ESARC - Enterprise Services Architecture Reference Cube for Capability Assessments of Service-oriented Systems

Alfred Zimmermann
Reutlingen University
Software Architecture Research Group
Reutlingen, Germany
alfred.zimmermann@reutlingen-university.de

Gertrud Zimmermann
ZIMMERMANN UND PARTNER
Enterprise Software Architecture Research
Pfullingen, Germany
gertrud.zimmermann@online.de

***Abstract* – An original ESARC-Enterprise Services Architecture Reference Cube for supporting evaluation and optimization of service-oriented architectures is introduced. Current approaches for assessing architecture quality and maturity of service-oriented enterprise software architectures are rarely validated and were intuitively developed, having sparse reference model, metamodel or pattern foundation. Cyclic assessments of complex service-oriented systems and architectures should produce convergent and comparable evaluation results. Today architecture evaluation findings are hardly comparable. This is a real problem in cyclic evaluations of advanced architecture quality concepts to get a stable foundation for introducing service-oriented enterprise architectures for adaptive systems. Our idea and contribution is to extend existing enterprise and software architecture reference models and maturity frameworks to accord with an integral enterprise architecture reference model approach. We have applied our service-oriented ESARC in several assessment workshops with global vendors of service-oriented platforms. This experience provides the base for further investigations and improvements of our approach. ESARC provides for both cyclic architecture quality assessments and for the architecture construction and optimization a standardized and normative classification scheme of important architecture artifacts for service-oriented enterprise systems.**

Keywords – Service-oriented Architecture; Enterprise Architecture; ESARC; Reference Architecture; Architecture Patterns; Architecture Capability and Maturity Assessments.

I. INTRODUCTION

Since recent years innovation oriented companies have introduced service-oriented computing paradigms and combine this with traditional information systems. Typical service-oriented technologies include systems following a service-oriented architecture (SOA). Service-oriented systems close the business - IT gap by delivering efficiently appropriate business functionality and integrating legacy systems with standard application platforms. Our approach is to investigate the practical use of the SOA ability of standard platforms in commercial use [1] for members of the SOA Innovation Lab, which is a major innovation and research network in Germany and Europe.

In assessing the quality of implemented SOA vendor platforms and the integral architecture of service-oriented enterprise systems, we were faced with the problem of not real comparable evaluation findings from consecutive (cyclic) assessments of heterogeneous systems. Our previous assessment findings were done without an architecture reference model. This causes that multiple evaluations of enterprise systems with service-oriented architectures were blurry and hardly comparable within a series of consecutive architectural tests and therefore produced less meaningful assessment results. The aim of our research is to enhance analytical instruments for cyclic evaluations of business and system capabilities of different service-oriented platforms and enterprise systems.

The hypothesis in our current research paper for the ESARC is:

1. ESARC - Enterprise Services Architecture Reference Cube is an effective concretization of the TOGAF [2] framework and other seminal work on enterprise architectures [3], service-oriented reference models and software architectures [4], [5], [6], and defines useful architecture artifacts with their main relationships.
2. ESARC provides a useful foundation of a reference structure for metamodel-based capability assessments of service-oriented systems and their architecture [1], [7], as well as for architecture assessment patterns [8] from previous work.

We are reporting about a novel holistic approach of the ESARC – the Enterprise Services Architecture Reference Cube, which helps enterprise and software architects to define and structure their evaluation object - the service-oriented enterprise and software architecture - in a standard way. In order to specify our innovative enterprise and software architecture assessment method, we used a metamodel-based approach for capability evaluations of architecture elements and their main relationships. For this purpose we have extended, integrated and adapted elements from convergent architecture methods, patterns, related standards and reference models from the state of art.

In the following Section II, we introduce base and seminal related work on reference models, reference architectures and architecture patterns, as well as open architecture standards, frameworks, and service-oriented architecture maturity models. In Section III, we present the main view of our original developed ESARC architecture

 ISBN: 978-1-61208-152-6

reference model. Section IV mentions results from our method validation of ESARC from assessments of four major SOA vendor platforms. Finally, Section V summarizes our conclusions and mentions some ideas from current research and for future work.

II. RELATED WORK

Our research is based on following formal architecture concepts from [9] and their relationships: software architecture, reference architecture, reference model, and architecture patterns. A reference model for SOA [4] is a generic fundamental model as in [9] that embodies the basic idea and provides a decomposition of functionality of a given problem, together with the data flow between elements. The reference model contains an abstract technology agnostic representation of the elements and their relationships, showing the interactions between basic concepts. The concept of reference architecture [9] and [5], [6] is the result of a mapping of an architecture reference model to software elements and contains the related data flow between them.

Architecture patterns are representations of a set of architectural constraints for architecture elements and their relation types. Architecture patterns [9], and [10], [11] show quality attributes and represent known solutions for a given problem. An architecture pattern records the architecture decisions taken by many architects in order to resolve a particular architecture problem. Patterns are human readable structures of text and graphics showing a standardized and repeatable way to derive a solution from a specified problem in a specific context. Our developed and practically validated pattern catalog [8] for quality patterns of enterprise software architectures relies originally on our previous developed service-oriented architecture maturity framework [7] for assessing architecture capabilities and maturity of service-oriented enterprise systems.

The Architecture Tradeoff Analysis Method (ATAM) [15] is a foundation method for our specific architecture evaluations of service-oriented enterprise systems. A seminal work used in the preparation of our service-oriented architecture assessments is [16], which provides concrete guidelines for the design of our questionnaire as in [1].

Service-oriented architecture SOA [12] is the computing paradigm that utilizes services as fundamental flexible and interoperable building blocks for both structuring the business and for developing applications. SOA promotes a business oriented architecture style, based on best of breed technology of context agnostic business services that are delivered by applications in a business focused granularity. To provide agile composition of services within a worldwide environment and to enable flexible integration of published and discovered components, SOA uses a set of XML-based standards like WSDL, SOAP, UDDI, and others. A main innovation introduced by SOA is that business processes are not only modeled. Business process models are used in a more mature way consistently within a Model Driven Architecture (MDA) approach to generate new and agile orchestrations or compositions of web services based on process diagrams. Early definitions of SOA were technology focused and the differences between SOA and web services were often blurred. SOA technologies emerged due to the expansion of the Internet technology during the last years and produced abundance specifications and standards as in [4], [5], [6], and [13], which are developed by open standard organizations like W3C, OMG, OASIS, and The Open Group. The perspective of a service development process is offered by [14] and [11].

Our architecture reference model ESARC relates closely to SOAMMI, which is our previous designed maturity framework for evaluation of enterprise and service-oriented product architectures. Unfortunately most of existing SOA and EA maturity models lack a clear metamodel base. Therefore we have extended CMMI [17] in our previous research, which is a framework for assessments of software processes, and transformed it into a specific framework for the assessment of the maturity of service-oriented enterprise and software architectures [1] and [7]. Therefore we have combined and extended CMMI with architecture quality criteria from current architecture frameworks and architecture maturity models. In particular we use TOGAF [2] as a basic structure for enterprise architecture, spanning all relevant levels. In addition, we have cross checked and – if appropriate - extended our metamodel with supporting elements from known maturity models.

The Architecture Capability Maturity Model (ACMM) [18] 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 quantitative weak areas and to show an improvement path for the identified gaps of the assessed architecture. The ACMM spans six maturity levels and defines nine specific architecture elements.

The SOA Maturity Model in [19] considers the following multidimensional aspects of a SOA: scope of SOA adoption, SOA maturity levels - to express architecture capabilities, SOA expansion stages, SOA return on investment, as well as SOA cost effectiveness and feasibility. The scope of SOA adoption in an enterprise is differentiated by following levels: intra-department or ad hoc adoption, inter-departmental 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 Model from Sonic [20] distinguishes five maturity levels of a SOA, and associates them - in analogy to a simplified metamodel of CMMI - with key goals and key practice. Key goals and key practices are reference points in SOA maturity assessments.

The SOA Maturity Model of ORACLE in [21] characterizes in a loose correlation with CMMI five different maturity levels and associates them with strategic goals and tactical plans for implementing SOA. Additional capabilities of a SOA are referenced with each maturity level: Infrastructure, Architecture, Information & Analytics, Operations, Project Execution, Finance & Portfolios, People & Organization, and Governance.

 ISBN: 978-1-61208-152-6

III. ESARC – ENTERPRISE SERVICES ARCHITECTURE REFERENCE CUBE

ESARC is an original abstract architecture reference model which defines an integral view for main interweaved architecture types. ESARC was derived primarily from state of art architecture frameworks like TOGAF [2], essential [3], the service model of ITIL, and from resources for service-oriented computing [11], [12], [5]. The aim of the ESARC architecture reference model is to be universally applicable in different cyclic repeatable architecture evaluations and structural optimizations of enterprise and software architectures. ESARC abstracts from a concrete business scenario or technologies.

The Open Group Architecture Framework (TOGAF) [2] is the current standard for enterprise architecture and provides the basic blueprint and structure for the service-oriented enterprise software architecture domains of ESARC: Architecture Governance, Architecture Management, Business & Information Architecture, Information Systems Architecture, Technology Architecture, Operation Architecture, and Service Architecture.

The formal foundation for ESARC, as detailed in [4], [5], and [6], is an abstract representation of standardized architecture building blocks in a layered acyclic relationship. The layer semantics implies that the basic layers are prerequisites for higher architecture layers. At a higher granularity, all architecture domains are parts of the holistic architecture composition framework of ESARC.

The ESARC – Enterprise Services Architecture Reference Cube unifies orthogonal architecture domains into aligned architecture views, which yield an aid for examination, comparison, classification and quality rating of different architecture aspects. ESARC is our holistic definition of a full service-oriented architecture used both for assessing and optimization of service-oriented product lines and for families of application systems. Our unifying perspective of service-oriented enterprise systems integrates and helps to align business and the technology aspects.

Figure 1. ESARC – Architecture Governance and Management.

The main types of enterprise software architectures like Business & Information Architecture, the Information Systems Architecture, and the Technology Architecture are organized by the Architecture Governance and Management framework. Architecture Governance as in Figure 1 conforms to the SOA Governance Framework in [13] and defines and maintains the Architecture Governance cycle.

The Architecture Governance cycle sets the abstract governance frame for concrete architecture activities within the enterprise software or a product line development. The Architecture Governance cycle specifies the following management activities: plan, define, enable, measure, and control. The second aim of Architecture Governance is to set rules for architecture compliance with internal and external standards. Policies for governance and decision definition are set, to allow a standardized and efficient process for architecture decisions within the enterprise architecture organization. Because enterprise and software architects are acting on a sophisticated connection path coming from business and IT strategy to the architecture landscape realization of interrelated business domains, applications and technologies, Architecture Governance has to set rules for empowerment of software people, define the structures and procedures of an Architecture Governance Board, and set rules for communication.

Benefits from well organized architecture governance (adapted from [2]) are: transparency of accountability, informed delegation of authority, controlled risk management, protection of the existing asset base through maximizing reuse of existing architectural components, proactive control, monitoring, and management mechanisms, value creation through monitoring, measuring, evaluation, and feedback, increased visibility of decision-making in supporting internal processes and external requirements, and greater shareholder value. The enterprise architecture increasingly represents the core intellectual property of the enterprise systems. It is a precondition for an effective business and system integration with existing processes and methodologies and adds control capabilities.

With specifications from Architecture Governance we define our main Architecture Management procedures for service-oriented enterprise software architectures: service strategy and life cycle management of software and system architecture artifact's state, service security, service testing and monitoring, service contracts, registries, service reuse, service ownership, definition, and versioning.

Figure 2. ESARC – Business & Information Reference Architecture.

The ESARC - Business & Information Reference Architecture in Figure 2 defines the link between the

enterprise business strategy and the resulting business and information design for supporting strategic initiatives. The Business & Information Reference Architecture provides a single source and comprehensive repository of knowledge from which corporate initiatives will evolve and link. This knowledge is model-based and is an integrated enterprise model of the business, which includes the organization and the business processes. The Business & Information Reference Architecture opens a connection to IT infrastructures, systems, as well as to software and security architectures. It provides integration capabilities for IT management, software engineering, service & operations management, and process improvement initiatives. The Business & Information Reference Architecture defines and models the business and information strategy, the organization, and main business requirements for information systems: key business processes, business rules, business products, and business control information.

The ESARC – Information Systems Reference Architecture in Figure 3 provides an abstract blueprint for the individual application architecture to be deployed. It adds specific interactions and specifies relationships to the core business processes of the organization. The OASIS Reference Model for Service Oriented Architecture [4] is an abstract framework which guides our ESARC reference architecture. The ESARC defines the abstract model for specific applications architectures and implementations.

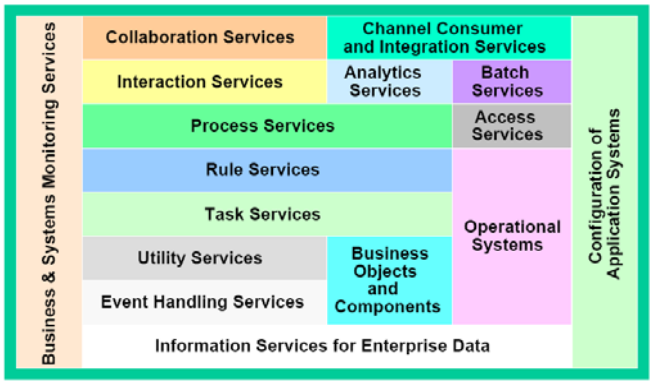

Figure 3. ESARC - Information Systems Reference Architecture.

In our ESARC – Information Systems Reference Architecture we have differentiated layered service types. The information services for enterprise data can be thought of as data centric components [14], providing access to the persistent entities of the business process. The capabilities of information services combine both elementary access to CRUD (create, read, update, delete) operations and complex functionality for finding/searching of data or complex data structures, like data composites or other complex-typed information. Close to the access of enterprise data are context management capabilities, provided by the technology architecture: error compensation or exception handling, seeking for alternative information, transaction processing of both atomic and long running and prevalent distributed transactions. Information services [14] and their related data architecture [2] are core company assets and should be close and centrally managed for reuse. Task services implement business capabilities related to specific actions of the business process. Task services could be own or third-party services. Usually task services don't manage state information directly, but work in cooperation with information services. The access to information services follows an acyclic graph - from top to down layers.

From [11] and [5], [6], [12] result important design rules for task services. Operations of task and entity services shouldn't have any knowledge about their process or interactive usage context. Task service operations [14] should be independent from users and sessions and should only implement business functionality. Authorization checks should be done outside of the business operations. Task and information services should use a transactional context, but their operations shouldn't implement by their own transactions. Task service operations should be usable both in batch and in online system transactions. Task services are used in process services - as multiple composites of services and should therefore be centrally managed high reusable assets. Rule services provide knowledge representation and processing capabilities for adaptable business product and business services. Rule services provide in addition flexible controls for agile business processes.

Process services [14] are long running services which compose task services and information services into workflows, to implement the procedural logic of business processes. Process services can activate rule services, to swap out a part of the potentially unstable gateway-related causal decision logic. Process services are frontend by interaction services or by specific diagnostic service and process monitoring services. Often process services manage distributed data and application state indirectly, by activating task and information services. Process services participate in atomic transactions only when they are activated from batch services. When processes services participate in human interaction workflows, they have to support long-running transactions where compensation of possible errors or exceptions happens in the business logic.

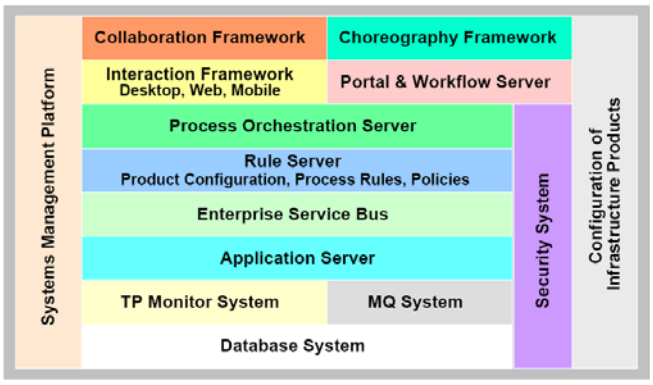

Figure 4. ESARC - Technology Reference Architecture.

The ESARC – Technology Reference Architecture in Figure 4 describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT

infrastructure, middleware, networks, communications, processing, and standards. The layers of the ESARC – Technology Reference Architecture and the layers of the ESARC – Information Systems Reference Architecture correspond to each other. The database system is the vendor supported database management system for handling enterprise data. We have included in our architecture stack the TP-Monitor system and have associated it with the database system. Optionally we have integrated a messaging system. The application server is the container environment architecture for objects and enterprise components. The enterprise service bus uses a flexible and standard-based messaging mechanism to interconnect services on a process and service execution platform. On top of the service bus we have placed the rule server, which is able to represent business rules and operate rule processing by building dynamic inference chains. The process orchestration server executes process services by calling suitable services of earlier mentioned types. For running interaction and collaboration services additional infrastructures are included in our stack of the technology reference architecture model: interaction frameworks, portal servers, workflow engines, and nowadays collaboration frameworks. Security services are part of an integral framework-based security system of standards and components and are impacted by mentioned services and distributed service technologies.

IV. VALIDATION AND RESULTS

Architecture assessments need to address the key challenges for companies during the built-up and management of service-oriented architectures in heterogeneous IT environments. Assessments of the SOA ability of standard software packages can be viewed additionally as a mean to engage with vendors on relevant challenges of SOA in practical use.

The basic structure of a working example is our questionnaire [1], which was based on our currently reported ESARC for the assessment and optimization of architecture artifacts. Our questionnaire is also close associated with our previous developed SOAMMI – a service-oriented enterprise architecture maturity framework [7], [1] and on adapted elements from [16]. Detailed working examples of the ESARC Reference Model, the SOAMMI Maturity Model and our Architecture Capability Patterns in action can be found in our current research paper [22].

We have synthesized the following 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. For example, 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.

 ISBN: 978-1-61208-152-6

V. CONCLUSION AND FUTURE WORK

Our original approach for architecture evaluation and optimization of service-oriented enterprise software architectures is based on ESARC - a special architecture reference model, an associated architecture metamodel and on architecture patterns. In our research we have motivated the necessity to extend both existing architecture reference models and service-oriented maturity models to accord to a clear metamodel approach due to the well understood and verified CMMI model. Our approach provides a sound basis from theory for practical evaluations of service oriented standard platforms in heterogeneous environments with four major global acting technology vendors. Future work has to consider conceptual work on both static and dynamic architecture complexity, and in connecting architecture quality procedures with prognostic processes on architecture maturity with simulations of enterprise and software architectures. Additional improvement idea deals with patterns for visualization of architecture artifacts and architecture control information to be operable on an architecture management cockpit. To improve semantic-based navigation within the complex space of EAM-visualization and service-oriented enterprise software architecture management and we are working on ontology models for the ESARC – The Enterprise Software Architecture Reference Cube.

ACKNOWLEDGEMENT

Many thanks to the SOA Innovation Lab initiative from Germany and Europe, a research and innovation network of information technology applicators, the technology vendors IBM, SAP, ORACLE, Microsoft, the associated consulting firms and scientific network for providing us the context for our research.

REFERENCES

[1] H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, and A. Zimmermann, "*Analyzing the SOA-ability of Standard Software Packages with a dedicated Architecture Maturity Framework*", EMISA 2010: October 7– 8, 2010 - Karlsruhe, Germany, GI-Edition - Lecture Notes in Informatics (LNI), P-172, 2010, pp. 131-143.

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

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

[4] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, OASIS "*Reference Model for Service Oriented Architecture*" 1.0, OASIS Standard, 12 October 2006.

[5] J. A. Estefan, K. Laskey, F. G. McCabe, and D. Thornton, OASIS "*Reference Architecture for Service Oriented Architecture*" Version 1.0, OASIS Public Review Draft 1, 23 April 2008.

[6] J. A. Estefan, K. Laskey, F. G. McCabe, and D. Thornton, OASIS "*Reference Architecture Foundation for Service Oriented Architecture*" Version 1.0, OASIS Committee Draft 02, 14 October 2009.

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

[8] A. Zimmermann, E. Ammann, and F. Laux, "*Pattern Catalog for Capability Diagnostics and Maturity Evaluation of Service-oriented Enterprise Architectures*", PATTERNS 2010 - The Second International Conferences on Pervasive Patterns and Applications, November 21-26, 2010 - Lisbon, Portugal, IARIA Proceedings of the PATTERNS 2010 Conference, pp. 13-19, 2010.

[9] L. Bass, P. Clements, and R. Kazman, "*Software Architecture in Practice"*, Second Edition, Addison Wesley, 2003.

[10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, „*Pattern-oriented Software Architecture"*, Wiley. 1996.

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

[12] T. Erl, "*Service Oriented Architecture"* Prentice Hall, 2005.

[13] The Open Group "*SOA Governance Framework*", August 2009.

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

[15] R. Kazman, M. Klein, and P. Clements, *"The Architecture Tradeoff Analysis Method (ATAM)"*, CMU/SEI-2000-TR-004, Carnegie Mellon University, Software Engineering Institute, 2000.

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

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

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

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

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

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

[22] A. Zimmermann, H. Buckow, H.-J. Groß, O.F. Nandico, G. Piller, and K. Prott, "*Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model*", IEEE-SCC2011: Washington DC – July 5-10, 2011, to be published.

 ISBN: 978-1-61208-152-6