

A Conceptual Framework for the Governance of Service-Oriented Architectures

Jan Bernhardt¹ and Detlef Seese²

¹ SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany

² Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
jan.bernhardt@sap.com, seese@aifb.uni-karlsruhe.de

Abstract. With the widespread adoption of service-oriented architecture (SOA) as the leading paradigm for the development of business applications, the need for adequate governance mechanisms to keep control of the increasing complexity inherently introduced by SOA arises. However, current approaches to SOA governance are often limited to either design time or runtime aspects of SOA governance, despite the need for adequate governance mechanisms covering the complete SOA lifecycle. Furthermore, no common understanding of SOA governance exists today, as many solution vendors misleadingly label SOA management products as solutions for SOA governance.

This work presents a reference model for SOA governance that is based on the OASIS Reference Model for Service Oriented Architecture. It aims at establishing a common understanding of SOA governance and is intended to serve as a basis for comparing and developing concrete SOA governance solutions.

1 Introduction

To date neither a common understanding of, nor a common model for SOA governance exists. Moreover, current approaches proposed by SOA governance solution vendors are incomplete in that they only address some aspects of SOA governance, mainly technical governance and management aspects. Our reference model for SOA governance strives at closing these gaps. It is based on a reference model for SOA methodologies (see [1]), which in turn is based on the OASIS Reference Model for Service Oriented Architecture (SOA-RM) [2].

In order to account for the special nature of SOA – increased complexity (e.g. [3]), augmented security requirements (cf. [1]), and demand for better business and IT alignment (e.g. [4]) – the discipline of SOA governance has evolved. It is an extension or evolution of IT governance defining the organizational structures and processes that are required to successfully adopt SOA (e.g. [4], [5]).

Drawing on [6] and on definitions of corporate [7] and IT governance [8] we define SOA governance as follows:

Definition 1 (SOA governance). *SOA governance consists of the organizational structures, processes, and policies an organization puts in place to ensure*

that the adoption, implementation, and operation of SOA in an organization is done in accordance with best practices, architectural principles, government regulations, and laws, that is, in a manner that sustains and extends the organization's strategies and objectives.

The SOA-RM defines SOA as “a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains” providing “a uniform means to offer, discover, interact with and use capabilities to produce desired effects” [2]. Based on this understanding of SOA and taking into account its special nature we have developed a reference model for SOA governance consisting of a comprehensive set of governance policies, processes, and corresponding organizational constructs.

The remainder of this work is organized as follows. Sect. 2 summarizes related work in the area and identifies gaps in current approaches, motivating the need for a reference model for SOA governance. The reference model is presented in Sect. 3. We conclude our work in Sect. 4 by giving an outlook on areas for future research.

2 Related Work

Over the last few years SOA governance has been a topic primarily driven by analysts (e.g. [6], [9], [10]) and technical governance and management solution vendors (e.g. [11]), and only recently scientific literature has been published in the area (e.g. [5]). In addition, some of the recent books on SOA also contain chapters about SOA governance (e.g. [4], [12]). Finally, numerous online articles are available covering the topic in more or less detail (e.g. [13], [14], [15]).

Some main characteristics of SOA governance recur throughout the literature. First if all, the need for SOA governance is motivated from the increased complexity of SOA originating from the number of services and their interdependencies that exist in a SOA landscape (e.g. [4], [5], [12]). Secondly, consensus seems to exist about the fact that SOA governance can be seen as a specialization (or extension) of IT governance accounting for the special nature of SOA (e.g. [4], [12], [14], [15]). Thirdly, [4], [5], and [6] name policies, processes, organizational structures, and metrics as core constituents of a SOA governance program. In addition it is stated that in order to successfully fulfill its mandate SOA governance needs control over project approval and funding as well as portfolio management decisions (e.g. [4], [13]). Finally, reuse of services is commonly named as a main benefit of SOA (e.g. [9], [13]). However, [5] legitimately argues that reuse contradicts agility.

Based on the SOA-RM [2] OASIS has published a Reference Architecture for Service Oriented Architecture [16], which contains a generic model for governance augmented with some specifics for SOA governance. Furthermore, The Open Group has set up a project on SOA governance [17] as part of its SOA working group, whose aims include the definition of a reference model for SOA governance. However, no results have been published to date.

What distinguishes the reference model presented in this work from available material in the domain is the rigorously holistic approach assumed, covering the complete SOA lifecycle from service proposition to retirement. Moreover, all aspects that should be considered by SOA governance have been identified from a reference model for SOA methodologies (see [1]) and are presented in a comprehensive, uniform set of governance policies, processes, and organizational constructs, which to the best of our knowledge is unprecedented in the literature.

3 A Reference Model for SOA Governance

In order to successfully adopt SOA in an organization, certain organizational structures, consisting of a number of organizational units, have to be established to govern the introduction and operation of SOA. The organizational units define governance policies and processes, execute governance processes, and employ metrics to measure adherence to them (Fig. 1).

Governance processes enable governance policies and enforce compliance. Metrics provide visibility into the processes allowing for supervision, control, and identification of areas for improvement. Furthermore, a SOA governance infrastructure helps to manage governance information, enforce governance policies, and collect corresponding metrics (cf. [6]). The following subsections describe the proposed constituents of SOA governance in detail.

3.1 Organizational Structure

The organizational structure for SOA governance comprises the SOA governance board, the SOA program office, and a number of delegate councils.

SOA Governance Board. The SOA governance board constitutes the main governance body of a SOA initiative. It should be composed of specialists from both the business and technology domains such as business analysts, enterprise architects, and service architects. In addition, it is important that decision makers such as the line of business managers that are affected by the SOA initiative

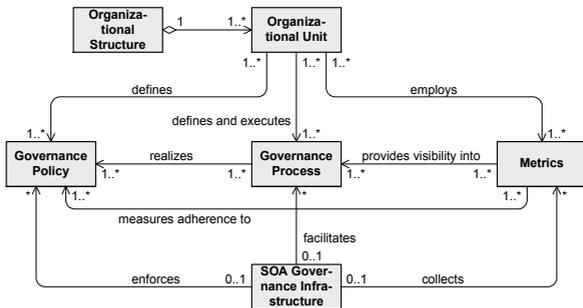


Fig. 1. Elements of SOA governance

are also part of the governance board to establish the required influence and authority.

In particular, the SOA governance board takes into account business objectives, architecture and design standards and best practices, and applicable laws to translate them into corresponding governance policies. It also defines metrics that will be used to measure adherence to and success of the policies. Furthermore, governance processes have to be defined and set up that integrate with the SOA lifecycle to realize the policies.

The governance board is supported by a number of councils to which it may delegate work (cf. [16], see [1] for details). Depending on whether an organization applies a central or distributed SOA governance scheme (cf. [4]), the board either possesses exclusive decision rights regarding policies and processes, or it may empower the delegate councils to make decisions themselves.

As a transition to SOA is a highly complex undertaking with impact on many different parts of an organization, the transition should take place iteratively (cf. [4]). It is the responsibility of the SOA governance board to set up an overall SOA roadmap that accounts for progressive adoption of SOA.

Finally, project prioritization and funding are necessary means to steer the SOA initiative into the desired direction, particularly with respect to business alignment of services. Without practical control of funding projects are more likely to produce project-centric services which might not meet the overall requirements of the organization [13]. Therefore the SOA governance board must at least possess authority to influence decisions on project prioritization and funding.

SOA Program Office. The SOA program office is the main organizational management body in a SOA initiative, which is in charge of the overall SOA program management. Its responsibility is to set up the required organizational structures, first and foremost the delegate councils.

The program office reports to the SOA governance board. It is authorized to decide on project prioritization, selection and funding in accordance with the board, which may take influence in the decision process. The program office also assigns people to the roles defined by the governance board. Finally, it may also influence the SOA roadmap.

3.2 Policies

Policies are the means to define what is *right*. This reference model distinguishes two notions of policies. Governance policies, on the one hand, formulate general rules that address important aspects of SOA such as how to develop, introduce and operate a SOA. Governance policies state on a high level what to do and how to do it. Service policies, on the other hand, are more concrete; they directly affect, and can be attached to one or more services.

Definition 2 (Governance policy). *Governance policies comprise rules that on a high level address all aspects that are important in developing, deploying*

and managing a SOA, that is, the SOA lifecycle. Governance policies define what to do and how to do it.

Definition 3 (Service policy). *Service policies define rules that specifically apply to one or more services. They are always derived from governance policies.*

Service and governance policies are both specializations of policies. A policy consists of one or more policy assertions, the fulfillment of which is measurable [2]. When an individual or an organization adopts one or more policy assertions as their policy, the policy assertions become the policy of that individual or organization. In the scope of this reference model we do not explicitly distinguish between policies and policy assertions, but reluctantly speak of policies instead.

Every policy has a dedicated policy owner. The policy owner is the one who has an interest in the policy. He usually is responsible for the enforcement of the policy [2].

In our work we have identified 41 different governance policies for the complete SOA lifecycle (i.e. service proposition, design, implementation, provisioning, consumption, and management). These policies are motivated from the definition of SOA given in [2]. They ensure that all activities associated with the SOA lifecycle are executed in a manner that sustains the organization's objectives. In the following paragraphs we present the most important of the identified governance policies; for a full description see [1].

Service Proposition. For service proposition the approach that should be taken has to be specified (service proposition policy). Services can either be proposed in a top-down, bottom-up, or in a combined manner (cf. [18], [19]). Top-down service proposition identifies service candidates from some kind of business model, whereas bottom-up service proposition uses existing application system functionality to propose services. It is also possible to apply a combination of the two alternatives.

Furthermore, fundamental choices regarding general service-oriented design principles, which also have an effect on service proposition, have to be made. Two aspects that need to be considered in this regard are service granularity and reusability (e.g. [20]; service granularity and reusability policies). Designing services for reuse is more expensive [5], wherefore the benefit of avoiding duplicate functionality has to be weighed against the additional costs. Moreover, service reuse decreases agility [3], as it increases the number of dependencies that have to be considered in case services need to change. Service granularity and reusability also affect each other, since services with a narrower focus may be more easily reused [20].

Service Design. Regarding service design it has to be determined whether different service types should be distinguished (e.g. infrastructure, data, functional, and process services; cf. [21]). This is stipulated in a corresponding service type policy.

Another fundamental aspect of service design is the use of service design patterns. A number of design patterns for services are emerging (e.g. [22]), and

it has to be determined if any of these patterns should be applied for service design (design pattern policy).

Furthermore, the service error handling scheme has to be determined (error handling policy). Options include using a standard (e.g. SOAP) or a proprietary error handling mechanism, and to handle all types of errors (e.g. application errors, communication errors) in the same way or differently.

Finally, the transaction behavior of services also has to be taken into account (transaction behavior policy). The question to answer here is whether service invocations should adhere to the ACID (atomicity, consistency, isolation, and durability) characteristics common in database design.

For service interface modeling it has to be defined whether service interfaces should be designed according to common message schema standards (e.g. ebXML, RosettaNet). In case many services will be disclosed to external parties, it may be advantageous to comply with common standards. On the other hand, if the number of public services is low, the development of proprietary schemas following the organization's data architecture may be more applicable (message schema policy).

An important aspect of a SOA initiative is the unification of data types across the organization (cf. [23]) to achieve a common understanding of data types, to allow for reuse, to create compatible service interfaces in terms of syntax and semantics, and to keep message transformations during service operations to a minimum (data type unification policy). In addition, data type definition and reuse policies stipulate naming rules for data types, application of common standards for data type definition (e.g. Core Components Technical Specification) to facilitate semantic service interoperability, and rules regulating reuse of data types.

Another fundamental decision refers to transaction patterns (transaction pattern policy). The definition and prescription of transaction patterns such as query/response or request/confirmation for all service interfaces provides a means to unify the semantics of the interfaces, which enhances service interoperability (see e.g. [20]).

Consistent naming of service interfaces, i.e. the definition of naming conventions, is another important aspect (naming policy). In case transaction patterns are employed, it may be beneficial to include the employed pattern into the interface and operation names to clearly indicate their semantics. For example, the interface of a customer management service that adheres to the request/confirmation pattern may have *customerCreateRequest* and *customerCreateResponse* operations.

Finally, it has to be discussed whether reliable communication is required and how it should be achieved (reliable communication policy).

Service Implementation and Quality Assurance. Regarding service implementation governance has to decide whether services should be implemented on the basis of a component technology (e.g. Service Component Architecture (SCA) [24]). Furthermore, the implementation and interface technologies together with the technical frameworks that should be used have to be determined

(service component policy and implementation technology policy, respectively). As it is one of the aims of SOA to allow services built using a variety of platforms and programming languages to interact, it is possible that multiple languages are chosen for implementation depending on the specific requirements.

Governance concerning service quality assurance is a crucial aspect for the success of SOA and ranges from determining tests that have to be conducted (testing policy) to reviews that have to be passed (review policy). Common tests for services include dependency, interoperability, and regression tests for functional service testing, and performance, stress, and security tests for non-functional service testing. In addition, scenario tests to test a number of dependent services in combination are particularly important (cf. [6]).

Service Documentation. The service description is a fundamental part of SOA as it provides information necessary to interact with a service to potential service consumers. A service documentation policy therefore has to ensure that all relevant information is actually included in the service description. It is probably useful to define a service description template in this respect which has to be adhered to. Moreover, with regard to service discovery, it is advisable to define a service taxonomy and to classify all services according to this taxonomy to alleviate discovery of services that can satisfy a consumer's needs.

According to [2] the service description should at least contain information about the functionality of the service, access to the service (i.e. the service address), and policies the service operates under. Concerning the service interface the required message syntax, the underlying semantics, and allowed message sequences, including applicable transaction patterns, have to be specified. In addition, main characteristics of the service such as statelessness, idempotency, and warranties regarding reliable communication should be part of the service description as well.

Finally, service policies such as guaranteed service levels and according remuneration schemes, as well as the service's error handling mechanism should also be described.

Service Publication and Consumption. Regarding service publication it has to be decided where services (i.e. service descriptions) should be published, i.e. if a service registry should be employed, or whether services should simply be published in a file, for example (service publication policy). In case it is decided to use a registry, options include using separate registries for service publication and service governance (e.g. for maintaining administrative information and service lifecycle states), or a single registry for both.

Furthermore, a role concept and according access control mechanisms for the registry have to be determined, including decisions regarding who is allowed to publish and who entitled to release services for production.

Finally, in case a service taxonomy has been defined for services, the attributes assigned to each service should be published in the registry as part of the service description.

The role concept and access control mechanisms for the registry also have to take into account rules for service consumption. The corresponding service consumption policy has to specify who is allowed to search for services, access services (e.g. for testing, for production), and to generate service proxies from the service interfaces in order to consume services.

Another important aspect of service consumption is contract negotiation. Therefore the service consumption policy also has to determine who is responsible for and entitled to negotiating contracts with service consumers. This could, for example, be the service owner.

Finally, the contract language is another important aspect to consider. Depending on the subject of the contract it may be specified that contracts have to be expressed in a machine-processable form to permit automated interpretation and enforcement (cf. [2]).

Service Management. Services in production have to be monitored to ensure that non-functional service requirements specified during service design, and service levels negotiated during service consumption are maintained. Service monitoring inherently is a management activity. It is the responsibility of governance, however, to specify what has to be monitored and to define remedial actions in case violations occur (service monitoring policy). Service monitoring likely includes service level, availability, security, and error monitoring.

Besides, distinct policies define how individual management responsibilities should be handled. A service level management policy defines how contracted service levels should be maintained, remedial actions in case service levels are violated, and different service levels and according pricing models offered.

A service availability policy specifies how service availability management should be performed in order to guarantee the degree of service availability assured to service consumers. This again includes the definition of remedial actions in case availability is threatened.

A security management policy determines how security management should be performed. This may reach from high level governance rules stating that sensitive services have to be secured in some way, to concrete rules prescribing the actual access control and encryption mechanisms to be applied.

Finally, a dependency management policy specifies how service dependencies should be managed. One of the main challenges in SOA is the increased complexity that has to be handled and which stems from the number of interdependent services existing in a SOA landscape together with their organizational structure and the dynamics of their interaction (cf. [5]). If these dependencies are not structured well and managed without gap, service changes may lead to critical errors.

Service Versioning, Change, Deprecation, and Retirement. Due to changing business and other requirements, services are likely to change during their lifetime. Governance has to decide how changes are handled, and whether incompatible or only compatible changes are allowed (change policy).

Closely related to change is the topic of versioning (versioning policy). For example, it could be specified that compatible changes entail a new service version,

whereas incompatible changes require a new service with a different name and different taxonomy attributes. Another aspect pertains to how many versions of a service are allowed to exist in parallel. Increasing maintenance costs probably necessitate the retirement of older versions at some point in time.

Furthermore, service deprecation and retirement procedures, as well as transition periods have to be appointed (service deprecation and retirement policies). Transition periods may be subject to negotiation and codified in the service contract.

Finally, service providers may want to notify service consumers of events such as service change, deprecation, and retirement (notification policy). The obligation to send notifications may also be included in the service contract.

3.3 Governance Processes

Governance processes are the means to realize governance policies. This includes policy-related processes such as definition and enforcement, as well as project-related processes such as prioritization and funding. Furthermore, review processes as an instrument to verify policy compliance are another important aspect of SOA governance (for details see [1]). To minimize the overhead inherently introduced by any governance mechanism, governance processes should be automated wherever feasible.

Policy-Related Processes. A number of governance processes revolve around the definition, propagation, and enforcement of governance policies. Having established an appropriate organizational structure for SOA governance, the respective organizational bodies (i.e. the SOA governance board, the delegate councils) start defining and documenting governance policies for their specific areas of expertise. The policies may have to pass a resolution process, particularly if policies developed by delegate councils have to be ratified by the governance board. Besides, it is necessary to educate on the policies (policy enablement).

Furthermore, mechanisms enabling policy enforcement have to be established. Enforcement can either be achieved manually (e.g. by a design review), or automatically (e.g. by a SOA management system). Closely related to enforcement is the handling of policy violations, which likely comprises initiating compensatory actions (cf. [2]). In addition, policy waiver procedures should be defined (cf. [6]) to be able to bypass certain policies in case special business requirements require to do so.

Finally, policies may have to be adapted due to changing business requirements, or because they prove inadequate in practice. The concept of versioning is therefore applicable not only to services, but to policies as well.

Approval, Review and Reporting. Review processes are a means to enforce governance policies, particularly regarding design time policies. Passing a review may be a prerequisite for the transition to the next phase of the SOA lifecycle (cf. [13]). For example, before moving on to service implementation, service models may have to pass a service design review.

Closely related to reviews are approval processes, which are another means for policy enforcement. In particular, new services or service changes may have to be approved by the SOA governance board before they may be realized. Besides services, other design decisions such as the introduction of new data types may be subject to approval as well.

Finally, reporting processes provide feedback from the organizational units concerned with realizing the SOA to the SOA governance board, which it requires to maintain overview of the SOA initiative and to intervene if necessary.

3.4 Metrics

Metrics improve transparency and provide visibility of essential states and parameters of the SOA initiative on the whole, and into governance processes in particular to the SOA governance board (compare corresponding developments on lower levels such as software or process metrics, see e.g. [25]). They may be partitioned into metrics related to services, service operations and projects.

Regarding services it is interesting to know how many services exist in total, have been proposed, are being developed, are published, and are actually consumed. Furthermore, numbers on changed, deprecated and retired services also provide useful insights. Finally, information on the number of consumers per service and service version is helpful as well, particularly regarding the impact of changes.

To provide visibility into service operations and to identify areas for improvement a number of operations-related metrics are helpful. These include the number of successful vs. the number of erroneous interactions, including information indicating the reasons for the errors (e.g. SOAP faults, invalid input data, size of input data etc.). Another important aspect is performance KPI's, as these help to identify both bottlenecks in the infrastructure and the need for service design revisions. Furthermore, service level and security violations are also important to report.

Finally, in order to maintain the overview of the SOA initiative on the whole it is important to provide feedback on the number of successful and failed projects and the reasons for both failure and success.

3.5 SOA Governance Infrastructure

SOA governance processes must be as unobtrusive as possible to encourage adoption of SOA. A SOA governance infrastructure (SGI) helps facilitating SOA governance by managing governance information, automating governance processes, enforcing governance policies, and collecting metrics (cf. [6]).

The elements of an SGI are determined by the specific governance requirements. A service registry helps to maintain overview of existing services and to manage administrative information such as service lifecycle states. It may also allow for service consumer notifications, e.g. using some kind of publish/subscribe mechanism for change and deprecation events. For the management of design artifacts a repository can be employed. It is also a natural candidate for a central data type inventory.

Policy and contract management are supported by policy and contract management systems. A policy management system helps to define and document policies, attach policies to artifacts, and to propagate policies to their respective points of application. It may also provide capabilities for analyzing the impact of policy changes.

With respect to the elevated testing requirements in SOA a specialized quality management system may support testing complete service compositions, as well as impact analysis in case of service changes, taking into account dependencies between services.

Finally, a SOA management system should facilitate service administration (e.g. lifecycle transitions) and monitoring. Moreover, automatic policy enforcement mechanisms, as well as reporting capabilities should also be provided.

4 Conclusion

This work has outlined a comprehensive reference model for SOA governance that is based on the standardized SOA-RM [2] and motivated from aspects relevant to methodologies for SOA (see [1]). Unlike previous approaches to SOA governance, our reference model addresses governance aspects for the complete SOA lifecycle, stipulated in a comprehensive set of governance policies, processes, and organizational considerations.

Having established a common understanding of the concepts relevant for SOA governance, in a next step we plan to investigate in detail the relationships between our reference model and common frameworks for IT governance (e.g. COBIT) and Enterprise Architecture (e.g. TOGAF, Business Engineering Model [26]).

References

- [1] Bernhardt, J.: A Conceptual Framework for the Governance of Service-Oriented Architectures. Master's thesis, Universität Karlsruhe (TH), Germany (April 2008)
- [2] MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R.: OASIS Reference Model for Service Oriented Architecture 1.0 (October 2006)
- [3] Schelp, J., Winter, R.: Towards a Methodology for Service Construction. In: HICSS 2007: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Washington, DC, USA, p. 64. IEEE Computer Society, Los Alamitos (2007)
- [4] Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R.: Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap, 2nd edn. IBM Press Pearson Education, Upper Saddle River (2006)
- [5] Schelp, J., Stutz, M.: Serviceorientierte Architekturen (SOA). In: SOA-Governance, 1st edn. ser. HMD – Praxis der Wirtschaftsinformatik, pp. 66–73. dpunkt-Verlag, Heidelberg (2007)
- [6] Manes, A.T.: You Can't Buy Governance, InfoWorld SOA Executive Forum (November 2007), <http://www.infoworld.com/>
- [7] Organisation for Economic Co-Operation and Development, "OECD Principles of Corporate Governance (2004), <http://www.oecd.org/>

- [8] IT Governance Institute (ITGI), Board Briefing on IT Governance, 2nd edn. (2003), <http://www.itgi.org/>
- [9] Malinverno, P.: Service-Oriented Architecture Craves Governance (January 2006), <http://www.gartner.com/>
- [10] Malinverno, P.: Sample Governance Mechanisms for a Service-Oriented Architecture (April 2006), <http://www.gartner.com/>
- [11] Systinet, SOA Governance: Balancing Flexibility and Control Within an SOA (September 2006), <http://www.hp.com/>
- [12] Starke, G., Tilkov, S. (eds.): SOA-Expertenwissen: Methoden, Konzepte und Praxis serviceorientierter Architekturen, 1st edn. dpunkt-Verlag, Heidelberg (2007)
- [13] Carlson, B., Marks, E.: SOA Governance Best Practices – Architectural, Organizational, and SDLC Implications (January 2006), <http://webservices.sys-con.com/>
- [14] Brown, W.A., Cantor, M.: Soa governance: how to oversee successful implementation through proven best practices and methods (August 2006), <http://www.ibm.com/>
- [15] Starke, G.: SOA Governance: Crucial Necessity or Waste of Time?(November 2007), <http://www.infoq.com/>
- [16] McCabe, F.G., Estefan, J.A., Laskey, K., Thornton, D.: OASIS Service Oriented Architecture Reference Architecture 1.0 Public Review Draft 1 (April 2008)
- [17] The Open Group, SOA Working Group SOA Governance Project (August 2006), <http://www.opengroup.org/projects/soa-governance/>
- [18] Arsanjani, A.: Service-oriented modeling and architecture – How to identify, specify, and realize services for your SOA (November 2004), <http://www.ibm.com/developerworks/library/ws-soa-design1/>
- [19] Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G.: Impact of service orientation at the business level. IBM Systems Journal 44(4), 653–668 (2005)
- [20] Erl, T.: SOA Principles of Service Design, 1st edn. Prentice Hall, Upper Saddle River (2008)
- [21] Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall, Upper Saddle River (2006)
- [22] Erl, T.: SOA Design Patterns. Prentice-Hall, Upper Saddle River (scheduled for 2008)
- [23] Manes, A.T., Tilkov, S.: Anne Thomas Manes on SOA, Governance and REST (July 2007), <http://www.infoq.com/>
- [24] Open SOA Collaboration, Service Component Architecture (SCA) v1.0 (March 2007), <http://www.osoa.org/>
- [25] Melcher, J., Seese, D.: Process Measurement: Insights from Software Measurement on Measuring Process Complexity, Quality and Performance. University of Karlsruhe (TH), Institute AIFB, Tech. Rep. (2008)
- [26] Österle, H., Winter, R. (eds.): Business Engineering, 2nd edn. ser. Business Engineering. Springer, Berlin (2003)