

Pattern Catalog for Capability Diagnostics and Improvement of Service-oriented Enterprise Architectures

Alfred Zimmermann, Eckhard Ammann, Fritz Laux
Fakultät Informatik, business informatics research center
Reutlingen University
D-72762 Reutlingen, Germany

Email: {alfred.zimmermann | eckhard.ammann | fritz.laux}@reutlingen-university.de

Abstract—An original pattern catalog for capability diagnostics and optimization for change of service-oriented enterprise architectures is introduced. The current approaches for assessing maturity of software architectures were intuitively developed, having sparse meta model foundation and being rarely validated. This is a real problem because enterprise and software architects should know what is a successful path for introducing and changing service-oriented enterprise architectures.

Our contribution is to extend existing Service Oriented Architecture (SOA) maturity models to accord with a sound meta model approach 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 meta model-based synthesis and conception and was grounded on the current Open Group Architecture Framework (TOGAF) standard for enterprise architectures.

Applying the maturity framework in consecutive assessment workshops with global vendors of service-oriented platforms provides the base for developing our pattern catalog for capability diagnostics and for improvement of service-oriented enterprise architectures.

Index Terms—Pattern catalog; SOA; SOAMMI; CMMI; TOGAF; service-oriented enterprise architecture; capability and maturity diagnostics; assessment; architecture maturity; meta model integration; maturity framework.

I. INTRODUCTION AND RELATED WORK

Innovation oriented companies have introduced in recent years service-oriented architectures (SOA) to assist in closing the business and IT gap by delivering efficiently appropriate business functionality and integrating legacy systems with standard application platforms. Our approach of investigating the SOA ability of standard platforms in commercial use (see Buckow et al. [4]) assembles elements from convergent architecture methods and technologies like software related patterns as in Gamma et al. [11], Buschmann et al. [6], Fowler [10], and Buckl et al. [3], together with enterprise architecture management (EAM), SOA, and package based standard software applications. According to Alexander et al. [1] a pattern records the architecture decisions taken by many builders in many places over many years in order to resolve a particular problem.

The hypothesis of our research [18] is as follows:

- 1) The Capability Maturity Model Integration (CMMI) [7] is well known as suitable framework to assess software processes, nevertheless the meta model of CMMI can be extended to evaluate capabilities for change of enterprise and service-oriented architectures.
- 2) The idea of software patterns could be applied consistently for both capability diagnostics and for improvement of architecture areas starting from solid evaluation results of enterprise and service-oriented architectures.

The Open Group Architecture Framework (TOGAF) [17] as the current standard for enterprise architecture provides the basic blueprint and structure for our architecture domains:

- Architecture Strategy and Management,
- Business Architecture,
- Information Architecture,
- Application Architecture,
- Technology Architecture,
- Service & Operation Architecture, and
- Architecture Realization.

The Architecture Capability Maturity Model (ACMM) [2] 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.

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, 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 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 distinguished: 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.

The SOA Maturity Model from Sonic [16] 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.

The SOA Maturity Model of ORACLE [15] 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.

Service-oriented architecture (SOA) 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. A basic positioning into fundamental SOA concepts, technologies and case studies is offered by Erl [8] and for SOA-aspects on Enterprise Application Integration in the book of Krafzig et al. [13]. 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 but consistently used within a Model Driven Architecture (MDA) [14] 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 an abundance set of specifications and standards developed by open standard organizations like W3C, OMG, OASIS, and The Open Group.

In the following Section II we provide our model synthesis

for SOA Maturity Model Integration (SOAMMI) by assembling and shifting basic maturity model elements into our conceptual model for architecture maturity diagnostics. Based on the meta model of SOAMMI, Section III presents examples from our pattern catalog, which we have developed to assist in diagnostics and optimization of service-oriented enterprise architectures. Section IV states our conclusions and validation results from assessments and presents some ideas for future work.

II. SOA ARCHITECTURE MATURITY FRAMEWORK (SOAMMI)

The aim of the SOAMMI was developed to provide a holistic framework to assess service-oriented enterprise architectures. The development process consisted of two interwoven phases. First, CMMI [7] was transformed from an assessment framework for software processes into a specific framework [18] to diagnose systematically the maturity of enterprise and software architectures.

Second, our maturity assessment approach was conducted by SOA applicators having experience in specific business domains and analyzing SOA vendor products for heterogeneous environments of legacy and standard applications. For the analysis we used assessment criteria, maturity domains, architecture capabilities, and level rankings from state of art SOA maturity models as described in [2] [12] [16] [15]. In addition specific architecture elements from [17] and [9] were selected to develop our architecture maturity model .

The SOAMMI architecture maturity framework introduces new architecture areas and organizes them within extended architecture domains, which are mainly based on TOGAF. Our intention was to leave most parts of the original CMMI meta model untouched and to extend the CMMI logic carefully.

The meta model of SOAMMI in Figure 1 has similarities with the CMMI meta model and defines additional specific elements, which are defined in the next sections for our architecture evaluation purpose. The extension uses maturity levels to measure the architecture maturity of vendor products in respect of requirements from customer oriented domain models:

- Maturity Level 1: Initial
 - Vendor service architecture is not performed or is incomplete or with no or initial coverage only
 - Architecture is unpredictable and poorly controlled
 - Initial service architecture methods and knowledge transfer about services and architectures
- Maturity Level 2: Managed
 - Vendor service architecture is managed, having medium completeness and coverage
 - Vendor supports learning about architectures and corrective actions are taken when necessary
 - Vendor service architecture is institutionalized within own products
- Maturity Level 3: Defined

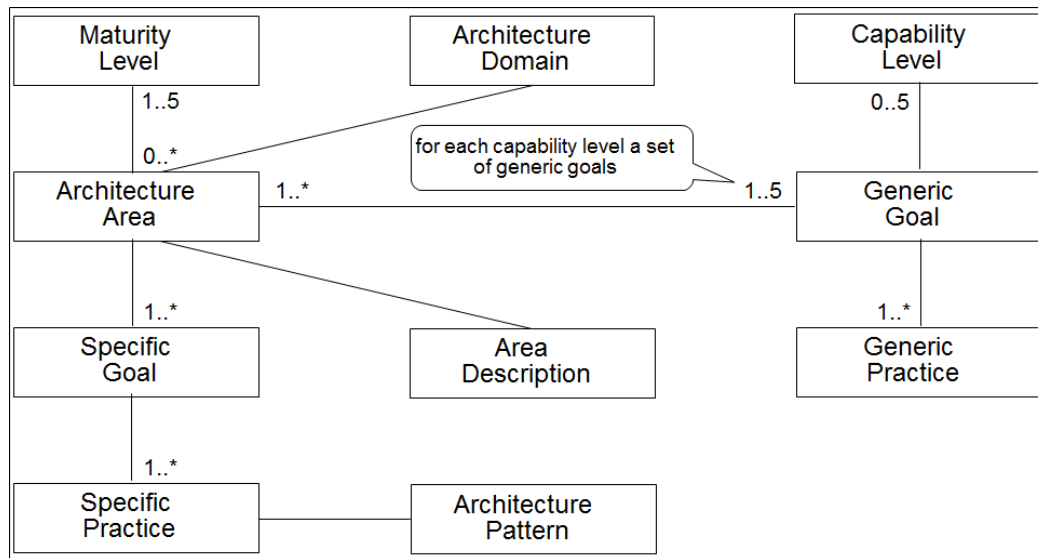


Fig. 1. SOAMMI Meta Model

- Vendor service architecture is defined, having large, increasing completeness and coverage
- Customer service architecture is agile tailored from standard vendor architecture
- Vendor supports service strategy, architecture governance, methods and tools
- **Maturity Level 4: Quantitatively Managed**
 - Architecture artifacts and benefits are measured at vendor and customer side
 - Architecture is based on measured parameters from monitored business services
 - Causes of special variations are addressed
- **Maturity Level 5: Optimizing**
 - Defects are prevented at customer and vendor side
 - Innovations are added based on a vendor / client mutual roadmap
 - Change is expected, not feared and improvements are proactive.

Architecture domains were derived mainly from TOGAF [17], where they are used as specific architecture subtypes and corresponding phases of the ADM (Architecture Development Method). The top level structure of SOAMMI is organized by the following orthogonal architecture domains: Architecture Strategy and Management, Business Architecture, Information Architecture, Application Architecture, Technology Architecture, Service & Operation Architecture, and Architecture Realization.

Architecture areas are correspondent parts of process areas from CMMI. We have defined 22 specific architecture areas of SOAMMI in Figure 2 - fitting our architecture diagnostic scope, but different from CMMI - and structure them according to standard architecture maturity levels in line with the mentioned architecture domains. Each of the 22 delimited architecture areas are accurately described by a name and a

short identification, and later on supplemented by a detailed description.

The following example of a standardized form shows in detail two specific architecture areas of the Business Architecture Domain, which were structured similarly to process areas of CMMI:

A. Architecture Area: BPS Business Products & Services

Purpose: Structure, design, model, and represent business products and associated business services, which are necessary to support modeled products.

Maturity Level: 3

Specific Goals (SG) and Specific Practices (SP):

- **SG 1: Model Business Products as Origin of Business Processes**
 - SP 1.1 Structure business products within product lines
 - SP 1.2 Design business products by defining product structures and product rules
 - SP 1.3 Model and represent business products
- **SG 2: Model Business Services associated with Business Products**
 - SP 2.1 Structure business services according product types
 - SP 2.2 Design business services by defining service structures and service levels
 - SP 2.3 Model and represent business services

B. Architecture Area: BPR Business Processes & Rules

Purpose: Structure, design, model, and represent business value chains and business processes to support modeled products and services.

Maturity Level	Architecture Domain	Architecture Area	ID
5: Optimizing	Architecture Strategy & Management Architecture Strategy & Management	Architecture Innovation & Development Causal Analysis and Resolution	AID CAR
4: Quantitatively Managed	Architecture Strategy & Management Architecture Strategy & Management	Organizational Performance Monitoring Quantitative Architecture Management	OPM QAM
3: Defined	Architecture Strategy & Management Architecture Strategy & Management Architecture Strategy & Management Business Architecture Business Architecture Information Architecture Service & Operation Architecture Architecture Realization	Enterprise Arch. Management Architecture Requirements Development Architecture Governance Business Capabilities & Services Business Products & Services Business Information Alignment Service Design & Transition Architecture Contracts	EAM ARD GOV BCS BPS BIA SDT ACO
2: Managed	Architecture Strategy & Management Business Architecture Business Architecture Information Architecture Information Architecture Application Architecture Application Architecture Technology Architecture Technology Architecture Architecture Realization	Architecture Requirements Management Business Domains & Capabilities Business Processes & Rules Data Entities & Components Business Objects System Services & Capabilities System Domains Platform Services Technology Services & Capabilities Architecture Standards & Compliance	ARM BDC BPR DEC BOB SSC SDO PFS TSC ASC
1: Initial			

Fig. 2. SOAMMI Architecture Areas and Maturity Levels

Maturity Level: 2

Specific Goals (SG) and Specific Practices (SP):

- SG 1: Model Business Value Chains as Root of Business Processes
 - SP 1.1 Identify business value for business operations
 - SP 1.2 Structure value chains
 - SP 1.3 Optimize business considering customer channels and supplier networks
- SG 2: Model and Optimize Business Processes
 - SP 2.1 Identify business activities for business processes: system activities, user interaction activities, manual activities
 - SP 2.2 Structure business processes for business roles and organizational units
 - SP 2.3 Define business workflows and business process rules
 - SP 2.4 Model and represent business processes
- SG 3: Model and Represent Business Control Information
 - SP 3.1 Identify and represent control information for product monitoring
 - SP 3.2 Identify and represent control information for process monitoring

The sketched Architecture Area BPS Business Products & Services was mapped as a premium architecture discipline to the higher Maturity Level 3 on top of the basic Architecture Area BPR Business Processes & Rules, which was allocated with the basic Architecture Maturity Level 2.

III. ENTERPRISE ARCHITECTURE PATTERNS

Our patterns for enterprise and service-oriented architectures consist of a set of methods which use best practices for diagnosing malfunctions and improving enterprise and information systems architectures. We have derived the methods from the structures of the metamodel of SOAMMI presented in Section II. Patterns, as described originally by Alexander et al. [1] are collections of best practices which are based on representing compactly core causalities for problem solving starting with a description of a recurring problem directing us to a standardized solution. Additionally to the core causalities for problem solving each pattern approach has added important but divergent extensions resulting in specific canonical forms for describing these patterns like in [11] [6] [10] [3].

Our pattern catalog for diagnostics and improvement of enterprise and service-oriented architectures organizes the collection of patterns according to the SOAMMI metamodel structures:

- Architecture Domains
- Architecture Areas
- Problem Descriptions associated with Specific Goals, and
- Solution Elements of the patterns connected to relate Specific Practices.

Linking solution elements to specific practices of the SOAMI Framework enables concrete solutions for diagnostics and improvement of service-oriented enterprise architectures. This diagnostic and improvement knowledge is no design knowledge, it is rather a procedural knowledge based on standards, best practices, and assessment experience for software

and enterprise architectures. It is therefore both concrete and specific for setting the status of service-oriented enterprise architectures, and helping to establish an improvement path for change.

Patterns of our catalog show what to assess. Our patterns aim to represent diagnostic and improvement procedural knowledge to support cooperative assessment and improvement work of many people over many years in cyclic assessments of service-oriented enterprise architectures.

Associated with this pattern catalog we have set up an assessment process showing how to assess architecture capabilities. This process is based on a questionnaire for assessment workshops providing concrete questions and answer types, and helping to direct and standardize the related assessment process.

Additionally, we have included process methods for workshops, result evaluations, improvement path information for technology vendors and for application organizations, as well as change support and innovation monitoring instruments.

Based on the two Architecture Area examples from Section II- BPS Business Products & Services Architecture and BPR Business Process & Rules - we are deriving exemplarily the related subset of architecture patterns from the mentioned Specific Goals:

- 1) Model Business Products as Origin of Business Processes
- 2) Model Business Services associated with Business Products
- 3) Model Business Value Chains as Root of Business Processes
- 4) Model and Optimize Business Processes
- 5) Model and Represent Business Control Information.

We have chosen the reduced canonical form consisting of a succinct representation of the core causalities of our diagnostic and improvement patterns denominating consciously only the problem and the solution part as basic elements of our diagnostic and improvement patterns for service-oriented enterprise architectures. This basic canonical form of our currently used patterns is extendable by additional parts like contexts, examples, explanations, linked patterns, and others. We note that our diagnostic and improvement patterns are basically process patterns for enterprise architecture management and are therefore not fine granular classical design patterns. The following examples show a concrete extract from our set of 38 diagnostic and improvement patterns.

A. Example Pattern 1: Model Business Products as Origin of Business Processes

Problem: How can we structure, design, model, and represent business products as an origin for modelling business processes?

Solution:

- Structure business products within product lines
- Design business products by defining product structures and product rules

- Model and represent business products

B. Example Pattern 2: Model Business Services associated with Business Products

Problem: How can we structure, model, and represent business services needed to support business products?

Solution:

- Structure business services according product types
- Design business services by defining service structures and service levels
- Model and represent business services

C. Example Pattern 3: Model Business Value Chains as Root of Business Processes

Problem: How can we structure, optimize and represent value chains as roots for business process modelling?

Solution:

- Identify business value for business operations
- Structure value chains
- Optimize business considering customer channels and supplier networks

D. Example Pattern 4: Model and Optimize Business Processes

Problem: How can we structure, optimize and model business processes, related workflows, and business process rules?

Solution:

- Identify business activities for business processes: system activities, user interaction activities, manual activities
- Structure business processes for business roles and organizational units
- Define business workflows and business process rules
- Model and represent business processes

E. Example Pattern 5: Model and Represent Business Control Information

Problem: How can we model and represent business monitoring and control information?

Solution:

- Identify and represent control information for product monitoring
- Identify and represent control information for process monitoring.

The basic causality of our architecture pattern allows us to navigate in two directions: from the problem statement to the solution and backwards from the expected solution to the problem. From this navigation possibilities follow two important problem solving strategies for:

- Diagnostic: for verifying suggested solutions and defining the problem (from pattern solution to the pattern problem statement)

- Improvement: for identifying suitable solution elements for a given problem (from the pattern problem statement to the solution statements).

We have identified and distinguish a set of 38 Patterns in the context of 7 Architecture Domains and 22 Architecture Areas. The full list of patterns and its catalog structure follow the SOAMMI Framework [18]. From each mentioned 7 architecture domains we are sketching typical examples for enterprise patterns from our pattern catalog suited for architecture diagnostics and improvements:

- 1) Architecture Domain: Architecture Strategy and Management
 - Architecture Area: GOV Architecture Governance
 - Manage and control architectures of information systems
 - Support architecture governance
- 2) Architecture Domain: Business Architecture
 - Architecture Area: BPR Business Processes & Rules
 - Model business value chains as root of business processes
 - Model and optimize business processes
 - Model and represent business control information
- 3) Architecture Domain: Information Architecture
 - Architecture Area: BIA Business Information Alignment
 - Determine alignment of business and information architecture
- 4) Architecture Domain: Application Architecture
 - Architecture Area: SDO System Domains
 - Define and model a system domain map
- 5) Architecture Domain: Technology Architecture
 - Architecture Area: PFS Platform Services
 - Identify and model platform services from basic infrastructure
 - Determine fitness of vendor platform services
- 6) Architecture Domain: Service & Operation Architecture
 - Architecture Area: SDT Service Design & Transition
 - Identify and model services to support information systems and enable transition of services for support by service providers
 - Ensure service management offering for SOA
- 7) Architecture Domain: Architecture Realization
 - Architecture Area: ASC Architecture Standards & Compliance
 - Manage and control architecture standards and ensure compliance of architectures with standards
 - Support architecture standards, methods, and tools.

The practical benefits of our pattern catalog in the reduced canonical form is documented by the successful use as guideline for questionnaire design for two major capability assessments of service-oriented vendor technology architectures.

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 all relevant challenges of SOA in practical use.

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.

Following these ideas, the basic structure of our questionnaire [5] was taken from SOAMMI architecture areas with one or more questions per specific goal respectively the problem statement in our diagnostic and improvement patterns. User requirements have been consolidated and mapped against specific goals. Wherever no user requirements could be mapped, specific practices or solution elements in our patterns 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.

The assessment process takes about 3 months in total to complete for each software technology provider. The first step is a Pre-Workshop (2-3 hours) 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 (a consortium of SOA applicators, consulting companies, system integrators, and academic consultants) then prepares the summary of the findings and presents these 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.

IV. CONCLUSION

A pattern catalog for diagnosing capabilities and improvement of organizational maturity of enterprise and service oriented architectures has been introduced. In this paper we have motivated the necessity to extend existing SOA maturity models to accord to a clear meta model approach due to the well understood and verified CMMI model. Based on the related work to CMMI, which is an assessment and improvement model for software processes, we have transformed and developed suitable models for the evaluation of SOA capability and maturity. Our specific architecture evaluation approach from the SOAMMI framework was founded on the current TOGAF standard for enterprise architectures. SOAMMI - The SOA Maturity Model Integration is the result of a meta model based conception and synthesis to provide a sound basis for practical evaluations of service oriented standard platforms in

heterogeneous environments. Additionally a SOAMMI dashboard was developed to support practical assessment processes, which were aligned both with the SCAMPI process for CMMI and with empirical questionnaire and interview methods. The presented SOAMMI framework was validated in consecutive assessment workshops with two global vendors of service-oriented platforms and has provided transparent results for subsequent changes on service oriented product architectures and related processes. Our empirical validation and optimization of the presented maturity framework for its future usage is an ongoing process, which has to be synchronized with future cyclic evaluations of SOA platforms and their growing number of services. Extended validations of customers of service oriented technologies are planned for the next phase of our framework research and development. An idea towards a framework for individual enterprises is to generically extend the architecture areas to provide distinct views for architecture maturity diagnostics of vendor architectures and to support diagnostics for customers' and suppliers' abilities to handle service-oriented application architectures. Future work additionally has to consider conceptual work on both static and dynamic architecture complexity, and in connecting architecture diagnostic procedures with prognostic processes on architecture maturity with simulations of enterprise and software architectures.

REFERENCES

- [1] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. (1977): A Pattern Language, Oxford University Press New York
- [2] ACMM (2007): Architecture Capability Maturity Model - The Open Group. URL: <http://www.opengroup.org/architecture/togaf9-doc/arch/chap51.html>
- [3] Buckl, S., Ernst, A.M., Lankes, J., Matthes, F. (2008): Enterprise Architecture Management Pattern Catalog. Technical Report TB 0801, sebis Technische Universität München
- [4] Buckow, H., Groß, H.-J., Piller, G., Prott, K., Willkomm, J., Zimmermann, A. (2010): Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes. Proceedings of the 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010) Workshop at the Software Engineering 2010 Conference in Paderborn, February, 22 - 23, 2010
- [5] Buckow, H., Groß, H.-J., Piller, G., Prott, K., Willkomm, J., Zimmermann, A. (2010): Analysing the SOA ability of Standard Software Packages with a dedicated Architecture Maturity Framework. (accepted paper) EMISA 2010 - Entwicklungsmethoden für Informationssysteme und deren Anwendung, Karlsruhe, FZI Forschungszentrum Informatik, October, 7 - 8, 2010
- [6] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. (1996): Pattern-oriented Software Architecture. Wiley
- [7] CMMI (2006): CMMI for Development. Version 1.2, Carnegie Mellon University, Software Engineering Institute. URL: <http://www.sei.cmu.edu/reports/06tr008.pdf>
- [8] Erl, T. (2005): Service Oriented Architecture. Prentice Hall
- [9] essential (2009): The Essential Architecture Project. URL: <http://www.enterprise-architecture.org/>
- [10] Fowler, M. ed. (2003): Patterns of Enterprise Application Architecture. Addison Wesley
- [11] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994): Design Patterns. Addison Wesley
- [12] Inaganti, S., Aravamudan, S. (2007): SOA Maturity Model. BP Trends, April 2007. URL: <http://www.bptrends.com/publicationfiles/04-07-ART-The%20SOA%20MaturityModel-Inagantifinal.pdf>
- [13] Krafzig, D., Banke, K., Slama, D. (2005): Enterprise SOA. Prentice Hall
- [14] MDA (2003): Model Driven Architecture (MDA) Guide OMG 1.0.1. URL: <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [15] ORACLE SOA Maturity Cheat Sheet (2009) URL: <http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet>
- [16] Sonic (2005): SOA Maturity Model. Sonic Software Corporation, AmberPoint Inc., Systinet Corporation. URL: http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf
- [17] TOGAF (2009): TOGAF The Open Group Architecture Framework. Version 9. URL: <http://www.opengroup.org/architecture/togaf9-doc/arch/>
- [18] Zimmermann, A. (2009): SOAMMI - SOA Maturity Integration Model - Conceptual Framework. Research Study from Sabbatical at Daimler AG with SOA Innovation Lab, July 2009