Enterprise Architecture Ontology for Services Computing

Alfred Zimmermann
Reutlingen University, Faculty of Informatics
Architecture Reference Lab of the
SOA Innovation Lab, Germany
alfred.zimmermann@reutlingen-university.de

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

***Abstract* – Enterprise services computing is the current trend for powerful large-scale information systems, which increasingly converge with cloud, computing environments. In this paper, we propose an original ontology-based Architecture Classification Framework for supporting cyclic architecture evaluations and optimizations of enterprise systems based on service-oriented architectures: ESARC - Enterprise Services Architecture Reference Cube. ESARC provides a standardized and normative classification framework for important architecture artifacts of service-oriented enterprise systems. 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, pattern, metamodel, or ontology foundation. Cyclic assessments of complex service-oriented systems and architectures should produce comparable evaluation results. Today architecture evaluation findings are hardly comparable. Our current idea and contribution is to extend the basic architecture classification framework of ESARC from our previous research by developing specialized metamodels and ontologies for a coherent set of reference architectures, to be able to support machine-based architecture diagnostics and optimizations in enterprise services computing.**

Keywords – service-oriented architecture; enterprise services; enterprise architecture; ESARC; reference model; refernce architecture; ontology; classification framework; diagnostics.

I. INTRODUCTION

Since recent years, innovation oriented companies have introduced service-oriented computing paradigms and combine them with traditional information systems. As the architecture of service-oriented enterprise systems becomes more and more complex, and we are going rapidly into cloud computing scenarios, we need a new and improved set of methodological well-supported instruments and tools for managing, diagnosing and optimizing complex enterprise service-oriented information systems. Service-oriented systems close the business – information technology (IT) gap by delivering efficiently appropriate business functionality and integrating legacy systems with standard application platforms.

Our research work and current innovation practice is about new methods for architecture assessments, and architecture diagnostics, monitoring and optimization. We intend to provide a unified and consistent methodology for enterprise architecture management for service-oriented and cloud computing systems. Our research results are currently validated and extended, and are further to be used for assessments and for integral monitoring of heterogeneous business processes and complex integrated information systems in commercial use [1] by members of the SOA Innovation Lab in Germany and Europe.

Our new introduced approach of an enterprise architecture ontology for services computing is a work in progress research, which is ongoing extended to cover the integral scope of the existing, but still evaluating, classification framework of our ESARC–Enterprise Services Architecture Reference Cube. In assessing the quality of implemented SOA vendor platforms and the integral architecture of service-oriented enterprise systems, we face the problem of not having real comparable evaluation findings from consecutive (cyclic) assessments. Only an architecture classification framework, which sets a relative standard of comparison, makes it possible to track the improvement path of different enterprise services and systems, their architectures and related technologies.

The current state of art research in enterprise services and cloud computing research lacks an integral understanding of architecture classification and semantic representation of service-oriented and cloud computing enterprise systems. Our previous assessment findings were done without an architecture reference model. As a result multiple evaluations of enterprise systems with service-oriented architectures were blurry and hardly comparable within a series of consecutive architectural tests and therefore have 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 for real business enterprise system environments. In this paper, we disclose our ontology-based approach toward a unified classification framework for enterprise architectures of services and cloud computing systems.

The novelty in our current research paper for the ESARC comprises new aspects and extended ideas for Enterprise Architecture Management (EAM) for Services & Cloud

 ISBN: 978-1-61208-215-8

Computing (SCC). As worked out in this paper, metamodels and related ontologies for ESARC - Enterprise Services Architecture Reference Cube - are the useful extension and integration aid for a holistic set of reference architectures, which we have derived from the ESARC classification framework and the Open Group's standard on Service-oriented Architecture Ontology. Our architecture ontology should provide a base for semantic-supported navigation and automatic inference in architecture diagnostics.

In the following Section II, we present the main view of our original developed ESARC architecture classification framework. We define interrelating reference architecture domains of service-oriented enterprise systems, as part of an architecture layer model, which we built from integrated standards. In Section III, we introduce correlated architecture metamodels and our developed architecture ontology in the context of standards. In Section IV, we set the base for our study from the state of art and other related work. Finally, Section V summarizes our conclusion and gives ideas from current research and for future work.

II. ENTERPRISE ARCHITECTURE REFERENCE MODEL

ESARC – the Enterprise Services Architecture Reference Cube – is an integral and continually growing ontology-supported architecture classification framework [2] to be used by enterprise and software architects, to define, structure, verify, and improve service-oriented enterprise and software architectures in a standard way. In order to specify our innovative enterprise and software architecture assessment method, we used a metamodel-based approach [3] for capability evaluations of architecture elements and their main relationships. For this purpose, we have extended, integrated and adapted elements from convergent architecture methods, architecture patterns [4] and [5], related standards and reference models from the state of art.

ESARC is an abstract architecture classification framework [3], which defines an integral view for main interweaved architecture types. ESARC was derived primarily from state of art research and standards [6] and [7], and from architecture frameworks like TOGAF [8], essential [9], the service model of ITIL, and from resources for service-oriented computing [10], [11], and [12]. The aim of the ESARC architecture classification framework is to be universally applicable in cyclic, repeatable and comparable architecture evaluations and structural optimizations of enterprise and software architectures for services and cloud computing. ESARC abstracts from a concrete business scenario or from specific technologies. The main focus of our present paper is to provide exemplarily a detailed view for the three main interdependent reference architecture views of ESARC: Business & Information Reference Architecture, Information Systems Reference Architecture, and the Technology Reference Architecture.

The Open Group Architecture Framework (TOGAF) [8] is the current standard for enterprise architecture and provides the basic blueprint and structure for the service-oriented enterprise software architecture domains. ESARC follows the main architecture domains of TOGAF and extends them substantially and in a unique way to a unified architecture classification framework. ESARC sets a standardization framework for cyclic diagnostics and optimizations of the following interrelated views of the reference architecture: Architecture Governance, Architecture Management, Business & Information Reference Architecture, Information Systems Architecture, Technology Architecture, Operation Architecture, Security Architecture, and Cloud Services Architecture.

The Architecture Governance and Management framework organizes main architecture types, like the Business & Information Architecture, the Information Systems Architecture, and the Technology Architecture. The architecture governance [12] cycle sets the abstract governance frame for concrete architecture activities within the enterprise software and product line development. The architecture governance cycle specifies constitutive management activities: plan, define, enable, measure, and control. The second aim of architecture governance is to set rules for architecture to comply with internal and external standards. Policies for governance and decision definition are set, to allow a standardized and efficient process for architecture decisions inside the enterprise architecture organization. Because enterprise and software architects are acting on a sophisticated connection path (from business and IT strategy to the realization of an architecture landscape of interrelated business domains, applications and technologies), architecture governance has to set rules for the empowerment of software architecture staff, defining structures and procedures of an architecture governance board, and setting rules for communication.

The ESARC - Business & Information Reference Architecture, as set in [3], extends the Business Architecture from TOGAF [8] and defines the link between the enterprise business strategy and the integral 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, like key business processes, business rules, business products, and business control information.

The ESARC – Information Systems Reference Architecture from [3] provides an abstract blueprint for the individual service-oriented 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 [13] is an abstract framework, which defines significant relationships among a small set of unifying architectural concepts for services computing. The reference model guides our correlating ESARC reference architectures, as in [14] and [15]. ESARC defines the abstract model for specific application architectures and implementations, which are in conformity with [7] and the Open Group's architecture standards [13], [14], and [15].

In ESARC – Information Systems Reference Architecture we have differentiated layered service types, inspired from [16]. The information services for enterprise data can be thought of as data centric components, 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.

Process services [16] 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.

The ESARC – Technology Reference Architecture [3] describes the abstract 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. 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.

III. ENTERPRISE ARCHITECTURE ONTOLOGY

We have developed exemplarily metamodels and related ontologies seeded by a student research project [17] for the following main architecture domains from ESARC, as a starting and extendable set of work results: Business & Information Reference Architecture, Information Systems Reference Architecture, and the Technology Reference Architecture. Metamodels are used, as standardized in [18], to define architecture model elements and their relationships for the reference architectures of ESARC. Metamodels define models of models. In our approach for architectural modeling, as in [2], [3], and [5], we use metamodels as an abstraction for architectural elements and relate them to architecture ontologies.

The Reference Model for Service Oriented Architecture of OASIS [13] is an abstract framework, which defines basic generic elements and their relationships of a service-oriented architecture. This reference model is not a standard, but provides a common semantic for different specialized implementations. Reference models are, as in [13], abstract conceptual models of a functional decomposition of model elements together with the data flows between them.

Reference architectures, in [14] and [15], are specialized models of a reference model. It is a composition of related architectural elements, which are build from typed building blocks as the result of a pattern-based mapping of reference models to software elements. Architecture patterns, in [4], [7] and [10], are human readable abstractions for known architecture quality attributes, and represent standardized solutions, considering architectural constraints for given recurring problems.

The technical standard of Service-oriented Architecture Ontology from [6] defines core concepts, terminology, and semantics of a service-oriented architecture in order to improve the alignment between the business and IT communities. Following stakeholders are potential users of the SOA ontology, related architecture metamodels, as well as concrete architectural building blocks: business people and business architects, architects for the information systems and software architecture, architects for the technological infrastructure, cloud services architects and security architects.

In our understanding architecture ontologies represent a common vocabulary for enterprise architects who need to share their information based on explicitly defined concepts. Ontologies include the ability to infer automatically transitive knowledge. Our developed ontology for ESARC has some practical reasons: share the common understanding of the ESARC Architecture domains and their structures, reuse of the architectural knowledge, make architectural requirements, structures, building blocks explicit and promote reusability of architectural artifacts, separate the architectural knowledge according orthogonal architectural domains, classify, analyze, diagnose enterprise systems according to the service-oriented reference architecture od ESARC.

For our purpose, an ontology is, as in [19], a formal and explicit description of shareable and automatically navigable concepts of our architectural domain. For modeling purposes we are using UML class diagrams to represent concepts, and we are describing the attributes as properties (sometimes called roles) and role or property restrictions as facets. This structure of an ontology constitutes together with the instances of these concepts the knowledge base. Practically the knowledge base is a growing structure, which starts with the basic concept structures and is enlarged by a more or less amount of growing number of instances.

The SOA Ontology in [6] is represented in the Web Ontology Language (OWL) [20]. The ontology models the core concepts of SOA as classes and properties. The SOA ontology includes in addition natural language description of main concepts and relationships UML diagrams, which show graphically the semantic concepts as classes and the properties as UML associations. The intent of the UML diagrams are for explanations only, but are helpful constructs for understanding the modeled domain of SOA architecture and more concise than the more spacious formal descriptions in OWL. The SOA ontology defines the relations between semantic concepts, without mentioning the exact usage of these architecture concepts. To illustrate the SOA ontology the standard uses examples and descriptions of these in natural language.

The two core concepts of the SOA ontology in Figure 1, as in [6], are: System and Element. These two core concepts are generic and often used concepts to define a composite structure of systems that have elements. These abstract meaning of systems and elements is used in different specific architecture modeling situations. An example of an architectural element is ESB – the Enterprise Service Bus, which is an integration infrastructure for cooperating services. With the concept of *Element* the technical standard associates following core properties: *uses* and *usedBy* as well as the properties *representedBy* and *represents*. The technical standard of SOA Ontology defines additionally other concepts of the SOA Ontology like *HumanActor*, *Task*, *Service*, *ServiceContract*, *Effect*, *ServiceInterface*, *InformationType*, *Composition*, *ServiceComposition*, *Process*, *Policy*, and *Event*.

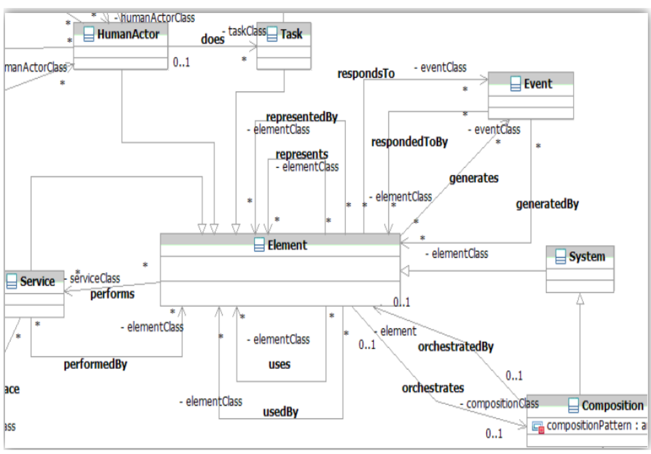

Figure 1. Open Group – Service-Oriented Architecture Ontology.

Element is the central generic service concept on which specialized model elements of ESARC – Enterprise Services Architecture Reference Cube – are constructed.

The metamodel of the ESARC-Business & Information Reference Architecture consists of specialized concepts, which are represented with associations and are generically linked using "is-a"-relationships with the generic concepts like Element and Composition from the Open Group's SOA Ontology [6].

To validate the developed metamodel from Figure 2, we modeled an instantiation scenario of the usage domain of a virtual travel agency. So the ontology was applied as an example to a particular domain by adding class instances of things in our test domain. A particular application, which is based on the ESARC Ontology and the generic SOA Ontology, can add new application-specific classes and properties. The result of these model concepts is a formal representation of the ESARC architecture-types and can be used as an aid to automatically navigate and infer architectural knowledge.

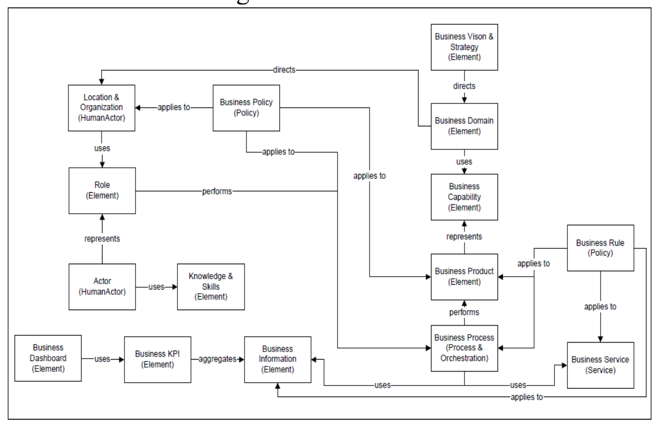

Figure 2. ESARC – Metamodel of Business & Information Reference Architecture.

We have developed the ESARC Ontology as in [20] and defined ontology concepts for ESARC using the ontology editor Protégé [21]. We have merged our specialized ESARC Ontology, as in [17], with the more generic SOA Ontology from [6]. The so-called *Asserted View* from Protégé in Figure 3 shows the *is-a-relationship* between specific concepts of the Business & Information Reference Architecture and the Open Group's generic SOA Ontology Reference Architecture.

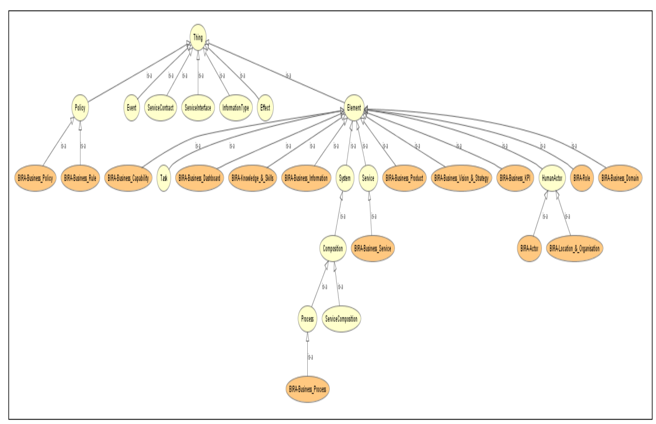

Figure 3. ESARC – Ontology of Business & Information Reference Architecture.

The terminal concepts are specific concepts of ESARC. In contrast we are representing the linked generic concepts of the SOA Ontology on the top of the diagram in Figure 3. Additionally, we determined knowledge properties for the modeled ontology concepts of ESARC. Using the developed ESARC Ontology, we can navigate in the multidimensional space of enterprise architecture management structures and enable a future research effort semantic-supported navigation for architects as well as a base for intelligent inference along specific inference chains. In addition, we have planned to add visualizations for these ontology concepts, as part of a sematic-supported architecture management cockpit.

IV. Related Work

Our research is based on formal architecture concepts from [7] and their relationships: software architecture, reference architecture, reference model, and architecture patterns. A reference model for SOA [13] is a generic fundamental model 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 [7] and [14], [15] is the result of a mapping of an architecture reference model to software elements and contains the related fundamental relationships between them.

Architecture ontologies are quite new study objects. Related and fundamental work on ontologies with their development processes and tools, as in [19], [20], and [21], will allow a better understanding of the modeled domain of enterprise services computing, can help to organize complexity in categories of interrelated concepts, and are an efficient and machine-understandable representation for the modeled classifications of concepts. Ontologies provide an aid both for software architects as well as for automatic inference procedures, to enable diagnostics and improvements within a predefined classification framework, which is defined by formal represented ontology concepts.

The Open Group's SOA Ontology in [6] is the fundamental work for of our ESARC Ontology. This basic ontology was the seminal work for our research. The ontology contains concepts and properties from the domain of service-oriented architectures. Formal OWL definitions are supplemented by text explanations and by UML class diagrams for the related models of the ontology. These diagrams and models are intended only for explanations of the formal OWL representations. We have done additional experimental work in a long-term student research project [17] to model exemplarily related ontologies for three main reference architectures for ESARC. Based on this work we are currently extending our ontology modeling and research to support multidimensional architecture representations and inference processes for diagnostics and optimizations of software architectures in enterprise services computing and extending our work for cloud computing, as in [22] and [23].

Service-oriented architecture SOA [11] 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. 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 Web technology during the last years and produced abundance specifications and standards as in [13], [14], [15], and [6], [12], [18], 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 [16] and [10].

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 [24] 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 [2].

The main scope of the intuitively specified Architecture Capability Maturity Model (ACMM) [25] framework from TOGAF is the evaluation of enterprise architectures in internal enterprise architecture assessments.

The SOA Maturity Model in [26] considers intuitively multidimensional aspects of a SOA.

The SOA Maturity Model from Sonic [27] 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 [28] 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.

V. Conclusion and Future Work

Our 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 we are working on ontology

 ISBN: 978-1-61208-215-8

models for the ESARC – The Enterprise Software Architecture Reference Cube.

ACKNOWLEDGEMENT

This work was initiated and partially supported by the SOA Innovation Lab, a major research and innovation network on Enterprise Architecture Management for Services and Cloud Computing in Germany and Europe.

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, 2010.

[2] 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.

[3] A. Zimmermann, H. Buckow, H.-J. Groß, F. O. Nandico, G. Piller, K. Prott, "*Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model*", in SCC 2011 - IEEE International Conference on Services Computing, July 4-9, 2011, Washington DC, USA, IEEE Proceedings of the SCC 2011, ISBN 978-0-7695-4462-5/11, IEEE Computer Society USA, pp. 592-599, 2011.

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

[5] A. Zimmermann, F. Laux, R. Reiners, "*A Pattern Language for Architecture Assessments of Service-oriented Enterprise Systems*", in PATTERNS 2011 - The Third International Conferences on Pervasive Patterns and Applications, September 25-30, 2011 Rome, Italy, ISBN 978-1-61208-158-8, IARIA Proceedings of PATTERNS 2011, pp. 7-12 (2011).

[6] The Open Group, *"Service-Oriented Architecture Ontology"*, Technical Standard, 2010, https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=c104, last access: March 20th, 2012.

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

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

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

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

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

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

[13] 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.

[14] 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.

[15] The Open Group, *"SOA Reference Architecture"*, Technical Standard, 2011, https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=c119, last access: March 20th, 2012.

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

[17] S. Bourscheidt, T. Breuer, T. Brunner, B. Fetler, G. Fogel, "*ESARC-Referenzmodell und Ontologie für Enterprise Architecture Management (EAM)*", Research Report, Reutlingen University, Enterprise Services Architecture Reference Lab, 2012.

[18] OMG, "*Meta Object Facility (MOF) Core Specification*", Version 2.0, Object Management Group, 2006.

[19] N. F. Noy, D. L. McGuineness, "*Ontology Development 101: A Guide to Creating Your First Ontology*", Stanford University, 2001

[20] D. Gasevic, D. Djuric, V. Devedzic, "Model Driven Engineering and Ontology Development", 2nd Edition, Springer Verlag, 2009.

[21] M. Horridge, and H. Knoblauch, A. Rector, R. Stevens, C. Wroe, S. Jupp, G, Moulton, N. Drummond, *"A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools"*, Edition 1.2, The University of Manchester, 2009.

[22] L. Youseff, M. Butrico, D. Da Silva, "*Towords a Unified Ontology of Cloud Computing*", in Grid Computing Environments Workshop (GCE# 08), Nov. 2008, pp. 1-10, 2008.

[23] H. Zhou, H, Yang, A. Hugill, "*An Ontology-Based Approach to Reengineering Enterprise Software for Cloud Computing*", in 34th Annual IEEE Computer Software and Application Conference, 2010, pp. 383-388, 2010.

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

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

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

[27] Sonic "*A new Service-oriented Architecture (SOA) Maturity Model*" http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf, last access: March 20th, 2012.

[28] Oracle "*SOA Maturity Model*", http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet, last access: March 20th, 2012.