1061
13. Ramollari, E., Dranidis, D., Simons, A.J.H.: A survey of service oriented development methodologies. In: 2nd Young Researchers' Workshop on Service Oriented Computing, Leicester, U.K. (June 2006) pp. 1–6
14. OMG: Business Process Modeling Notation, Version 1.2 (2009) http://www.omg.org/spec/BPMN/1.2/PDF/.
15. Happel, H., Schuster, T., Szulman, P.: Leveraging source code search for reuse. In: High Confidence Software Reuse in Large Systems. (2008) 360–371
16. Simon, F., Seng, O., Mohaupt, T.: Code Quality Management: Technische Qualitaet industrieller Softwaresysteme transparent und vergleichbar gemacht. 1 edn. Dpunkt Verlag (May 2006)
17. Trops, B.: SOA Maturity Modell, oder der Weg zu einer Service Orientierten Architektur. Java Forum Stuttgart (2006) http://www.jfs2006.de/jfs/2006/folien/B7_Trops_ORACLE.pdf.

18. Meier, F.: Service Oriented Architecture Maturity Models - A guide to SOA adoption. Master's thesis, University of Skövde, School of Humanities and Informatics (2006)
19. Rathfelder, C., Groenda, H.: iSOAMM: An independent SOA Maturity Model. In: Proc. of 8th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'08). Volume 5053/2008. (2008) 1–15
20. Kolovou, V.: Junction of the isoamm for soa with the evaluation framework cloudtco for cloud computing offerings. Master's thesis, Universit"at Karlsruhe (2009)
21. W3C: Web Services Description Language (WSDL) Version 2.0 (2007) http://www.w3.org/TR/2007/REC-wsdl20-20070626.
22. Koschmider, A., Mevius, M.: A petri net based approach for process model driven deduction of BPEL code. In: On the Move to Meaningful Internet Systems 2005: OTM Workshops. (2005) 495–505
23. OASIS: Web Services Business Process Execution Language Version 2.0 (2007) http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
24. Paoli, H., Schmidt, A., Lockemann, P.C.: User-Driven Semantic Wiki-based Business Service Description. In: 3rd International Conference on Semantic Technologies (I-Semantics 07). (2007)
25. Bellwood, T.: UDDI Version 2.04 API Specification (2002) http://uddi.org/pubs/ProgrammersAPI_v2.htm.
26. Kroetzsch, M., Vrandecic, D., Voelkel, M.: Semantic MediaWiki. In: ISWC 2006. Volume 4273., Springer-Verlag New York, Inc. (2006) 935–942
27. W3C OWL Working Group: OWL 2 Web Ontology Language – Document Overview (2009) http://www.w3.org/TR/owl-overview.
28. Becker, S., Koziolek, H., Reussner, R.: The Palladio component model for model-driven performance prediction. Journal of Systems and Software **82** (2009) 3–22
29. Koziolek, H., Reussner, R.: A Model Transformation from the Palladio Component Model to Layered Queueing Networks. In: Performance Evaluation: Metrics, Models and Benchmarks, SIPEW 2008. Volume 5119. (2008) 58–78
30. Andrej, R.: Evaluation des Vorhersageverfahrens "Palladio" im industriellen Kontext der CAS Software AG. Master's thesis, Univ. of Karlsruhe (TH) (2008)
31. Huber, N., Becker, S., Rathfelder, C., Schweflinghaus, J., Reussner, R.: Performance Modeling in Industry: A Case Study on Storage Virtualization. In: ACM/IEEE 32nd International Conference on Software Engineering, Software Engineering in Practice Track, Capetown, South Africa. (2010) Acceptance Rate: 23
32. Comuzzi, M., Kotsokalis, C., Rathfelder, C., Theilmann, W., Winkler, U., Zacco, G.: A framework for multi-level sla management. In: Proceedings of the 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC), Stockholm, Sweden (2009)
33. Lenk, A., Sandholm, T., Klems, M., Nimis, J., Tai, S.: What's Inside the Cloud? An Architectural Map of the Cloud Landscape. In: Cloud09 workshop at the Int. Conference on Software Engineering. (2009)
34. Gottfrid, D.: Self-service, Prorated Super Computing Fun! (2007) http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/.
35. Amazon Web Services LLC: Amazon Web Services (2010) http://aws.amazon.com/.
36. Schuster, N., Zirpins, C., Tai, S., Battle, S., Heuer, N.: A service-oriented approach to document-centric situational collaboration processes. In: WETICE. (2009) 221–226