

A lifecycle approach to SOA governance

T.G.J. Schepers
Deloitte
TSchepers@deloitte.nl

M.E. Iacob
University of Twente
M.E.iacob@utwente.nl

P.A.T. Van Eck
University of Twente
P.A.T.VanEck@utwente.nl

ABSTRACT

Due to the distributed nature of Service-Oriented Architectures (SOA), maintaining control in a SOA environment becomes more difficult as services spread over different lines-of-business. The concept of SOA governance has emerged as a way to implement control mechanisms in a SOA. In this paper we identify a lifecycle based approach for executing SOA governance. This approach consists of defining a SOA strategy, aligning the organization, managing the service portfolio, controlling the service lifecycle, enforcing policies and managing service levels. By incorporating a maturity model in this approach, it is possible to minimize the required effort while still having sufficient governance. From a series of interviews that have been carried out we could conclude that most current SOA projects - although relatively limited in their scope - raise governance issues that need to be addressed to prevent future problems.

Categories and Subject Descriptors

D.2.9 [Software Engineering]: Management – *lifecycle, productivity, software configuration management, software process.*

General Terms

Management

Keywords

Service-oriented architecture, governance, lifecycle, policy, service level, software quality, maturity,

1. INTRODUCTION

The field of Service-Oriented Architecture (SOA) has received much attention in the past years. SOA is a software architecture that is designed around loosely coupled software components called services, which can be orchestrated to improve business agility [9]. As defined by W3C, services provide functionality at the application and business levels of granularity using widely applied standards [1]. Mitra [14] is one of the first to notice governance issues in relation with SOA. He claimed that by embracing SOA, governance needs to be taken more seriously into account because of the distributed nature of services across lines-of-business.

The goal of this paper is to provide an approach for solving governance problems in an SOA. Our focus is on the role of IT

and organizational management in maintaining control over a SOA. We present a holistic, high-level approach in the form of a SOA governance lifecycle and we set out the scope of SOA governance.

SOA promises the ability to improve integration of information systems and improve the alignment of business and IT through loose coupling and design around services. As such SOA is in fact a new architecture style in the broader enterprise architecture field. Therefore our approach for SOA governance draws upon general research on enterprise architecture governance (e.g., [22], [27], [3]). However, the focus in our lifecycle approach will be on the specific problem areas of service orientation such as increased complexity due to distribution of business logic and the increased number of artefacts in the IT landscape, service portfolio and service levels management, etc.

This paper is organized as follows. First, the problems that SOA governance should solve are analyzed (Section 2). Through this problem analysis, the scope of SOA governance is defined. Next, the approach we propose for the identified problem areas will be described (Section 3). Although some of these areas may overlap, they form distinguishable areas of expertise and are discussed separately. Furthermore, an analysis of the SOA maturity levels is also included (Section 4). The purpose of this part is to make SOA governance scalable to different requirements. In Section 5 we discuss the practical implications for using an SOA governance lifecycle and we indicate how this approach affects the current practice. Finally, we draw a number of conclusions in Section 7.

2. PROBLEM STATEMENT

The advantages SOA offers by its distributed nature and loose coupling also lead to its main challenges. It can be said that software complexity is pushed from the software development domain into the area of choosing the right services and orchestrating and composing them. The idea of creating small understandable blocks of logic requires a lot of coordination from software developers in order to maintain the overview in a jungle of services.

Below we summarize several commonly occurring problems in SOA projects that have been previously documented in the literature [1], [9], [16], [18] or have been reported during projects we have surveyed. We have classified governance related issues into the following general problem categories:

- *Compliance to standards or legislation, requiring audit trails of IT systems.* The loose coupling of SOA makes it hard to audit the behavior of all the services that are called by a user request.
- *Creating a budget* is complicated since benefits and costs of services are attributed to multiple organizational units.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

Copyright 2008 ACM 978-1-59593-753-7/08/0003...\$5.00.

- *Consequences of changing a service* are hard to predict since a service has many consumers of which the service developer has no knowledge.
- *Ensuring quality of service is hard.* Design-time quality is needed to let a service comply with standards and policies. Run-time quality relates to all quality aspects needed/measured after publication of a service, such as service performance.
- One of the features of a SOA is that it allows *business activities to be encapsulated in services*. This requires a change in attitude from people. They need to see their job as a value-adding service offered to consumers, instead of focusing on their own area of expertise.

Many of the above-mentioned issues can be seen in relation with common IT control problems and can be attributed to the concept of IT governance, whose main purpose is to ensure that “the organization’s IT sustains and extends the organization’s strategies and objective” [12]. Through frameworks such as COBIT [11], IT governance has been implemented by rigid procedures to manage resources and record decisions. IT governance is usually performed by IT management and supported by other IT staff. Nevertheless, in our opinion, SOA governance is essentially different from IT governance in that it also requires the involvement of business management and of line-of-business employees. The intention of SOA is that services represent business activities, so IT not only sustains but also becomes a translation of what the business performs.

The problem we therefore want to solve is how to address governance in the context of SOA. To our knowledge, there is no dedicated method available which addresses the business-specific demands of this architecture. Therefore, in the following section, we propose such a method that improves the control that can be executed on a SOA by a series of activities we call the SOA governance lifecycle.

3. THE SOA GOVERNANCE LIFECYCLE

The problems discussed in the previous paragraph are based on different areas of expertise, which are also differently placed in the lifecycle of a SOA. The most noticeable topics in SOA governance have been placed in a lifecycle to ensure a consistent path from strategic considerations to delivery of the architecture. The relation between these phases and the problems stated will be discussed in the end of this section. The SOA governance lifecycle model we are proposing is shown Figure 1.

Creating a SOA strategy is the task which triggers the whole process. The lifecycle shows the order in which the phases should be initiated. However, this order does not imply that a phase must end before the next one can begin. For example, often the strategy needs some updates during the rest of the lifecycle. Therefore, the arrows in our model are an indication of some causality relationship between the phases and should not be seen as a chronological ordering. In the remainder of this section we will address the role and corresponding approach of each of the phases in the SOA governance lifecycle.



Figure 1: phases in the SOA governance lifecycle

3.1 Define a SOA strategy

First, the strategic direction of SOA needs to be defined in order to align the SOA with business requirements. A successful SOA requires an ambitious long term, but also realistic, strategy. During this phase, the business vision is translated into strategic SOA goals. These long-term goals are also the basis for the definition of the business case and the scope of the SOA program in the current iteration of the lifecycle. Creating a business case for the SOA program can be problematic, as the investment needed for the first projects is often greater than the initial benefits. The possible benefits from follow-up projects also need to be considered in order to create a valid business case.

Another important issue in this early stage is to gain the support of all the relevant stakeholders. Having different stakeholders committed from the beginning decreases their resistance to change and reduces the risk of program failure in a later stage. Nevertheless, having too many stakeholders involved can also have negative side-effects, such as a difficult and slow decision making process. This drawback can be addressed by assigning specific roles to stakeholders according to the RACI method [7].

3.2 Align organization to SOA

The next phase concerns the alignment of the strategic planning to the organizational context. Creating clear SOA governance bodies by assigning responsibilities and establishing project groups will lead to a more structured approach in the next lifecycle phases.

An example of such a body is the so-called Centre of Excellence (CoE), which is a team that collects expertise and drives programs in the organization. A CoE can be of good use for SOA, as it can become a helpdesk for both service developers and service users [24]. The CoE defines best practices and helps to train people and communicate the purpose of SOA. Since they receive feedback from different types of users, the CoE is able to notice problems directly, and is able to advice on corrective actions. It is possible to give this body authority for making decisions, but often its role is limited to finding problems and making recommendations.

The success of organizational alignment to SOA depends on the current practice and culture in the organization. Some organizations will prefer one centralized unit in control, while others let individual business units in control of their own parts, as long as they comply with the standards and policies that have been set at corporate level. This decentralized approach can improve flexibility, but also requires more coordination effort [16].

In order to efficiently control the development of services, their ownership must be assumed by business management. Prerequisite hereof is the definition of structuring criteria for the service portfolio. Often a set of services relates to the same organizational domain and performs similar functions. By choosing to group these services into service domains, their management is greatly simplified since there are fewer artifacts to govern. We distinguish four types of structuring criteria that can be used to group and govern services. The first is a *process domain*, where end-to-end processes are typically seen as distinct service domains. The second type is the *product domain*, built around similar products or services delivered by the organization. The third option is to use *geographical domains*, useful when each location of the organization has a high degree of independence. Finally, *functional domains* can be used for services that belong to a certain organizational function (e.g., a specific department).

3.3 Manage service portfolio

Once the governance structure is in place, it is time to think about the services that need to be developed. Portfolio management ensures that a sound method is used consistently to decide which services need to be developed and how the necessary investments are prioritized.

Services to be developed can be identified in a bottom-up fashion, meaning that the identification process is triggered by service requests from users, after which an assessment is done to check whether these requirements can be fulfilled. Alternatively, in a top-down approach, the process starts with the overall goals of the project. Through an abstraction process, the goals lead to changes in processes, and these processes can be translated into process steps, from which service candidates can be derived [24]. A top-down approach requires more coordination effort and may cause resistance to change, but it leads to a more coherent solution. [9]

The need to incorporate legacy systems in SOA may become another source of services. One of the most important advantages of the SOA paradigm is the fact that it allows for relatively easy incorporation of legacy applications. Older software can be encapsulated in services and subsequently integrated in a SOA by means of newly created service interfaces. An approach for developing a migration strategy of older software architectures to SOA has been described in the Service Migration And Reuse Technique (SMART) method by Lewis [14]. This method starts with gathering information about the legacy system, its context and the purpose it should serve in the SOA. Then the gap between the current situation and the desired one is evaluated to form a migration strategy.

3.4 Control service lifecycle

The control of a service's lifecycle concerns the development and delivery of individual services in a SOA. This relates to principles used during the design, development and delivery of services. This includes deciding on service granularity, change management procedures for services and on registration of available services in the SOA. Service lifecycle control aims at consistency between services, which is expected to improve their manageability.

Service lifecycle control raises some new specific issues that are less of a problem in traditional application lifecycles. The

distributed nature of the service development process may lead to "rogue services" [24] (i.e., unregistered services that cannot be governed) entering the SOA. This may have as result the development of similar services because of insufficient knowledge of the functions provided by existing services or services built by other developers.

The distance between service provider and service consumer also increase the complexity of the change management process in SOA. Thus all the consumers of a service need to be informed when the implementation details of that service change. Before changing a service, it is important to make an impact analysis, in which the consequences and possible risks of the change are estimated. Changes in the responses of a service may require changes in other services that invoke the changed service. It is therefore common to allow consumers some time before switching, by temporarily offering multiple versions of a service at the same time.

An important role in the control of the service lifecycle is played by service registries. These are essentially catalogue tools that manage the publication of services and define taxonomies of the published services. Service consumers may use the registry to find suitable services. While registry tools only store references, repository tools actually hold data about services. Service repository tools are also able to provide auditing functions to check the changes made to a service [1]. The service itself and its code is usually located on some application server, but documentation, policies and metadata about versioning of the service is contained in the repository. The functions of registries and repositories in the service lifecycle are shown Figure 2.

Service registries can be used by governance bodies to impose validations or other checks to be performed before a service is published. Registries can also support the rights management of users by checking if they are authorized to use a certain service.

3.5 Incorporate policy enforcement

Policies are formalized business rules that pose constraints on the way services are developed and used. Policy enforcement has become a major issue in SOA to ensure desirable behavior and consistency of services. A popular distinction is made in the literature between design-time and run-time policies. Design-time policies are directed to the developer, and concern, for example, the use of security mechanisms or compliance to data standards. Run-time policies involve the operational environment and often concern requirements that have to be met by the service at run-time (e.g., performance requirements) [24].

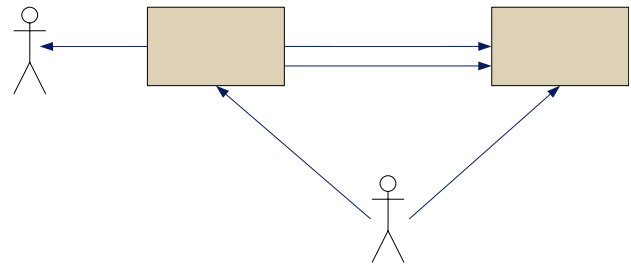


Figure 2: role of registries and repositories in service lifecycle management

Policy enforcement can be done in different ways. One option is to perform a check before the service is published. For some policies however, it is also possible to automatically verify if a service complies with them. Automated policy enforcement can be performed at three locations [24]: the service itself, at a gatekeeper that is built around a group of services or through the use of a message transport layer.

Figure 3 shows how tools can support the enforcement process. Repositories store all versions of policies, the active policy can be then submitted to a registry tool (often the registry and the repository are integrated). Infrastructure tooling (service buses or other message brokers) controls the traffic and monitors the service usage and exceptions. This information is passed on to management tools which can generate reports to assist policy owners in improving them.

3.6 Service level management

Guaranteeing a sufficient Quality of Service (QoS) becomes a major issue in SOA. Since one request may assume the usage of a several services, each of these services has to perform at the desired level of quality. And even then, the risk of problems is higher. When individual services are required to have an uptime of 99% and 5 different services are required for one task, the chance of a successful execution drops to 0.99^5 , which is a just over 95%. A similar problem emerges with response times, as services have to wait for each other and one service can cause delays in many systems.

In order to create transparency regarding the expectations, a Service Level Agreement (SLA) should be specified for each service. When a consumer uses a service, he agrees with the SLA contract stating the guaranteed service levels, possible fees for using the service and possibly, fines for not complying with the terms of this contract [1].

Service level management also concerns the evaluation of services and their SLA contract. The evaluation process should put in balance the benefits a service delivers and the related costs. Services might become obsolete after changes in business requirements or because of the publication of similar and better performing services. The results from the individual service evaluations may serve as input for the service lifecycle management process (see Section 3.4).

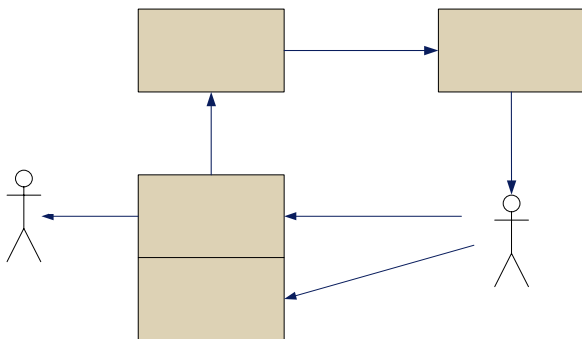


Figure 3: role of tooling in policy enforcement

3.7 Reflection on problems

In chapter 2, some common problems were discussed, based on literature. We briefly reflect on these problems to see how they relate to the phases in our SOA governance lifecycle method.

- *Compliance to standards or legislation, requiring audit trails of IT systems.* This issue is covered by policy management, which allows services to be monitored on compliance to business rules. The required standards need to be considered during service development and can be incorporated into the design guidelines for services.
- *Creating a budget* is done in the second phase of organizational alignment. Here the strategy is translated to the organizational context, where attention is given to ownership and costs of services.
- *Dealing with consequences of service changes* is part of the service lifecycle. Our method addresses this problem in two ways. First, service lifecycle control deals with cataloguing services to make the relationships visible. This allows for impact analysis of changes to services. Second, this phase addresses the importance of correct change management procedures.
- *Ensuring quality of service.* Good quality of services starts with their definition in the portfolio management phase. But more important is the control of design quality in the service lifecycle, where design principles are created and used. Finally, the operational quality of services is dealt with in the final phase of service level management.
- *Changing the attitude of people* is formalized in the second phase of the lifecycle. Getting sufficient support is an important part of creating a strategy. In the second phase a Centre of Excellence is installed to promote SOA in the organization.

4. MATURITY LEVELS FOR SOA GOVERNANCE

The scope of the activities discussed so far in the SOA governance lifecycle is very broad and an immediate adoption of all of them is probably not realistic. Imposing excessive governance procedures can degenerate in a source of unnecessary bureaucracy, which is especially dangerous for SOA-immature organizations. To prevent this, governance should be aligned with the maturity of a SOA. To this purpose, we incorporate a SOA maturity model in our governance approach.

One maturity model that has been designed specifically for SOA is the Service Integration Maturity Model, which defines the business view, methods, applications, architecture and infrastructure for seven SOA maturity levels [2]. While this model is useful for describing the architecture, it does not relate to governance issues. A better solution, in terms of governance, is the business-IT maturity alignment model by Scheper [18], which has been designed for issues on the border between business and IT.

The business-IT maturity model expresses the influence of maturity on business-IT alignment by means of strategy & policy, monitoring & control, organization & processes, people & culture and IT. A consistent maturity level between all of these aspects makes sure that one lagging aspect does not become a bottleneck for improving the organization. The model has been adapted to the SOA governance lifecycle as shown in the table 1. It allows an organization to specify a focus for SOA governance on each of the four maturity levels. This maturity level can be aligned with current business-IT maturity, so that current procedures can be reused for governing SOA. Thus, the lifecycle method allows for increasing maturity by improving the SOA on consecutive cycles.

The proposed maturity model relates with the Capability Maturity Model Integration (CMMI) [20]. CMMI proposes process areas in order to implement improvements to a process. The maturity levels described below can be thought of as implementations of CMMI process areas. SOA governance can be related to several process areas, such as Configuration Management or Project Management and Control.

5. FIELD STUDY

We interviewed persons who have been involved in SOA projects to investigate the current state of affairs with respect to SOA governance. We have collected information from projects in the government, manufacturing and banking sector as well as from an SOA tool vendor. All participants are Dutch organizations (with sizes varying from 500 to 25,000 employees worldwide). Some of them are operating globally. The SOA projects varied from about 5 to 50 directly involved participants. The interviews were set up to investigate the impact of the projects and to find out to what extent phases, as proposed in our SOA governance lifecycle method, are present in projects.

One important observation we made is that the organizational impact of current SOA projects is rather limited and still in an “experimental” phase. There were in The Netherlands some initiatives to launch larger projects, but the fear of losing control and of assuming high risks in such projects has considerably slowed them down. The SOA projects we researched were started either to solve systems integration issues, or to support Business Process Management (BPM) initiatives. However, since the services developed in these workflows were only related to a small number of workflow steps confined to one business activity, governance was not perceived as an issue. Since most projects had a fairly limited scope and services were used in the same part of the organization, in most cases the service consumer and

service provider knew each other. This situation proved to have a major impact on governance requirements. For example, in such a project setting, enforcing policies is not an immediate priority, since policies can be simply agreed upon in a rather informal way. Thus, the focus of SOA governance lies now mostly on the use of service level agreements and change management. Service changes can cause problems even in small systems, since the developer often does not know how his colleagues have used the service. Service level management is, in general lines, carried out in the same way it was done for traditional applications, and therefore it is not perceived as a problem area.

Although all the issues we have included in the SOA governance lifecycle method have been recognised as being relevant, the interviews have shown that there is no immediate necessity to apply them on the current SOA projects. This is caused by the current level of maturity and limited complexity of SOA projects.

This finding confirms our claim with respect to the necessity of aligning SOA governance and SOA maturity. The projects we have investigated were often at the first, or sometimes partially at the second maturity level. However, our interviewees expected that as projects will grow in size and complexity, the demands on other aspects SOA governance will grow as well.

6. RELATED WORK

Below we discuss and relate our approach to several existing SOA governance models that have been examined during our research. It should be noted that several relevant aspects of these models have been incorporated in our approach.

According to IBM’s model ([5], [4]) the SOA governance lifecycle has four phases: plan, define, enable and measure. They relate directly and support the four phases of the IBM service lifecycle process: model, assemble, manage and deploy. Thus IBM takes a very IT driven approach to SOA which is complementary to our approach that sees SOA predominantly as a business and organizational driver. Although the IBM model addresses some business aspects of SOA, it mostly defined from the point of view of developing good applications, instead of delivering efficient business solutions. For example, it is mentioned that organizations should “define ownership”, nonetheless without providing any guidelines or approach for doing this.

CBDI’s SOA governance framework [29] considers business and organizational governance as distinct parts of SOA governance, next to the IT-related components of SOA governance (i.e.,

	1. Pioneer	2. Department	3. Enterprise	4. Network
Define SOA strategy	Look for most profitable services	Involve stakeholders	Align SOA with organization and IT	Make SOA part of corporate strategy
Align organization to SOA	Centralize governance structure	Record ownership of services	Translate business processes to services	Decentralize governance structure
Manage service portfolio	Agree on scoring criteria	Formalize portfolio process	Migrate & integrate legacy into SOA	Define how services should be sourced
Control service lifecycle	Develop best practices	Formalize development process	Catalogue services in a library	Align service lifecycle with networktrack changes partners
Incorporate policy enforcement	Create service guidelines	Formalize policies	Report on policy use & violations	Specify enforcement points between partners
Service level management	Identify critical services	Formalize simple contracts for services	Monitor usage and check SLA compliance	Billing of service use between partners

Table 1. SOA governance focus per business maturity level

portfolio, architectural, provisioning, usage and operational governance). The several governance categories mentioned by the CBDI model have a great similarity or overlap with the areas mentioned in our approach. Nevertheless, we were not able to get any information explaining why the model includes these categories. The main difference between the CBDI model and the model presented here is the distinction CBDI makes between business and organizational issues and IT issues in SOA governance, which may lead again to a divide between business and IT by leaving the impression that business does not need to be involved in the IT governance issues. Another difference is that the CBDI model does not include a lifecycle or process model. Furthermore, it does not define corresponding approaches/techniques or deliverables for the aforementioned governance categories, which makes it difficult to use it in a practical setting.

The SOA lifecycle model from MomentumSI [25] explores some important concepts related to SOA governance but it is far from providing methodological support for SOA governance processes. Vazquez relates governance with eight domains (i.e., portfolio management, planning, service development, consumer development, integration configuration, reporting & analytics, program promotion & marketing, platform, service, integration production support). However, in Vazquez' vision, SOA governance is mostly about defining/enforcing policies for all the aforementioned domains. Portfolio management, service development and reporting are present in our framework as well, but play a different role. We note in Vazquez' model the distinction between tactical and strategic governance. While these two levels will not be an issue in smaller scale SOA projects, large project organizations will have to deal with this distinction as well.

Forrester has specified a list of major SOA governance areas [10]: portfolio governance (responsible for managing application & service portfolios), technology governance (incorporating the use of standards, patterns and technologies), project governance (where service implementation and policy specification is done); service-level governance (including policy enforcement and service-level management).

As it can be seen from this brief overview and from an exploration of the field literature (e.g., [4], [5], [6], [10], [21], [23], [29], [29], [28]) most authors agree on the role policies and performance measurement plays in SOA governance, although the approaches proposed to support them vary a lot. Besides, variations can be also noted in the existing definitions of SOA governance ranging from just accountability to broader definitions that include many management processes as well.

7. CONCLUSIONS

In this paper we have proposed a methodological approach to SOA governance. We have chosen for an incremental approach, as we are of the opinion that using such an approach increases the chance of delivering good SOA governance. SOA governance measures should be connected to issues that play a role in the current situation. The maturity framework we incorporated in our model differentiates between problems on different SOA's and points out problem areas an organization is likely to encounter in their current maturity level.

Our method is based on the assumption that SOA governance should be a joint responsibility of the business and IT. SOA is different from other IT architectures, in that it is able to provide services related to individual process activities in the organization. This requires organizational input about the conditions that need to be managed in the context of those processes. Strong communication links and sharing of expertise are needed to make the benefit of better organization-IT alignment real.

It is important to realize that SOA governance is a continuous process. As one SOA project is completed and becomes operational, its results should be incorporated in an improved strategy, thus triggering a new iteration of our lifecycle model. Most of the phases in the governance lifecycle are actually continuous tasks. For example, service lifecycle management is such a task, since at any time change requests may arrive that need to be handled and governed. SOA governance is not a process, but a matter of continuously aligning strategic goals, new tactical opportunities and to use gained experience.

The amount of literature accounting on results and validation studies in this area is very limited. Obviously, more research is needed to further sharpen the focus of SOA governance and model it in such a way that it has practical use. The method presented here is a first step in this direction. Nevertheless, our method still needs to be further validated during SOA projects.

8. REFERENCES

- [1] Amar, M. and Muhammad, N., *Service oriented architecture Registry*, available at <http://idenet.bth.se/servlet/download/element/36064/SOA+Registry.pdf>
- [2] Arsanjani, A. and Holley, K., The Service Integration Maturity Model: achieving flexibility in the transformation to SOA, In *Proceedings of the IEEE International Conference on Services Computing (SCC'06)* (Chicago, US, 18-22 September 2006), 515.
- [3] Aziz, S. Obitz, T., Modi, R., Sarkar, S., *Enterprise Architecture: A governance Framework, Part I and Part II*, White papers, Infosys, September 2005, available at <http://www.infosys.com/services/systemintegration/EA-Governance-1.pdf>
- [4] Bieberstein, N.; Bose, S.; Fiammante, M.; Jones, K. and Shah, R., *Service-oriented architecture compass: business value, planning and Enterprise Roadmap*, IBM Press, Upper Saddle River, 2005.
- [5] Brown, W.A., Moore, G. and Tegan, W., *SOA governance—IBM's approach, Effective governance through the IBM SOA Governance Management Method approach*, White paper, August 2006, available at ftp://ftp.software.ibm.com/software/soa/pdf/SOA_Gov_Proc_ess_Overview.pdf
- [6] Carter, S., *The new language of business: SOA & Web 2.0*, IBM Press, Upper Saddle River, 2007.
- [7] Chung, S.; Chul An, J.B. and Davalos, S., Service-Oriented software reengineering: SoSR, In *Proceedings of the 40th Hawaii international conference on system sciences (HICSS'07)* (January 3-6 2007), 172c.

- [8] Dan, A., Web services on demand: WSLA-driven automated management, *IBM systems journal* 43, 2004, 136-156.
- [9] Erl, T., *Service-Oriented Architecture: concepts, technology and design*, Prentice Hall, Upper Saddle River, NJ, 2005
- [10] Heffner, R., *SOA investment strategies: Case studies on how enterprises are paying for SOA*, Forrester Research, Cambridge USA, May 19, 2006.
- [11] IT Governance Institute, *COBIT 4.1*, available at <http://www.isaca.org>
- [12] IT Governance Institute, *Board Briefing on IT Governance*, 2nd Edition, Rolling Meadows, IL, 2003.
- [13] Legner, C. and Heutschi, R., SOA adoption in practice – findings from early SOA implementations, In *Proceedings of the 15th European conference on Information Systems (ECIS 2007)* (St. Gallen, Switzerland, June 7-9-2007), 1643-1654.
- [14] Lewis, G; Morris, E.; O'Brien, L.; Smith, D. and Wrage, L.; *SMART: The service-oriented migration and reuse technique*, Technical Report CMU/SEI-2005-TN-029, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2005.
- [15] Mitra, T, *A case for SOA governance*, published 16-08-2005, viewed 17-07-2007 at <http://www.ibm.com/developerworks/webservices/library/ws-soa-govern/>
- [16] Nadhan, E.G., Service-Oriented Architecture: Implementation Challenges, *Microsoft Architect Journal* 2, April 2004, 24-32
- [17] Papazoglou, M.P., Service oriented architectures: approaches, technologies and research issues, *VLDB Journal* 16, 2007, 389-415.
- [18] Progress Software, *Why runtime governance is critical for SOA: a SOA Primer*, 2005, available at: <http://www.actional.com/resources/whitepapers/>
- [19] Scheper, W., *Mastering complexity using the Business Maturity Model*, May 2005, available at http://www.deloitte.com/dtt/cda/doc/content/nl_eng_report_b_usinss_maturity_model_230505x.pdf
- [20] Software Engineering Institute. *CMMI for development, version 1.2. 2006*, <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>
- [21] Systinet, SOA governance: balancing control & flexibility, september 2006, available at: http://www.systinet.com/resources/white_papers
- [22] The Open Group, *TOGAF™ Version 8.1.1 "Enterprise Edition"*, 2006, <https://www.opengroup.org/architecture/togaf8-doc/arch/>.
- [23] TIBCO, SOA governance best practices: an introduction, whitepaper from www.tibco.com, published November 2005.
- [24] Van den Dobbels, R., *Security in Service-Oriented Architectures*, Master thesis, University of Rotterdam, The Netherlands, 2007.
- [25] Vazquez, E., *Service Oriented Architecture Governance: The Basics*, viewed 15-06-2007, published 11-06-2007 at <http://www.infoq.com/articles/soa-governance-basics>
- [26] Von Goetz, J., *Methodology for Service Identification and Service Design within Collaborative Service Governance Lifecycle Reference Framework*, Master thesis, University of Mannheim, Germany, 2007.
- [27] Webb, P. Pollard, C. Ridley, G., Attempting to define IT governance: wisdom or folly?, In *HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, 04-07 Jan. 2006, Volume: 8, p.194a- 194a.
- [28] Webmethods, SOA governance: enabling sustainable success with SOA, published October 2006, viewed 22-02-2007 at http://www1.webmethods.com/PDF/whitepapers/SOA_Governance.pdf
- [29] L. Wilkes, Good Governance Practices for SOA, CBDI webcast, published 20-03-2007, viewed 28-03-2007 at http://whatis.bitpipe.com/detail/RES/1173402876_505.html
- [30] Windley, P., *Teaming up for SOA*, Infoworld.com, January 03 2007, available at http://www.infoworld.com/article/07/03/05/10FEcollabgov_1.html