2009 Third IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2009)

Challenges of Governance Approaches for Service-Oriented Architectures

Michael Niemann[1], Christian Janiesch[2], Nicolas Repp[1], and Ralf Steinmetz[1], Fellow, IEEE

[1] KOM – Multimedia Communications Lab, Technische Universität Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany, {niemann | repp | steinmetz}@kom.tu-darmstadt.de
[2] SAP Research CEC Brisbane, SAP Australia Pty Ltd, Building A4, Level 7, 52 Merivale Street, South Brisbane QLD 4101, Australia, c.janiesch@sap.com

***Abstract*— IT Systems in companies nowadays are confronted with constantly changing market conditions, new competitive threats and an increasing number of legal regulations. The service-oriented architecture (SOA) paradigm provides a promising way to address these challenges at the level of the company's IT infrastructure. These challenges and the management of the introduced complexity and heterogeneity are targeted by SOA Governance approaches. Hereby, the basic structure of IT Governance frameworks is applicable to SOA; however, they lack applicability concerning some SOA-specific challenges. In this paper, we discuss deficiencies and provide insights of what regulation challenges a SOA Governance approach is actually required to be capable of, in particular in the area of service lifecycles and service marketplaces.**

***Index Terms*— Service-oriented architecture, SOA Governance, service lifecycle governance, governance at service marketplaces**

I. Introduction

The SOA paradigm is a holistic approach towards the execution of business processes within enterprise architectures (EA). Services represent a central aspect and foundation of the architectural paradigm SOA. Non-composite services can be defined as "atomic invariable building blocks that can be combined flexibly over open communication mechanisms" [14]. Their functionality scope is small and they appear in multiplicity. Generally, services are designed to support *reusability* in different scenarios by simple reconfiguration. Service functionalities can be *automatically discovered* via service brokers or registries. Services are centrally registered at a database which provides information about the services upon request. While interacting, services are loosely coupled. This means mutual *association* via messages. Dependencies are minimized to mere awareness, facilitating a number of operations, e.g., their replacement by other services during runtime. Service operations always involve several parties or stakeholders. Services therefore adhere to a communications contract, a Service Level Agreement (SLA), defined by one or more service descriptions and related documents in order to regulate and control service execution [8, 10].

These characteristics make an SOA a powerful, flexible, easy to operate candidate for a company's enterprise architecture.

The SOA paradigm offers a number of advantages compared to common EA. However, these advantages imply challenges emerging from the large numbers of services and their heterogeneity, such as the urgent need for permanent monitoring and control of services – guidance for *service handling* is required. Therefore, a critical aspect for success in the adoption and the operation of SOA is governance. The primary goal of SOA Governance is compliance, i.e. compliance to intra-company, normative, or legal standards (required by, e.g., the Sarbanes Oxley Act, Basel II etc.). Following specific guidelines in a top-down approach, an SOA is directed through a number of maturity levels or development steps – it is adopted, commissioned, operated, and continuously monitored and checked for adherence to regulations.

In the area of IT Governance, a number of existing frameworks provide structures, action scope, guidelines, reference processes, and best practices. However, the basic structure of IT Governance frameworks often exceeds the needs of SOA, while lacking applicability concerning SOA-specific challenges, e.g., cross-company cooperation issues like inter-company service deployment. Hence, in order to meet SOA Governance requirements, existing IT Governance frameworks need to be extended [35].

Diligent SOA Governance has been recognized in recent years as a major requirement for successful adaptation and operation of an SOA, especially for large systems. Governance in general, be it political governance, Corporate or IT Governance, deals with the successful governing of organizations or projects. SOA Governance elaborates guidelines and rules that need to be adopted and realized by the affected management processes. It provides a means to effectively exploit the capabilities of SOA [17, 25]. The achievement of governance goals is supported by SOA Maturity Models and respective governance mechanisms.

SOA Governance focuses on the smooth adoption and successful operation of an SOA as the EA in a company. By providing guidelines, responsibilities, and reference processes, it ensures its integrity and adaptability to business and administration processes. Governance tools support the monitoring and control of services concerning the alignment to business processes. A best practices catalog serves as a repository of implementation recommendations that are continuously supplemented, supporting all of the mentioned procedures. Besides the achievement of IT goals and the realization of business-IT alignment, a further goal of SOA Governance is to realize system adherence to regulations and standards, such as ISO norms or internal regulations.

Existing approaches to governance frameworks do not fully cover special SOA Governance requirements. In this

2009 Third IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2009)

paper we identify particular SOA characteristics that need to be addressed by respective governance approaches – and that have not yet been integrated into governance efforts. We identify and outline requirements of future SOA Governance approaches based on survey findings and typical governance requirements of service-oriented IT systems.

In section 2 we focus on existing SOA Governance approaches, discussing particularities and identifying typical components. In section 3 we outline and discuss identified deficiencies of existing SOA Governance approaches and propose concrete measures to address these.

II. SOA GOVERNANCE: SURVEY FINDINGS

There are few scientific contributions that have dealt with SOA Governance so far. Nevertheless, a definition is important. There are numerous definitions of SOA Governance, all diverging in focus. In the context of this paper, based on [20], we understand SOA Governance as *a holistic long-term management model. It guarantees sufficient adaptability and integrity of an SOA system as well as the ability to check services concerning capability, reusability, security, and strategic business alignment. Overall goals are SOA compliance and the guarantee of reusability and standardization throughout the system*.

Numerous frameworks have been specified for IT Governance, e.g., CObIT, ITIL, ISO 17799, and many more. Each of them addresses different aspects of a company's IT management. While the ISO 17799 standard primarily targets security management [11], the IT Infrastructure Library (ITIL) mainly deals with IT process definition [21]. CObIT defines 34 reference processes as control framework, more tightly aligned with the business objectives of the organization than with operational issues [12]. Comparing all these frameworks discloses that they complement each other and, as a matter of fact, CObIT represents a frame integrating all other frameworks – it has, so far, become a *de facto* standard for IT control globally.

In the area of SOA Governance frameworks, there are a few research contributions. Existing concepts are mostly motivated by software providers that offer SOA business solutions and closely align their SOA Governance perspectives with their products ("the fairly narrow view" [2]).

An overview of the different understandings of this topic is given by [20], providing a survey and analysis of ten approaches to SOA Governance. Ten typical components of SOA Governance have been identified by times included in existing approaches (cf. Table 1). Detailed descriptions can be found in [20].

Examination of the comparison shows that components that are covered by IT Governance frameworks are considered less important for SOA Governance. *Role and accountabilities, metric models,* and *impact on behavior* are all not taken into account at the first place and, besides the first, they are part of standard IT Governance frameworks [32]. This illustrates one difference between the approaches – some aim at completely covering SOA Governance challenges, others, not considering maturity models and behavioral impact, build on the additional implementation of parts of IT Governance approaches.

Implementing parts of an IT Governance approach and additionally following an SOA Governance model increases cost and time efforts in IT departments – a diligent SOA Governance model should cover not less than all SOA-related regularization aspects.

Few approaches emphasize mechanisms to impact behavior of employees or people working with the system, as well as SOA Maturity Models - although these aspects seem important for the operation of an SOA [17, 32, 2, 5]. Common IT Governance frameworks like CObIT address these issues. Although there are plenty of maturity models, e.g., [28, 13], they did not yet establish as a prevailing element of SOA Governance approaches, although they are part of, e.g., CObIT. This supports the need mentioned at the beginning to bring together best practices from IT Governance and additional SOA-related regulation requirements – an extension of either SOA or IT Governance frameworks to allow for the respective needs, as already stated by [Woolf].

Overall, *organizational changes, roles/ accountabilities, policy catalogs,* and *service lifecycle,* integrated in the majority of approaches, can be considered the most important elements.

According to the service lifecycle, there are, in fact, lots of additional characteristics in a service lifecycle as opposed to a traditional software lifecycle. Some lifecycle models consider more stakeholders than service producer. Additionally, services can be thought of as goods that are traded online between service consumers, providers, and intermediaries, affecting developers, platform hosts, infrastructure providers and many more – all of these can be embodied by companies that interact in open service marketplaces.

Legend ○ – integrated ● – integrated and specified in detail × – not integrated	Organizational changes	SOA Maturity Model	Roles/ accountabilities	Best practices	Metrics model	Impact on behavior	SOA Lifecycle	SOA Roadmap	Policy Catalog	Service Lifecycle
Brauer/ Kline [6]	×	○	×	×	×	×	×	●	●	●
Bieberstein [4,5]	●	×	●	●	○	●	×	●	○	×
webMethods [18]	○	○	×	○	×	×	●	×	●	○
Software AG [27]	●	●	●	●	×	×	×	●	●	●
BEA Systems [3]	×	×	×	×	×	×	●	×	○	●
SAP [23, 24, 31]	●	×	×	×	○	×	×	×	●	○
Oracle [1]	●	○	●	●	○	○	●	×	●	○
IBM [19, 35]	○	×	○	●	○	×	●	×	×	●
Marks/Bell [17]	●	×	●	●	●	●	●	○	●	○
Schelp/Stutz [25]	●	×	●	×	×	×	○	×	○	×
Weill/ Ross [32]	●	×	●	×	●	●	×	×	○	×

Table 1 Comparison of SOA Governance Approaches [20]

2009 Third IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2009)

III. SOA GOVERNANCE CHALLENGES

As outlined, service-oriented architectures introduce a number of challenges for IT management [1, 2, 4, 17, 25]. We investigate service lifecycles as well as regulation requirements in service marketplaces and point out regulation requirements that have not been addressed yet.

Service Lifecycles

According to the analysis performed in [20], service lifecycle management is a central aspect of SOA Governance. Services are common software artifacts – however, their functionality scope is small and they appear in multiplicity. Services are delivered to the customer and installed, i.e., deployed, in completely different environments and contexts compared to common (monolithic) software. For these reasons, service lifecycles differ from common software engineering lifecycles [18, 6, 33].

Service lifecycle management (SLCM) targets the steering, direction, and control of all services of the system in all lifecycle phases. The goal is to assure manageability and conformity in spite of the large service numbers. Important aspects are *change management procedures, service deployment, service granularity,* and *consumer integration*.

Service lifecycle approaches are often divided into three super-phases: *design time, runtime,* and *change time*. The lifecycle phases up to *deployment* are commonly referred to as *design time*. The usage phase is referred to as *runtime*, followed by *change time,* which addresses service change and revision. [9, 29, 18]

An important part of SLCM are *change management procedures*. In case a service is to be changed in an SOA, this normally affects a big number of composite services or applications. As services are often bound to SLAs or have cost impact in other ways, this issue is especially focused on by service providing companies. This might also be the reason that gave birth to "change time", as many service lifecycle approaches define [18, 6, 29, 9]. In fact, service change is a crucial issue. However, it closely corresponds to changing a software component or a system which is not a new issue in software engineering (SE). Although services might have higher business or cost impact, SE principles can also be applied here. When handled as common software artifacts, services are marked deprecated and change is addressed and accomplished by a new lifecycle iteration. Finally, the deprecated service is taken off-line. Thus investigating the necessities shows that the characteristics of change time are covered by other phases and that there is, strictly speaking, no need for "change time".

Service deployment covers more crucial activities than installation does in a traditional software lifecycle. Additionally to being delivered to the customer and installed as any software system would be, services must be registered at one or multiple service registries. This implies the transmission and installation of policies, documentation, and metadata of the service. Furthermore, when lacking according guidelines and measures, it is possible that in a closed SOA-system a service is used without having been registered at a service registry or repository. These "rogue services" can lead to incompleteness and inconsistencies of service registries, as well as duplicate implementation and provisioning of services [26, 16].

Service granularity is an often addressed, but unsolved issue yet. The basic challenge here is to define functionality boundaries for services during the first phases of the SOA and service lifecycle. Functionality requirements must be divided in parts such that each of them is best suited to be addressed by one service. Largely, the relationship between service number and 'task size' is to be defined – which actually varies from case to case. [15]

A SOA system has a much more dynamic nature than deployment and usage of common software systems. Services can be requested, used, replaced, and discarded at any time, even for free as long as SLAs are not violated. This implies a more frequent interaction of service provider and consumer in general. This covers publishing services at known service registries, informing consumer upon service change, managing change requests, and SLA management (process reliability etc). In order to address *consumer integration*, some approaches propose a service lifecycle consisting of a consumer and a provider (or developer) part [29, 33, 9, 18]. The lifecycles basically comprise the phases *Operation, Discovery*, and *Handshake* on the consumer side. This illustrates the ability of being searched, discovered, as well as the negotiation of usage terms per service and service provider. These processes can be triggered when in operation phase, i.e., this additionally considers service substitutability.

Out of 10 approaches, one incorporates a service lifecycle considering the service consumer [20]. However, the incorporation of the service consumer into the SLCM approach seems advisable, as in fact, the way a company's services are handled have a strategic dimension. An important customer, for example, might prescribe interface guidelines due to legacy applications that imply security challenges to the service providing company. These issues show some differences between traditional software management and service lifecycle management that need to be considered by SOA Governance approaches.

In summary, the following aspects are to be taken into account when defining a governance approach for an SOA: *the service change process, service deployment, service granularity,* and *consumer integration*.

Service lifecycles in SOA Governance most commonly consist of seven phases: *system analysis*, covering global requirements, service creation policies, service granularity decisions, *service definition, service design, service development, service deployment, service operation*, and *service retirement.* Figure 2 shows a consolidated approach based on [9, 1, 6, 18, 29].

A defined *service change process* is important. In practice, obviously, this is often addressed by defining a "change time" – which is not useful as it obfuscates the need of cycle reiterations. In fact, a concrete process is needed, covering customer notification, interaction with the service registries, i.e., service description management, a

deprecated state, as well as SLA and contract management that prevent SLA violations due to change. Interface design guidelines can minimize visibility of the performed changes, aiming to keep the service description consistent. Amongst others, as part of governance policies, these steps are to ensure cost-neutral, transparent, flexible service changes.

Consistent handling of *service deployment* covers the definition of service registration conditions and procedures (metadata, registry specification, scope of availability) as well as installation procedures at the service host or intermediaries, covered by adequate diligent contract management. Additionally, sticking to the mentioned lifecycle (by regulation) avoids rogue services.

Granularity decisions can be supported by service blueprints, service definition scope concepts that apply in the definition phase, and guidelines, that structure typical service tasks by defining task levels. Measures like these, individually aligned to specific SOA requirements, help with globally controlling and keeping service granularity in a manageable range.

Beyond the usual control function of a service lifecycle, the *consideration of the consumer* emphasizes particularly important aspects of each phase and supports a better alignment of services to the requirements, such as conformity of description, interfaces, granularity, and many more. Such a lifecycle, explicitly covering the service consumer, allows for the special relation to the service customer, in particular considering the discovery process, which is a strategic factor at service marketplaces. The negotiation process is explicated, which also is of central relevance. Explicit awareness of the service consumer lifecycle supports internal process management and strategy issues.

Integration of these aspects make a service lifecycle as shows in figure 2 a governance and regulation supervising tool. Enforcing the phases makes it a strong control and revision tool for application in SOA Governance in order to assure the above regulations.

Stakeholder-spanning governance at open service marketplaces

The concept of service marketplaces, related to component marketplaces, has become attractive again in the last years. Envisioned as in the "Internet of Services" by the government-funded German research project Theseus/ TEXO, services become tradable, i.e. they can be offered, searched, found, sold, enhanced, and resold as part of service compositions [30, 34]. Given such an *open service marketplace*, service providers offer services to service consumers via portals (operated by a market-maker), the use of infrastructure services, and a system of registries and repositories. A large number of different services offered by different service providers and intermediaries at different service hosts are available (as well as infrastructure and similar services). In particular, service providers compete against each other in providing services with adequate performance, price, functionality, and further features. The purpose of open service marketplaces is "to create opportunities for buyers and sellers to meet and conduct business electronically, or aggregate service supply/demand by offering added-value services and grouping buying power" [22]. Service marketplaces are enabled by standardization of interfaces, loose service coupling enabling replacement, and differ from software component marketplaces in a way that integration is much easier due to the realized interoperability.

The overall goal is to realize full service tradability, where the term service covers e-services as well as business services ('b-services'). The vision bears the opportunity of creating new self-regulating online service marketplaces based on the standardization and interoperability of these services. Besides service providers and consumers, there are a lot of further actors such as service intermediaries, and service hosts, each of which operate and define an own internal proprietary service management system [7, 22]. In order to cooperate at the marketplace with the goal of adding value to services, e.g. by refining or combining services, a service-supply-chain-management is required, i.e. the co-

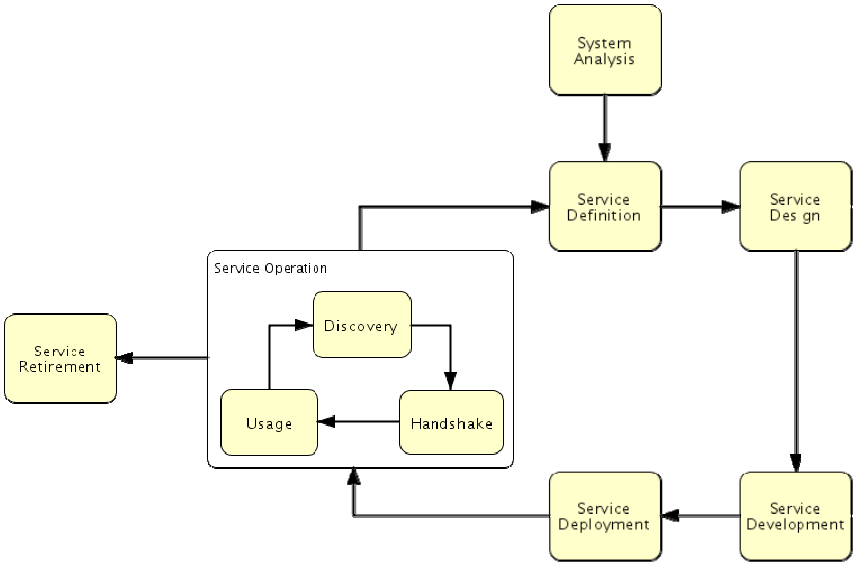

Fig.2 Consumer Service Lifecycle

operation throughout the whole service-centered value-added process. This can include pricing models, interface standards, business strategy, technology, process, and infrastructure alignments, service portfolio coordination, and so on. In order to do so efficiently, modern service-oriented enterprises will be obliged to bundle their SOA Governance efforts. Concluding, compatibility of governance strategies and efforts is mandatory when considering service offering at a marketplace and, e.g., forming a value-added supply chain,

As a further aspect, marketplace regulation is important. Legal issues, the formulation of terms of usage, and internationality are aspects to be considered, e.g., legal terms are not equivalent depending on the geographical place of service execution and consumption, as well as local laws might require terms to be expressed in the local language in order to form a valid contract. Security is an issue when freely providing services. Data security has to be guaranteed and trust between business partners has to be established on a technical level and assured throughout the marketplace, e.g., by dynamically configurable encryption methods etc. Market regulation governance has to define and provide according rules and measures in these areas to assure reliable long term operation.

Concluding, compatibility and homogenizations of regulations is crucial for cooperation on an open service marketplace. So far, these cross-organizational aspects are not addressed by IT Governance frameworks as well as SOA Governance approaches. Service supply chain-related governance is a decentralized issue that must be addressed by *the common* SOA Governance approaches. Marketplace regulation is the duty of the portal provider or market-maker governance.

IV. Conclusion

Governance frameworks address aspects of an SOA that need to be regulated to guarantee business-IT alignment and successful long-term operation. However, for SOA-specific regulation requirements beyond current IT Governance scope, current literature and industry efforts lack applicability in some aspects. In this paper, we discussed particular aspects of service-oriented systems that exceed coverage by existing frameworks.

We identified two major areas, where future SOA Governance approaches are to provide steering and control support in order to operate an SOA successfully: service lifecycle management addressing service-specific phases and stakeholder integration, as well as open service marketplaces and their implications on governance approaches for marketplace actors.

Concerning service lifecycle management, the precise definition and the regulation of particular phases are the most important aspects. In particular, service granularity is an important detail whose impact will increase with growing adoption of service-oriented systems and service marketplaces. Accurate service deployment regulations ensure consistent operation and well-maintained registries and cover tight integration of the service customer. A major challenge is to define and provide a reliable service change process. Using an adequate service lifecycle, service lifecycle management can be deployed as a powerful governance instrument addresses these challenges.

SOA Governance concepts do not yet foresee the integration of regulating the interconnection of cooperating partners in open service marketplaces, i.e. compatibility in service control. In particular, existing frameworks do not specify regulation support in the area of *service supply-chain management*.

Existing respective governance models do not address all outlined aspects to an extent which relates to their importance in the operation of an SOA. Service lifecycle management and the field of open service marketplaces give rise to a number of crucial tasks to be regulated by governance that have not been integrated in according frameworks yet.

Concluding, there are a number of aspects and characteristics of SOA systems that are not yet addressed exhaustively by SOA Governance approaches. This may be on one hand due to a lack of awareness for the necessity of SOA Governance, and on the other hand due to the orientation of the majority of governance approaches at specific vendor's products.

Future work covers further investigation of proposed models for SOA Governance and their validation concerning the requirements and challenges that an SOA really makes. In the Theseus/ TEXO project, we currently address the elaboration of a service host-centered SOA Governance framework, where we address these challenges in detail.

V. Disclaimer

This work is supported in part by the E-Finance Lab e. V., Frankfurt am Main, Germany (www.efinancelab.com).

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents.

VI. References

[1] Mohamad Afshar. SOA Governance: Framework and Best Practices. Oracle Whitepaper, May 2007. Retrieved July 20, 2007 from http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf.

[2] Paul Allen. SOA Governance: Challenge or Opportunity? CBDI Journal, pages 20–31, April 2008. Retrieved July 20, 2008 from http://www.cbdiforum.com/secure/interact/2008-04/challenge_opportunity_br.php.

[3] BEA Systems, Inc. Service Lifecycle Governance. BEA Whitepaper, 2006. Retrieved Oct 20, 2007 from http://www.itworldcanada.com/Admin/Pages/Assets/DisplayAsset.aspx?id=e0a24263-a10a-4887-8d45-582261587176.

[4] N. Bieberstein, S. Bose, L. Walker, and A. Lynch. Impact of Service-oriented Architecture on Enterprise Systems, Organizational Structures, and Individuals. *IBM Systems Journal*, 44(4):691–708, 2005.

[5] Norbert Bieberstein, Sanjay Bose, Marc Fiammante, Keith Jones, and Rawn Shah. Service-Oriented Architecture (SOA) Compass - Business Value, Planning, and Enterprise Roadmap. IBM developerWorks, 2006.

[6] Ben Brauer and Sean Kline. SOA Governance: A Key Ingredient of the Adaptive Enterprise, February 2005. http://www.managementsoftware.hp.com/products/soa/swp/soa_swp_governance.pdf Last checked Oct 10, 2007.

[7] Julian Eckert, Deniz Ertogrul, Andr´ Miede, Nicolas Repp, and Ralf Steinmetz. Resource Planning Heuristics for Service-oriented Workflows. In 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pages 591–597. IEEE, Dec 2008.

[8] Thomas Erl. Service-Oriented Architecture - Concepts, Technology, and Design. Prentice Hall Professional Technical Reference, Boston, 2005.

[9] Qing Gu and Patricia Lago. A stakeholder-driven Service Life Cycle Model for SOA. In Proceedings of IW-SOSWE 2007, Dubrovnik Croatia, September 2007.

[10] Michael N. Huhns and Munindar P. Singh. Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing, 9(1):75–81, 2005.

[11] International Organization for Standardization (ISO). ISO 17799. http://www.17799central.com/.

[12] IT Governance Institute (ITGI). CobiT 4.1, 2007. http://www.itgi.org/cobit.

[13] Wolfgang Johannsen and Matthias Goeken. Referenzmodelle für IT-Governance - Strategische Effektivität und Effizienz mit COBIT, ITIL & Co. dpunkt.verlag Heidelberg, 1. edition, 2007.

[14] Ulrich Kalex. Von der Geschäftsarchitektur zur SOA-Governance. In Gernot Starke and Stefan Tilkov, editors, SOA-Expertenwissen, pages 325–340. dpunkt.verlag Heidelberg, 2007.

[15] Wolfgang Keller. SOA-Governance: SOA langfristig durchsetzen und managen. In Gernot Starke and Stefan Tilkov, editors, SOA-Expertenwissen, chapter 19, pages 325–340. dpunkt.verlag Heidelberg, 1. edition, 2007.

[16] James Kobielus. SOA Governance: Preventing Rogue Services, June 2006. Retrieved July 2, 2007 from http://www.networkworld.com/supp/2006/ndc3/062606-ndc-soa-governance.html.

[17] Eric Marks and Michael Bell. SOA: A Planning and Implementation Guide for Business and Technology. John Wiley & Sons, Inc., New Jersey, USA, 2006.

[18] Miko Matsumura. The Definitive Guide to SOA Governance and Lifecycle Management, March 2007. Retrieved Feb 2, 2009 from http://www.scribd.com/doc/7056416/Guide-to-SOA-Governance.

[19] Gary McBride. The Role of SOA Quality Management in SOA Service Lifecycle Management. IBM Developer Works, March 2007. Retrieved July 20, 2007 from ftp://ftp.software.ibm.com/software/rational/web/articles/soa_quality.pdf.

[20] Michael Niemann, Julian Eckert, Nicolas Repp, and Ralf Steinmetz. Towards a Generic Governance Model for Service-oriented Architectures. In Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS 2008), Toronto, ON, Canada, 2008.

[21] Office of Governance Commerce (OGC). IT Infrastructure Library, 2007. http://www.itil.org.

[22] M. P. Papazoglou and D. Georgakopoulos. Service-Oriented Computing (Introduction). Communications of the ACM, 46(10):24–28, 2003.

[23] SAP AG. Enterprise Services Design Guide, 2005. https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/943e83e5-0601-0010-acb5-b16258f5f20a .

[24] SAP AG. Governance for Modelling and Implementing Enterprise Services at SAP Enterprise SOA Solution Management, April 2007. https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/f0763dbc-abd3-2910-4686-ab7adfc8ed92 .

[25] Joachim Schelp and Matthias Stutz. SOA-Governance. In Hans-Peter Fröschle and Stefan Reinheimer, editors, Serviceorientierte Architekturen (SOA), number 253 in HMD Praxis der Wirtschaftsinformatik, pages 1–10. dpunkt verlag, Heidelberg, February 2007.

[26] T. G. J. Schepers, M. E. Iacob, and P. A. T. Van Eck. A lifecycle approach to SOA governance. In SAC ’08: Proceedings of the 2008 ACM symposium on Applied computing, pages 1055–1061, New York, NY, USA, 2008. ACM.

[27] Software AG. SOA Governance – Rule your SOA, 2005. Retrieved July 11, 2007 from http://www.softwareag.com/de/Images/WP_SOA_Governance_D_tcm17-22130.pdf .

[28] Sonic Software Corporation. A New Service-oriented Architecture Maturity Model, 2006. Retrieved July 11, 2007 from http://www.sonicsoftware.com/solutions/service_oriented_architecture/soa_maturity_model/index.ssp.

[29] Systinet. SOA Governance: Balancing Flexibility and Control Within an SOA, 2006. Retrieved Feb 2, 2009 from http://www.vivit-worldwide.org/uploadedFiles/Product_Centers/Service_Oriented_Architecture_(SOA)/09EX001HPMAY07_Mercury-Systinet_SOA_Governance.pdf.

[30] Theseus/TEXO. TEXO – Business Webs in the Internet of Services, 2007. http://theseus-programm.de/scenarios/en/texo.html Last checked Feb 2, 2009.

[31] Jürgen Wagner and Friedhelm Krebs. PIC Council Charter, April 2004. Retrieved Feb 19, 2008 from https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/d987b590-0201-0010-1684-c4754d40ddb9.

[32] Peter Weill and Jeanne W. Ross. IT Governance - How Top Performers Manage IT Decision Rights for Superiour Results. Harvard Business School Press, Cambridge, MA, 2004.

[33] Phillip J. Windley. SOA Governance: Rules of the Game. INFOWORLD.COM, pages 29–35, 2006. Retrieved Oct 20, 2007 from http://akamai.infoworld.com/pdf/special_report/2006/04SRsoagov.pdf.

[34] Matthias Winkler, Jorge Cardoso, and Gregor Scheithauer. Challenges of Business Service Monitoring in the Internet of Services. In iiWAS ’08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, pages 613–616, New York, NY, USA, 2008. ACM.

[35] Bobby Woolf. Introduction to SOA Governance. IBM Developerworks, June 2006. Retrieved July 22, 2007 from http://www.ibm.com/developerworks/library/ar-servgov/.