Evaluating Service-oriented Vendor Platforms with a Dedicated Architecture Maturity Framework

Helge Buckow
McKinsey & Company
SOA Innovation Lab
Berlin, Germany
helge_buckow@mckinsey.com

Hans-Jürgen Groß
Daimler AG
SOA Innovation Lab
Stuttgart, Germany
hans-juergen.gross@daimler.com

Oliver F. Nandico
Capgemini
SOA Innovation Lab
Munich, Germany
oliver.f.nandico@capgemini.com

Gunther Piller
University of Appl. Sciences Mainz
SOA Innovation Lab
Mainz, Germany
gunther.piller@fh-mainz.de

Karl Prott
Capgemini
SOA Innovation Lab
Hamburg, Germany
karl.prott@capgemini.com

Alfred Zimmermann
Reutlingen University
SOA Innovation Lab
Reutlingen, Germany
alfred.zimmermann@reutlingen-university.de

***Abstract* – The SOA Innovation Lab - an innovation network of industry leaders in Germany and Europe - investigates the practical use of vendor platforms 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 service-oriented architecture evaluation framework has been developed and currently extended, leveraging and extending CMMI and TOGAF, as well as other service-oriented state-of-the art frameworks and methods. Besides details about our evaluation framework, we present and analyze results of various service-oriented platforms from assessments with four major vendors. Our idea and contribution is to extend existing Service Oriented Architecture (SOA) maturity frameworks to accord with a sound metamodel approach. Our metamodel for architecture evaluation is based on the well understood and standardized Capability Maturity Model Integration (CMMI), which was originally used to assess software processes and not architectures. Our specific architecture capability evaluation approach is the result of a metamodel-based analysis and synthesis from state of art models. The paper presents an original approach for systematically and cyclic evaluations of heterogeneous service-oriented platforms in practical use.**

Keywords – Evaluation; SOA Vendor Platforms; SOA Maturity Model; SOAMMI; CMMI; TOGAF; Assessment Questionnaire; Framework Validation; Results; Key Findings.

I. INTRODUCTION

The growing complexity of IT landscapes is a challenge for many companies. A large number of packaged solutions platforms - 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 [1] and [2] (EAM) initiatives to address this problem. In areas where flexibility or agility in business are important, SOA is the approach of choice to organize and utilize distributed capabilities. Here, the use of standard software [3] 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 SOA adoption and reports on the maturity of SOA technology from vendors is provided by [4], [5], and [6].

To analyze the SOA ability of major vendor platforms in a systematic way, the SOA Innovation Lab has developed a questionnaire-based assessment method based on a specifically designed SOA architecture maturity framework to support the fundamental evaluation method [7]. 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 consecutive assessments with four major vendors 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 [3]): 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.

 ISBN: 978-1-61208-152-6

For this, the SOA ability of the identified package needs to be evaluated against specific SOA use cases.

For software services 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 [8] and [4]). 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 has been compiled. Finally, a business-case-oriented implementation was defined and is currently under development and evaluation.

In this paper, we provide in Section II details about the related work background models leading to our SOA architecture maturity framework in Section III. Section IV summarizes the derivation of an assessment questionnaire for vendor workshops. Initial results from the evaluation of four major SOA vendor platforms are presented in Section V. In Section VI, we draw conclusions and sketch future developments.

II. BACKGROUND AND MODEL INTEGRATION

Enterprises need to systematically evaluate opportunities from a potential investment in SOA with standard platforms in heterogeneous IT-environments. 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 [9], which is originally an assessment framework for software processes and not for enterprise software architectures. To transform CMMI into a specific framework for the assessment of the maturity of enterprise and software architectures, we have originally combined CMMI with current architecture framework and maturity models. Our approach is more generally and different to ATAM [10], which is an architecture evaluation process, based on risk-adapted definable quality goals and fine granular architecture requirements. In particular we use TOGAF [1] as a basic structure for enterprise architecture, spanning all relevant enterprise and software architecture types.

Of course, TOGAF is missing as a general standard important architecture detail structure and doesn't cover all investigated architecture domains. In addition, we have cross checked and – if appropriate - extended our model with supporting elements from the following state of art SOA maturity models, and with our original model integration extensions, which are mentioned in Section III.

The Architecture Capability Maturity Model (ACMM) [11] 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.

The SOA Maturity Model of Inaganti/Aravamudan [12] 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 Maturity Model from Sonic [13] 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, and prove returns from use of standard technologies, which have influenced the definition of the Maturity Level 2 (Managed) of SOAMMI.

The SOA Maturity Model of ORACLE [14] 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.

 ISBN: 978-1-61208-152-6

III. SOAMMI - ARCHITECTURE MATURITY FRAMEWORK

To enable corresponding assessments of the SOA ability of standard software, we have originally extended our SOA architecture maturity framework - *SOA Maturity Model Integration* (SOAMMI) from [7] and added architecture classification models and architecture evaluation and integration patterns. In respect to requirements from customer oriented domain models and reference use scenarios, our SOAMMI architecture maturity framework introduces the following originally defined maturity levels, which define important quality criteria for software and enterprise architecture excellence and help to measure the architecture maturity of vendor products:

1. *Maturity Level: Initial*

The Initial Level is the entry level of architecture maturity. Here the vendor service architecture is incomplete or with no or initial coverage related to the customer demand. The architecture is unpredictable and poorly controlled. The software architectures are ad hoc and chaotic. The assessed software organization does not provide a stable environment to support software and enterprise architectures.

2. *Maturity Level: Managed*

Projects of managed organizations have ensured that architectures are planned and executed in accordance with an architecture policy. Projects typically employ skilled architects who have adequate resources to produce controlled outputs. Software architectures are monitored, controlled, reviewed and evaluated from time to time, for adherence with architecture standards.

3. *Maturity Level: Defined*

Architectures are well characterized and understood, and are rigorously described in standards, procedures, tools, and methods. The service architecture of the software technology vendor is defined, having large, increasing completeness and coverage. An organization's set of architecture standards is established and improved over time. The customer service architecture is agile tailored from standard vendor architecture.

4. *Maturity Level: Quantitatively Managed*

This high mature level software organization establishes and uses quantitative objectives and architecture specific metrics / key architecture indicators for software architecture quality and architecture management performance as criteria in managing architectures. Quantitative objectives are based on the needs of the customer, end users, organization, and architecture implementers. Architecture artifacts and benefits are measured at vendor and customer side.

5. *Maturity Level: Optimizing*

The highest level maturity organization continually improves its software and enterprise architectures based on a quantitative understanding of the common causes of variation inherent in architectures. Their organizational focus is on continually improving architecture performance through incremental architecture development, innovative architecture management and technological improvements.

The top level structure of SOAMMI is organized considering five Architecture Domains adapted from TOGAF [1]: Architecture Strategy and Management, Business Architecture, Information Architecture, Application Architecture, Technology Architecture, Service & Operation Architecture, Architecture Realization.

Architecture Areas where originally derived primarily from TOGAF [1], Quasar Enterprise [5] and Essential [2], as well as from business requirements and pilot use cases defined by members of the SOA Innovation Lab. Architecture areas are the correspondent architecture structures for process areas from CMMI. We have defined 22 genuine architecture areas of SOAMMI fitting our architecture evaluation scope, but different from CMMI (see [9]) - and structured them according to standard architecture maturity levels.

SOAMMI supports both the staged representation and the continuous representations (Figure 1). The same staging rules as in CMMI apply to SOAMMI and should therefore enable the flexible adoption of both model representations: continuous - for assessing single architecture areas and staged - for assessing the whole enterprise architecture.

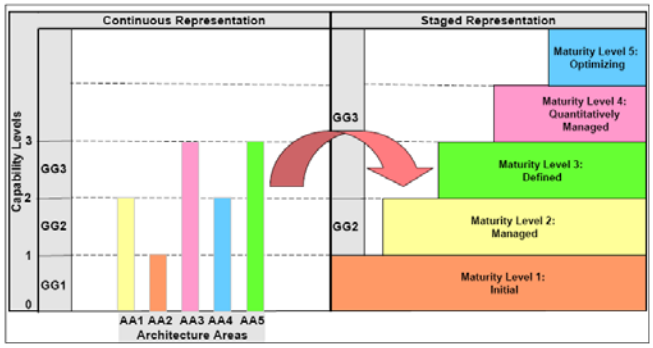

Figure 1. Architecture Capability and Maturity Levels

The continuous representation of SOAMMI is similar to CMMI, which uses levels to denote the capability and the incremental improvement path for specific architecture areas. The assessment of capability levels could be applied to iterate specific architecture areas or to assess or improve a focused innovation aspect, involving one ore more architecture areas. To verify and support the persistent institutionalization of architecture areas we have introduced in the SOAMMI framework generic goals and practices.

Specific Goals describe the objectives 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 Specific 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

 ISBN: 978-1-61208-152-6

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.

IV. ASSESSMENT QUESTIONNAIRE MODEL

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.

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 [7] with one or more question per specific goal. Additionally we have considered and adapted from [6] SOA design questions that affect quality attributes of vendor platforms. 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. 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 fulfillments.

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:

1. *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?
2. *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 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

 ISBN: 978-1-61208-152-6

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.

V. ANALYSIS AND SYNTHESIS RESULTS

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 2 shows a summary result.

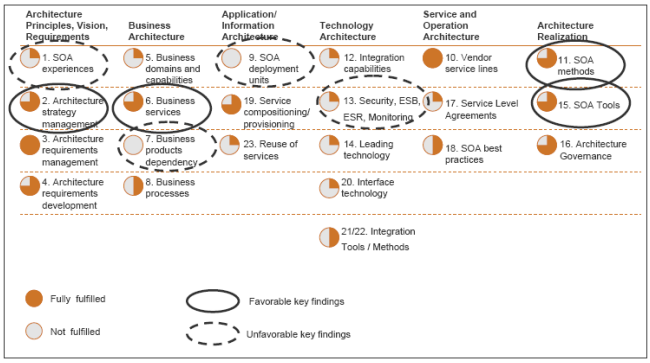

Figure 2: Overview of Vendor Workshop Results

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:

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 proof 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.

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.

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, which offer not just the individual service but a complete set of processes that implement a business solution.

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 therefore inflexible. Often dependencies between services were complex and could be ambiguous for the service composition.

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.

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.

Security, ESB, ESR, service 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.

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 service level agreements will be 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

 ISBN: 978-1-61208-152-6

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.

VI. 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 architecture evaluation patterns the framework will be continuously optimized.

ACKNOWLEDGMENT

The authors thanks the SOA Innovation Lab and his member companies for the joint research and practical validation of the architecture evaluation framework SOAMMI with major service technology vendors IBM, SAP, ORACLE, and Microsoft, supplemented by associated software and consulting firms and the network of scientific consultants.

REFERENCES

[1] TOGAF "*The Open Group Architecture Framework*", Version 9, The Open Group, 2009.

[2] *Essential Architecture Project*, http://www.enterprise-architecture.org, last access: June, 19th, 2011.

[3] H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, and A. Zimmermann, "*Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes*", Proc. 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010), Paderborn, 2010, GI-Edition - Lecture Notes in Informatics (LNI), P-160, 2010, pp. 219-230.

[4] T. Erl, "*SOA Design Patterns*", Prentice Hall. 2009.

[5] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.P. Richter, M. Voß, and J. Willkomm, „*Quasar Enterprise*" dpunkt.verlag, 2008.

[6] P. Bianco, R. Kotermanski, and O. Merson, "*Evaluating a Service-Oriented Architecture*", CMU/SEI-2007-TR-015, Carnegie Mellon University, Software Engineering Institute, 2007.

[7] A. Zimmermann, "*Method for Maturity Diagnostics of Enterprise and Software Architectures*", A. Erkollar (Ed.) ENTERPRISE & BUSINESS MANAGEMENT, A Handbook for Educators, Consulters and Practitioners, Volume 2, Tectum 2010, ISBN 978-3-8288-2306-8, 2010, pp. 129-172.

[8] G. Hohpe and B. Woolf, "*Enterprise Integration Patterns*", Addison Wesley, 2004.

[9] CMMI-DEV-1.3 2010 "*CMMI for Development, Version 1.3*" Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2010-TR-033, 2010.

[10] R Kazman, M. Klein, and P. Clements "*ATAM: Method for Architecture Evaluation*", Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2000-TR-004, 2000.

[11] ACMM, "*Architecture Capability Maturity Model*", in TOGAF Version 9, The Open Group Architecture Framework, The Open Group, 2009, pp. 685-688.

[12] S. Inaganti and S. Aravamudan, "*SOA Maturity Model*", BP Trends, April 2007, 2007, pp. 1-23.

[13] Sonic: "*A new Service-oriented Architecture (SOA) Maturity Model*", http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf, last access: June, 19th, 2011.

[14] Oracle: "*SOA Maturity Model*", http://www.scribd.com/doc/2890015/oraclesoamaturitymodel cheatsheet, last access: June, 19th, 2011.

 ISBN: 978-1-61208-152-6