

# Analyzing the SOA ability of Standard Software Packages with a dedicated Architecture Maturity Framework

Helge Buckow, Hans-Jürgen Groß, Gunther Piller,  
Karl Prott, Johannes Willkomm, Alfred Zimmermann

SOA Innovation Lab e.V.  
Workstream SOA and Standard Platforms  
c/o Deutsche Post AG  
Charles-de-Gaulle-Straße 20  
53113 Bonn

info@soa-lab.de, helge\_buckow@mckinsey.com, hans-juergen.gross@daimler.com,  
gunther.piller@fh-mainz.de, karl.prott@capgeminisdm.com,  
johannes.willkomm@iteratec.de,  
alfred.zimmermann@reutlingen-university.de

**Abstract:** The SOA Innovation Lab investigates the use of standard software packages in a service-oriented context. As a part of this investigation the SOA capabilities of products from different vendors need to be evaluated. For this purpose a SOA architecture maturity framework has been developed, leveraging and extending CMMI and TOGAF, as well as other state-of-the-art frameworks and methods. Besides details about our framework, we present and analyze first results from the assessment of various vendor platforms.

## 1 Introduction

The growing complexity of IT landscapes is a challenge for many companies. A large number of standard software packages - mostly extended and modified - individual software solutions, legacy applications, and different infrastructure components lead to high cost and limited ability to respond quickly to new business requirements. Many companies start enterprise architecture management (EAM) initiatives to address this problem. In areas where flexibility or agility is important, SOA is the approach of choice to organize and utilize distributed capabilities. Here, the use of standard software is often a challenge, in particular when dealing with services on a fine granular level.

Initially SOA was burdened with hype and inflated expectations. Now it is part of an ongoing discussion about software architecture. The benefits of SOA are recognized. They comprise flexibility, process orientation, time-to-market, and innovation. The adoption of tools and methods for SOA is growing. An overview about the current status of adoption can be found in the SOA Check 2009 [ME09]. Reports on the maturity of SOA technology from vendors are provided by various analysts (see e.g. [Na+08, Pe+08, KP09]).

To analyze the SOA ability of standard platforms in a systematic way, the SOA Innovation Lab has developed a questionnaire based on a SOA architecture maturity framework. The latter was constructed by integrating different analysis approaches for architecture dimensions, using a consistent meta-model based on correlation analysis of intrinsic model elements. Details about this framework, the corresponding questionnaire and general findings from first vendor assessments are focus of this paper.

Our SOA architecture maturity framework is part of an approach for the design of a service-oriented enterprise architecture with custom and standard software packages, which we briefly sketch in the following (for details see [Bu+10]): The method starts with the definition of company domain maps, identifying in particular areas where SOA benefits - like agility, flexibility, and reduction of redundancies - are a priority. As a next step one needs to define and decompose the services for these domains, to identify standardisable services. On this basis one is able to decide whether standard platforms should be used within a SOA architecture.

In addition to the overall decision, whether a standard platform should be used within a certain domain, it is necessary to map a vendor solution to a company's domain map and its services, to evaluate the overall functional fit. For this, the SOA ability of the identified package needs to be evaluated against specific SOA use cases. An important part of this analysis is besides the functional fit the identification of dependencies of the standard software package regarding processes, functions and data. The reason for this is that it often leads to problems when functionality of standard software packages cannot be used outside its usual context.

For standard software packages that fulfill the functional and non-functional requirements, a target-architecture needs to be developed. It includes the high-level system architecture, as well as integration patterns for the physical integration of systems (see also [FP03, HW04]). The SOA Innovation Lab has developed a capability map for integration that allows, to structure corresponding requirements. In addition, a taxonomy for integration patterns and corresponding best practices have been compiled (see .e.g. [PW10]). Finally a business-case-oriented implementation plan has to be defined and executed.

In this paper we provide details about our SOA architecture maturity framework in Section 2. Section 3 summarizes the derivation of an assessment questionnaire for vendor workshops. Initial findings from the evaluation of standard platforms are presented in Section 4. In Section 5 we sketch future developments.

## 2 SOA Architecture Maturity Framework

Enterprises need to systematically evaluate opportunities from a potential investment in SOA with standard platforms in heterogeneous IT-environments. To enable corresponding assessments of the SOA ability of standard software, we have developed a SOA architecture maturity framework - *SOA Maturity Model Integration* (SOAMMI) [Zi09]. For this purpose we have combined a consistent metamodel for assessing transcend disciplines, with content elements from holistic enterprise architecture frameworks which comprises all important architecture dimensions.

Regarding the metamodel we have built upon CMMI [Cm09] - which is originally an assessment framework for processes. To transform CMMI into a specific framework for the assessment of the maturity of enterprise and software architectures, we have combined CMMI with current architecture framework and maturity models. In particular we use TOGAF [To09] as a basic structure for enterprise architecture, spanning all relevant levels. In addition, we have cross checked and – if appropriate - extended our model with supporting elements of the following maturity models:

- The **Architecture Capability Maturity Model (ACMM)** [Ma09] framework, which is included in TOGAF, was originally developed by the US Department of Commerce. The main scope of ACMM is the evaluation of enterprise architectures in internal enterprise architecture assessments. The goal of ACMM assessments is to enhance enterprise architectures by identifying quantitatively weak areas and to follow an improvement path for the identified gaps of the assessed architecture. The ACMM framework consists of six maturity levels and nine specific architecture elements ranked for each maturity level - deviant from CMMI. SOAMMI was influenced by some definitions of ACMM for basic maturity levels of enterprise architecture, especially for the Maturity Level 0 (None) of ACMM - defining no existing enterprise architecture program to the Maturity Level 1 (Initial) with poor managed and ad hoc architecture processes, having limited operating unit acceptance of the enterprise architecture process. The definitions for Maturity Level 2 (Under Development) and 3 (Defined) of ACMM provided elements for a basic documented and managed enterprise architecture process covered by SOAMMI.
- The **SOA Maturity Model of Inaganti/Aravamudan** [Ma09] considers the following multidimensional aspects of a SOA: scope of SOA adoption, SOA maturity level to express architecture capabilities, SOA expansion stages, SOA return on investment, and SOA cost effectiveness and feasibility. The scope of SOA adoption in an enterprise is differentiated by following levels: intra department or ad hoc adoption, interdepartmental adoption on business unit level, cross business unit adoption, and the enterprise level, including the SOA adoption within the entire supply chain. The SOA maturity levels are defined related but different to CMMI using five ascending levels to add enhanced architectural capabilities: level 1 for initial services, level 2 for architected services, level 3 for business services, level 4 for measured business services,

and level 5 for optimized business services. In a two-dimensional view - SOA scope and SOA maturity level - proper expansion stages for the systematic introduction of SOA in an enterprise are differentiated: fundamental SOA in a local department view, networked SOA with architected services on business unit level, and process enabled SOA on the enterprise level or in conjunction with suppliers. The SOA return on investment (ROI) increases gradually with increased maturity levels and matured SOA adoption. Shaded areas in the maturity model represent additionally no go areas specifically the non-cost-effective and the infeasible areas of SOA adoption. We have considered the following more technical focused critical success factors from the definition of Maturity Level 2 (Architected Services) for specific Architecture Areas like TSC Technology Services & Capabilities of the Technology Architecture Domain of SOAMMI: heterogeneity and distributed systems, reliable messaging, mediation services (ESB), database integration, application integration, versioning, central security management, routing and transformation, minimal service reuse, ease of deployment, IT performance management.

- The **SOA Maturity Model from Sonic** [Ma09] distinguishes five maturity levels of a SOA, and associates them in analogy to a simplified metamodel of CMMI with key goals and key practices. Key goals and key practices are the reference points in the SOA maturity assessment. We mention the following Key Goals: institutionalize use of SOA, put in place architecture leadership for SOA, prove returns from use of standards technology, which have influenced the definition of the Maturity Level 2 (Managed) of SOAMMI.
- The **SOA Maturity Model of ORACLE** [Ma09] characterizes in a loose correlation with CMMI five different maturity levels - opportunistic, systematic, enterprise, measured, industrialized and associates them with strategic goals and tactical plans for implementing SOA. Additionally following capabilities of a SOA are referenced with each maturity level: Infrastructure, Architecture, Information & Analytics, Operations, Project Execution, Finance & Portfolios, People & Organization, and Governance. The Maturity Level 2 (Systematic) of the SOA Maturity Model from ORACLE has influenced technical views on Architecture Areas within the Application Architecture and the Technology Architecture Domain of SOAMMI specifying important SOA infrastructures like initial project level use of ESB and BPEL for service integration and orchestration, service-level access to information sources, enterprise applications through standards for Web Services: WSIF, JCA, JMS, initial use of service registry, basic service management infrastructure for monitoring and declarative application of runtime policies, e.g. message level security.

The resulting metamodel of SOAMMI is shown in Fig. 1.

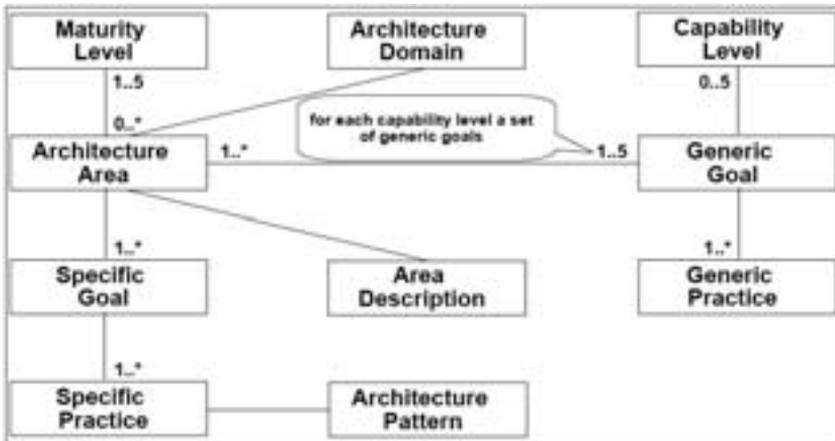


Figure 1: Metamodel of the SOA architecture maturity model SOAMMI [Zi09]

The model contains five Maturity Levels [Zi09]. They characterize the maturity of an organization as a whole, including all areas of enterprise architecture management. Currently most platforms are expected to be placed within level one and three. The main characteristics of those are:

#### **Maturity Level 1: Initial**

- Vendor service architecture is not performed or contains initial coverage only
- Architecture is unpredictable and poorly controlled
- Initial service architecture methods and corresponding knowledge transfer only

#### **Maturity Level 2: Managed**

- Vendor service architecture is managed, with medium completeness and coverage
- Vendor supports architecture enablement, corrective actions when necessary
- Vendor service architecture is institutionalized within own products

#### **Maturity Level 3: Defined**

- Vendor service architecture is defined with large completeness and coverage
- Customer service architecture is agile, derived from standard vendor architecture
- Vendor supports service strategy and architecture governance with methods, tools

The top level structure of SOAMMI is organized in five Architecture Domains adapted from TOGAF [To09]: *Architecture Strategy and Management, Business Architecture, Information Architecture, Application Architecture, Technology Architecture, Service & Operation Architecture, Architecture Realization.*

Architecture Areas were taken primarily from TOGAF [To09], Quasar Enterprise [En+08] and essential [Es09], as well as from business requirements and pilot use cases defined by members of the SOA Innovation Lab. Finally we end up with 22 areas [Zi09], including e.g.

- Architecture Domain: *Architecture Strategy and Management*; Architecture Area: *Enterprise Architecture Management*; Purpose: *Identify effectiveness of SOA strategy for ERP*
- Architecture Domain: *Business Architecture*; Architecture Area: *Business Capabilities and Services*; Purpose: *Identify and maintain supported business capabilities and business services from the business architecture of customers*
- Architecture Domain: *Information Architecture*; Architecture Area: *Data Entities & Components*; Purpose: *Identify and model main data entities and data components*
- Architecture Domain: *Application Architecture*; Architecture Area: *System Domains*; Purpose: *Define system domains from business capabilities, services and processes*
- Architecture Domain: *Technology Architecture*; Architecture Area: *Platform Services*; Purpose: *Identify and model platform services as agile building blocks from basic infrastructure*

In analogy with CMMI our metamodel contains Specific Goals, which describe the objective within a single architecture area. Necessary activities associated with a specific goal are expressed through Specific Practices. As an example, within the architecture domain *Business Architecture* and the architecture area *Business Capabilities and Services* we find specific goals (SG) and practices (SP) like:

- SG 1: Determine business services for SOA packaged software solutions and optimize business processes
  - SP 1.1: Identify and map business services to business capabilities
  - SP 1.2: Determine degree of coupling between services
- SG 2: Analyze coverage, adaptability and functional completeness of business capabilities and services
  - SP 2.1: Assess coverage of supported business services from customer perspectives
  - SP 2.2: Assess adaptability and functional completeness of business services

Following the philosophy of CMMI we also include Capability Levels. They relate to generic goals, which institutionalize the achievement of specific goals of an architecture area and are similar for all those areas. The implementation of specific goals is detailed by specific practices. As in CMMI, we use capability levels from 0 (Incomplete) to 5 (Optimizing). To achieve a specific maturity level, a certain capability level, with all corresponding generic goals, needs to be fulfilled. E.g. for Maturity Level 2 (Managed)

one need to fulfill the specific goals for all related architecture areas, as well as all generic goals for Capability Level 2 (Managed).

### **3 Assessment Questionnaire**

Vendor assessments need to address the key challenges for companies during the built-up and management of service-oriented architectures with standard software in heterogeneous IT environments. At this stage we therefore do not consider all dimensions of SOAMMI that fulfills all academic requirements, but restrict ourselves to a pragmatic approach which can be completed in a 3-4 hour workshop with vendor experts.

In the following we describe the artifacts which we developed for an effective vendor assessment. Then we sketch the procedure we followed in corresponding vendor workshops.

#### **Artifacts for an effective vendor assessment**

Assessments of the SOA ability of standard software packages can be viewed as an ideal mean to engage with vendors on all relevant challenges of SOA for standard software. Therefore, we did not design our assessment in form of a survey that could be filled out remotely, but rather focused on a discussion format where answers should include artifacts, cases, best practices, etc. As most questions have different relevance and meaning for different companies, our assessment is not intended to serve as a vendor ranking of any kind.

These goals imply that a pragmatic simplification of SOAMMI is required, that needs to be enriched with specific user requirements from companies using SOA in heterogeneous environments with standard platforms.

The complete SOAMMI model includes 22 architecture areas with 38 specific goals and over 122 specific practices. Answering all these questions would yield a complete picture, but it would lack the pragmatic use cases and would require more than 10 hours to complete. Still the structure is relevant, as it ensures the coverage of all important architecture areas and helps to stay focused.

To ensure practical relevance, members of the SOA Innovation Lab have collected their most important use cases for business contexts where they think SOA and standard platforms brings benefit, but where significant implementation challenges are expected. These use cases go down to the level of singular services, tools and technologies. This approach also helps to avoid generic responses from vendors on assessment questions.

Following these ideas, the basic structure of our questionnaire was taken from SOAMMI architecture areas with one or more question per specific goal. User requirements have been consolidated and mapped against specific goals. Wherever no user requirements could be mapped, specific practices have been used to generate questions on the level of specific goals. Through this procedure each specific goal could be related to at least one concrete question.

To avoid subjective judgment, the answer to each question was ranked, using one of three distinctive levels only:

- Not fulfilled (value zero): There is no evidence or example available
- Partially fulfilled (value one): The topic of the question is addressed, but there are still apparent gaps
- Completely fulfilled (value two): the topic of the question is fulfilled as best practice

For reasons of applicability, we simplified the SOAMMI model for the use in an assessment as shown in Figure 3.

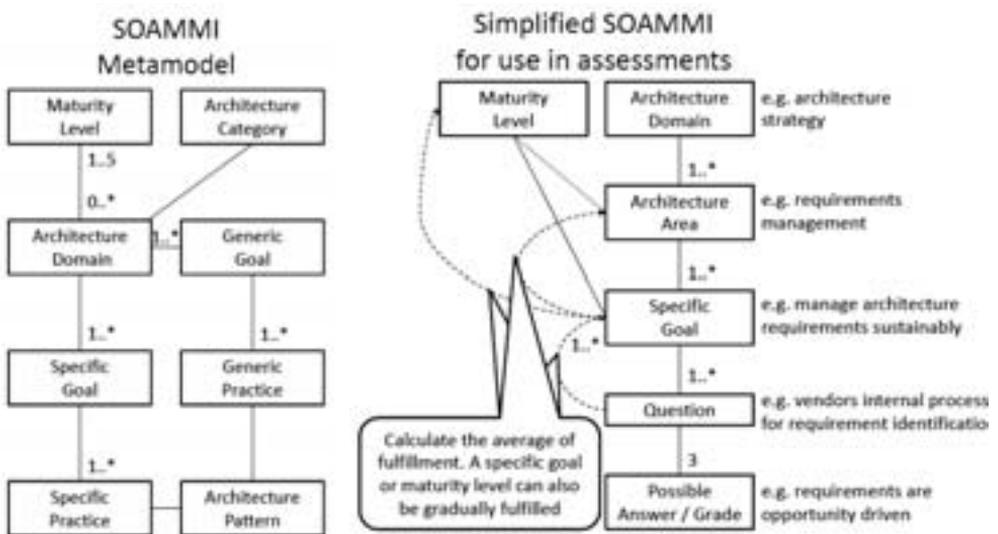


Figure 2: Simplified SOAMMI Model for use in vendor assessments

First, we did not formulate questions for generic goals but concentrated on specific goals and corresponding questions.

Second, the pure CMMI logic requires that a level can only be reached if all goals are completely achieved. If there is just one specific goal that could not be achieved, the corresponding maturity level cannot be reached at all. Given these constraints, most vendors would be at level 1. In order to highlight areas for improvement, we added a degree of fulfillment for maturity levels. For each maturity level, all assessment values from the questions of this level are added together. The fulfillment of a level is then indicated as the percentage of the maximal possible value for this level (i.e. number of questions per level multiplied by value two). As a result, each of the five maturity levels has a percentage of fulfillment.

Developing an assessment framework on this basis resulted in a questionnaire which was the foundation of the assessment process with the selected vendors. Here are examples of level 2 questions with their mapping to SOAMMI, taken from the assessment questionnaire:

- Architecture Domain: Architecture Strategy & Management  
Architecture Area: Requirements Management:
  - What is the vendor's internal process and governance to get manage SOA related requirements from customers and industry specific organizations?
  - How is the ideal business capability map created/generated?
  - How are requirements found/set/derived?
  - Which/what information is communicated back to the user and when?
  
- Architecture Domain: Business Architecture  
Architecture Area: Business Domains and Capabilities
  - Where are specific SOA capabilities in the vendor capability map?
  - What SOA capabilities are requested/planned/realized?
  - Are methods available to map a vendor specific capability map to customer specific domain/capability maps?

### **The Assessment Process**

The assessment process takes about 3 months to complete for each standard software provider overall. The first step is a Pre-Workshop (2-3 hours), in which the SOA Innovation Lab presents the background and questionnaire and the vendor has the opportunity to present his SOA strategy. This workshop is essential to make sure, that the vendor can identify the appropriate experts for the assessment workshop itself. Then the actual Assessment Workshop (4-6 hours) is held a few weeks later, so that the vendor has enough time to identify the experts that should participate and prepare answers. The SOA Innovation Lab then prepares the summary of the findings and presents these back to the vendor (1-2 hours). Finally, a series of follow up workshop for specific questions (3-4 hours each) is arranged with the vendor.

## 4 Analysis and Synthesis

Our experience with assessment workshops with vendors has been very positive. Each vendor showed strong interest and was happy to hear additional user views on the topic. Figure 3 shows a summary result.

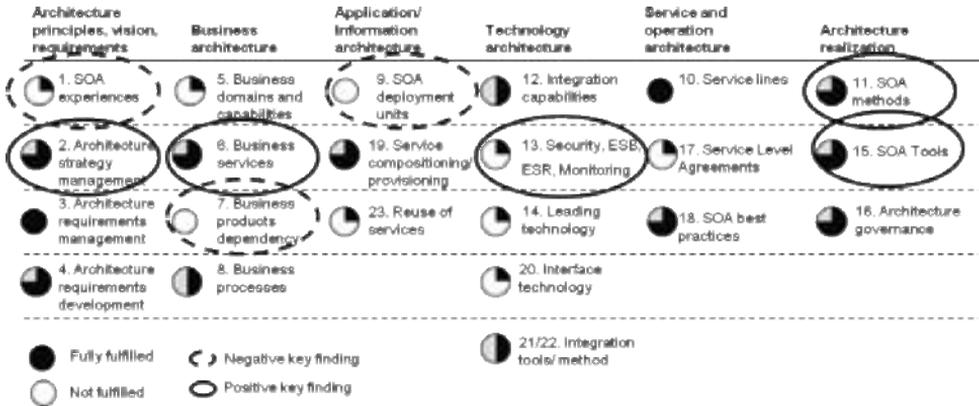


Figure 3: Overview of Vendor Workshop Results

In addition to the answers to all specific questions, we have synthesized key findings that highlight our view on the actual SOA ability of a standard platform across vendors.

01. SOA experiences: Even though SOA has been a topic for vendors for years now, there are no major SOA implementations that include standard software systems. Most cases have the quality of a *prove of concept*, often focusing on GUI integration, instead of deep functional integration. There seems to be a gap between those SOA capabilities that are offered and those which can be actually used in a SOA.

02. Architecture strategy management: SOA is seen as an important part of overall strategy with no alternative in the long term. All vendors have developed SOA strategies and have integrated it into their product roadmap. In most cases, SOA enablement is a mandatory requirement for the development of new functionality.

06. Business Services: Vendors offer solution maps that describe the functionality in terms of services and have developed methods to find existing services to a given requirement. In addition, vendors are developing solution scenarios, that offer not just the individual service but a complete set of processes that implement a business solution.

07. Business product dependencies: Vendors have invested substantially in SOA, but in many cases, SOA has been only applied as wrapping of existing systems, without changing the core of the application. This means that business services are tightly coupled and it often intransparent for the user how the dependencies between services work.

09. SOA deployment units: No vendor offers licenses that allow the usage of individual services instead of the whole system. This means that users still have to purchase the whole application, which hinders a best of breed approach for composite applications.

11. SOA methods: There is a rich offering for methods for governance, implementation guidelines, etc. for SOA available. SOA is not just seen as the technical implementation, but rather as an engineering discipline that goes beyond service interfaces.

13. Security, ESB, ESR, monitoring: Industry standards are implemented within the standard software, but standards like SAML leave room for interpretation. This makes it difficult to integrate solutions across several standard platforms, which is a requirement for most users.

15. SOA tools: All standard platform providers have added tool suites to their portfolio that support SOA development. The integration of these tools within development layers and across platforms is still not completely solved.

In summary, there are still obstacles to apply standard software in a heterogeneous SOA environment. Often, a vendor's SOA approach is specific to the vendor. E.g. each vendor has structured business functionality - a business domain map – defined and described in an individual way. However these business domain maps are vendor specific and often do not correlate with company specific domain maps. Vendors also often use specific semantics and data models and have incompatible technologies (ESB, repository,...) that do not integrate seamlessly into overall heterogeneous landscapes.

For most vendors, products are only SOA *enabled*. This means that SOA is implemented as wrapper around existing interfaces, and the internal structure is still monolithic. This typically results in a very granular and technical view (e.g. over 3.000 services) that is difficult for the user to identify and comprehend, and therefore to implement. In addition, there are many dependencies between services that often require certain modules to be implemented and populated with data, before services from other domains can be used.

Finally, most vendors have not adopted a business model that supports the usage of standard software through services. The deployment unit still is the entire software package. Individual services cannot be licensed and license models have not been adapted to service usage. Especially in a heterogeneous environment SOA SLAs will become important, but are not established yet.

Many vendors have invested early in SOA, long before users were ready to use new SOA enabled components in an appropriate way. The investment therefore was mostly to SOA enable products from a technical point of view, without considering the business scenarios that they should support. Therefore the adoption on the user side is slow, with only a few (100-300 per vendor) pilot SOA cases, mostly focusing on GUI integration. This is a tiny fraction of the overall installed base for standard software.

The most important result from the assessment workshops with vendors is that there is a strong interest from vendors to work together with the SOA Innovation Lab to further refine SOA methods and to develop solutions for the SOA use cases. We think that this is a great asset and we will continue to build on these relationships to further develop the maturity of SOA and standard software.

The experiences from the SOA Innovation Lab show that most companies see SOA as an important part of their architecture management strategy. In order to implement this strategy in an environment with standard software from different vendors, there are key requirements that should be developed together with the vendors:

- Users need services that are as independent of their context as possible and that do not require the full implementation of the standard software, this should also be reflected in the license model.
- Individual process building blocks, that can be orchestrated to an overall business solution are key, technical interfaces come second.
- Focus on service enablement should affect areas that have high requirements for process agility and are value added for the business.

## 5 Conclusion

A new method for evaluating the SOA maturity of standard software packages and its vendors has been introduced. Based on the work of CMMI - an assessment and improvement model for software processes - we have transformed and developed a suitable model for the evaluation of SOA capability and maturity. Our architecture evaluation approach was founded on the current TOGAF standard for enterprise architectures. SOAMMI – the SOA Maturity Model Integration – is the result of a metamodel-based conception and synthesis to provide a sound base for practical evaluations of service-oriented standard platforms in heterogeneous environments.

The SOAMMI framework was applied in several assessment workshops with vendors of service-oriented platforms and has provided transparent results for subsequent changes on service-oriented product architectures and related processes. It should be noted however, that the results of these assessments need to be interpreted in the context of company specific strategies and use cases. As a consequence they cannot provide vendor rankings of any kind.

Going forward, the SOA Innovation Lab plans to use SOAMMI as an ongoing framework for the cyclical evaluation of standard software packages. Based on real-world use cases and ongoing investigations in integration patterns [PW10, Zi09], the framework will be continuously optimized.

In general SOAMMI turned out to be an ideal tool to foster and guide the cooperation between companies and software vendors on current and future challenges for SOA and standard software packages in heterogeneous IT environments.

## Literature

- [Bu+10] Buckow, H., Groß, H.-J., Piller G., Prott, K., Willkomm J., Zimmermann, A.: Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010), Paderborn 2010. (to appear)
- [Cm09] CMMI for Development. Version 1.2 Carnegie Mellon University, Software Engineering Institute, 2006, <http://www.sei.cmu.edu/reports/06tr008.pdf>, 2009.
- [Cs96] CSC: Standardsoftware und geschäftliche Flexibilität. Foundation Bericht 107, Juni 1996.
- [En+08] Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.P., Voß, M., Willkomm, J.: Quasar Enterprise. dpunkt.verlag 2008.
- [Es09] The Essential Architecture Project: <http://www.enterprise-architecture.org>, 2009.
- [FP03] Fowler, M.: Patterns of Enterprise Application Architecture. Addison Wesley 2003.
- [Gr07] Grigoriu, A.: An Enterprise Architecture Development Framework. Trafford Publishing 2007.
- [HW04] Hohpe, G., Woolf, B.: Enterprise Integration Patterns. Addison Wesley 2004.
- [KP09] Kenney, F.L., Plummer, D.C.: Magic Quadrant for integrated SOA Governance Technology Sets. Gartner Research 2009.  
[http://www.gartner.com/DisplayDocument?doc\\_cd=166481](http://www.gartner.com/DisplayDocument?doc_cd=166481), 2009.
- [Ma09] ACMM Architecture Capability Maturity Model, The Open Group 2009.  
Inaganti, S., Aravamudan, S.: SOA Maturity Model. BP Trends, April 2007, <http://www.bptrends.com/publicationfiles/04-07-ART-The%20SOA%20MaturityModel-Inagantifinal.pdf>, 2009. Sonic: SOA Maturity Model.  
[http://soa.omg.org/Uploaded%20Docs/SOA/SOA\\_Maturity.pdf](http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf), 2009. Oracle: SOA Maturity Model,  
<http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet>, 2009.  
Open Group: OSIMM Maturity Model for SOA, <http://www.opengroup.org/projects/soa-book/page.tpl?CALLER=faq.tpl&ggid=1319>, 2009. IBM: SIMM Services Integration Maturity Model, <http://www.ibm.com/developerworks/webservices/library/ws-soa-simm/>, 2009.
- [ME09] Martin, W., Eckert, J.: SOA Check 2009. Ergebnisse einer empirischen Studie. [www.soa-forum.de/pdf/SOA-check2009.pdf](http://www.soa-forum.de/pdf/SOA-check2009.pdf), 2009.
- [Na+08] Natis, Y.V., Pezzini, M., Thompson, J., Iijima, K., Sholler, D.: Magic Quadrant for Application Infrastructure for New Systematic SOA Application Projects. Gartner Research 2008. [http://www.gartner.com/DisplayDocument?doc\\_cd=163409](http://www.gartner.com/DisplayDocument?doc_cd=163409), 2009.
- [Pe+08] Pezzini, M., Natis, Y.V., Iijima, K., Sholler, D., Thompson, J., Vecchio, D.: Magic Quadrant for Application Infrastructure for SOA Composite Application Projects. Gartner Research 2008. [http://www.gartner.com/DisplayDocument?doc\\_cd=163401](http://www.gartner.com/DisplayDocument?doc_cd=163401), 2009.
- [PW10] Prott, K., Wissing, M: Praxis erprobte Muster und Landkarten zur service-orientierten Integration von Standardsoftware. Submitted to INFORMATIK2010, Leipzig 2010
- [To09] TOGAF Version 9, The Open Group Architecture Framework 2009.
- [Zi09] Zimmermann, A.: SOAMMI – SOA Maturity Model Integration - Conceptual Framework. To be published