A conceptual information model for service management dimensions

Roman Belter, Rolf Kluge, Thomas Hering, Holger Müller

Information Systems Institute
University of Leipzig
Marschnerstr. 31
04109 Leipzig, Germany
{rbelter, rkluge, hering, hmueller}@wifa.uni-leipzig.de

Abstract: Service-oriented Computing (SOC) is known as the leading paradigm for the creation of agile and flexible enterprise IT infrastructures. The implementation of enterprise-wide Service-oriented Architectures (SOA) is a complex task. In most cases, more evolutionary approaches are used to deal with the arising complexity. However most of the design methodologies and implementation strategies focus on more technical, service realization specific aspects. Challenges regarding the definition and the management of related service artifacts throughout the whole service lifecycle are neglected. Also the implementation of a lifecycle-encompassing information management infrastructure is not addressed adequately in research and industry. In this paper we introduce different stakeholder roles and their information requirements as well as their influence on services within the lifecycle. Furthermore, this paper proposes a common service management information model (coSIM) that builds a foundation for the management of services and service infrastructures during design-, run- , and change-time.

Keywords: SOA, Service Management, Information Model, Governance, SOA Dimensions

1 Introduction

Service-oriented Computing (SOC) is a promising approach to face the increasing demand for business-aligned applications that provide the ability to react quickly to new requirements of continuously changing business environments. SOA is a design paradigm for IT systems that evolved from distributed computing and offers a way of designing a software system that provides service-based functionalities to end-users, applications, or other service systems [PTDL+06]. Today, an increasing amount of business applications are built as composed service applications [RBHS07]. Enterprises identified the tremendous advantage of SOC and continuously transform their applications and resources to services. However, since more IT-infrastructures follow the SOC paradigm, the number of services that are deployed and utilized, either from internal or external sources increases dramatically. Thus, management solutions that control available services and their service lifecycles and assist enterprises in managing the complexity of a service infrastructure, which grows exponentially with the number and type of services deployed, are needed. Especially in service infrastructures that span multiple enterprises, service providers are confronted with arising complexity in ensuring requirements such as security, policy, and governance additionally to other crosscutting business concerns such as manageability, scalability, and dependability [ADEF+07]. Enterprises will soon require solutions and infrastructure tools that fully support the management of SOC.

The remainder of this paper is structured as follows. Section 2 introduces the challenges in service management whilst section 3 outlines the dimensions that need to be taken into account for a common service management information model (coSIM). Section 4 describes the structure of the proposed coSIM. In section 5 we discuss related work and open issues. The following section 6 presents our summary and outlook to future work.

2 Challenges in Service Management

The problems which drive the management of service infrastructures are manifold. Infrastructure systems evolve from silos of technologies supporting only discrete applications and specific business processes to distributed service resources used in service compositions and various business processes. Infrastructure management must shift from focusing on the support of single applications, for specific user groups or business functions, to the support of complex infrastructure systems and service compositions supporting the needs of multiple groups utilizing service resources within different business functions across the whole enterprise.

The capability to encapsulate business functionality into separate computational services has led to a new way of how IT is used for business integration. The possibility to compose coarse-grained business functionality from finer grained service components is utilized to link different business units within an enterprise. On the down side, the overall management of distributed components, including their orchestration and tracing, as well as the integration of suppliers, partners, and customers through services poses new problems, including the need to handle a higher-level of infrastructural complexity and harder to maintain infrastructures [VS08]. Therefore a common information grounding providing specific, stakeholder tailored views of different aspects of the service infrastructure is a key requirement for an enterprise-scale SOA. Another challenge in service infrastructure management is the variable, hardly-predictable workload for service applications. One of the benefits of SOC is flexibility in supporting new business requirements by composing services to new business functions. Therefore, workload of a service is not predictable at the beginning of its lifecycle.

More and more services are exposed to customers, suppliers, and partners outside the company, leading to nearly unpredictable and traceable service utilization. To ensure the service availability recurring provisioning, configuration and optimization tasks need to be fulfilled, which require a complete understanding of the services provision requirements and dependencies. Such kind of service provisioning specific information is based and influenced on different decisions of various stakeholders during the service lifecycle and can only be provided by a service related coSIM. SOC forces infrastructure management to evolve towards service management. New paradigms need to be introduced within a service management solution, which turn management itself to a service that utilizes all available service related information to control and manage other services. One of the first steps to handle the increasing complexity of service management task, information and governance requirements as well as acting stakeholders will be to identify relevant information dimensions, analyze their associated requirements, information structure, artifacts, and dependencies. A second step will be to support the requirements and characteristics of these dimensions in a coSIM.

3 Dimensions of Service Management

This section outlines different aspects of a service computing infrastructure that need to be taken into account in developing a service information model. The following dimensions as well as the coSIM focus information model requirements only from a service provider perspective. Although service providers are generally supposed to act as service consumers too, this paper firstly aims on supporting the service lifecycle management issues on the provider side as service providers are due to meet QoS requirements issued by their consumers.

3.1 Service Lifecycle

A typical lifecycle of a service can be divided into the following different stages [SG05].

Idea is the first phase which indicates the general business-driven intention to create a new service which provides business functionality. Although, at this stage, service requirements are yet undefined, some information and management functions are already needed. These are the initial registration of a service entry in a service repository and the setup of a service information space to store all relevant service information and artifacts (created during the service lifecycle). During the service registration information like a rudimentary description of the planned service, service responsibilities, and information about the current service lifecycle stage are stored [SSOA07].

Creation, as the second stage, involves service definition, design, implementation, and testing tasks. It also includes the discovery and incorporation of already deployed and available services as one of the major advantages of SOA.

In the **deployment phase**, the service is introduced into its target environment. This involves acquiring of system resources and the provisioning of the service instance and all required components. From a management and service information perspective the relevant tasks at this stage are, e. g. the update of the lifecycle state and all service specific registry metadata [SSOA07], the subscription as new service consumer to all utilized operational consumed or supporting services as well as the setup of the state monitoring of this service instance during operation.

The **operational phase** of the service lifecycle is the main stage. The service is in use and provides its capabilities to different kinds of consumers like other applications or services. In the operational phase of the service lifecycle, a wide range of service management information are required and used to ensure the service functionality and operation. These include service consumer related information requirements like service description, service behavior and performance as well as service provider related information requirements like business activity monitoring metrics, service level definition coverage, or failure rate.

Service upgrades or changes are done within the **maintenance stage**, especially those tasks which cannot be carried out in operation stage. Service relevant tasks and information aspects are aligned with (re)configuration functionalities and resulting meta-information updates to the service registry.

The last stages at the end of the service lifecycle are **phase-out** and **archiving**. Both of them create service related artifacts and update existing service related information to ensure, that a service is not longer promoted through service registries nor intended to be used in available applications. In the achieved stage, to which services finally transit, the service provider should promote the information regarding the service retirement and available service substitutes.

This short section outlines that the maintenance of service lifecycles in a SOA can become very challenging and extensive, especially in complex and highly distributed service infrastructures. To handle this complexity a strong information and governance infrastructure with a coSIM needs to be established [SSOA07].

3.2 Stakeholder and Artifacts

A significant challenge in the establishment of a SOA in the area of Business/IT alignment is to deliver true value to the enterprises requirements. To achieve this, a widespread control and governance system needs to be introduced. SOA governance is such an approach that integrates an organization's people, processes, information, assets and artifacts [Bi08]. It ensures the creation, communication, maintenance and enforcement and adaptation of policies across the SOA lifecycle of design time, runtime and change time as well as measuring the overall effectiveness, SOA Governance mitigates many of the business risks inherent in SOA adoption [Er07]. Tasks of SOA Governance are e.g. the management of the service portfolio, documentation of connections and dependencies, planning for a continuous improvement and gradual modernization of the overall service infrastructure and landscape. A major and initially part in the establishment of governance structures is the definition of (SOA-specific) roles and persons in charge. The following roles are proposed as relevant stakeholders that produce or operate on artifacts within a SOA and therefore represent a service management dimension that needs to be taken into account (cf. [Ti07, Du08]).

The owner of the role **domain architect** (DA) is responsible for the overall coordination of the service landscape either in a specialized domain or for the whole enterprise-wide SOA. Therefore the DA has to ensure that the work of the individual service engineers (SE) is aligned in a global context in the meaning of coherency, quality and business strategy. The DA is the leading design authority regarding the service portfolio, service granularity, service dependencies, flexibility and service reuse. In this context the DA is the key role to deliver the business value of SOA.

The owner of the role **service engineer** (SE) is responsible for “good design” of a service in its context. This means the SE needs to find the optimal balance between suitability for the functional problem and re-use potential to ensure appropriate levels of reuse for other enterprise service applications. The SE defines the operations and therefore the granularity of a service. The decisions regarding communication styles and mandatory non-functional policies are also resided with the SE.

The **service developer** (SD) role is slightly similar to the well known role of the application developer. Generally this role is responsible for the implementation of the services operations-functionality based on the specifications from the SE and the business service owner (BSO). A reason to encompass the SD in the SOA-roles is twofold. Firstly the SD directly creates and operates on service related artifacts like service and implementation documentation, code documents, or deployment information. Secondly the SD is the “executing unit” of the SE and applies the platform and policies of the platform architect, which assigns an important feedback operation to the SD.

The **platform architect** (PA) defines the technical infrastructure stack and guidelines to make sure that all service implementations are compliant. This includes decisions for the "right" (combination of) technology (e.g. Web services, CORBA, J2EE/JMS [KBS04], REST [Fi00], etc.), supported standards (e.g. WS-I compliance, WS-Policy, WS-Security) as well as aspects of platform and standards maintenance, evolution, and enforcement. Furthermore the DA is responsible for all decisions related to the technical service landscape. For technical services like authorization, transformation, resource provisioning, or metadata management the PA is comparable to the DA.

The **business service owner** (BSO) is the person (or group of people) who will drive the strategy and decisions for the service from a business perspective. The owner of this role will be the final point for any business related issues to be resolved [Jo06]. Role owners are obliged to define the functional requirements for the service implementation and direct and control the service evolution and retirement. They own the functional scope of the service, the Service Level Agreements. The BSO is the information hub for all business related aspects of the services lifecycle and behavior and responsible for augmenting the business value the service delivers as well as the cost model to deliver that value.

The **technical service owner** (TSO) is responsible for all deployed services within an infrastructure. This includes business related (eventually composed) services as well as technical services that support other services in the SOA infrastructure and generally relate not directly to a business function (e.g. authentication, monitoring, transformation, etc.). The TSO is in charge for service related deployment and maintenance tasks, the management of operation level agreements in terms of availability, performance, and security as well as the maintenance of the overall enterprise structure including the technical implementation and enforcement of specifications and policies defined by the PA [Er07].

3.3 Infrastructure Maturity

One of the key learning's from enterprise-SOA projects is that SOA is an individual, evolutionary step-by-step approach to a new computing paradigm and infrastructure organization rather than a piece of software that is installed and provided in an over-night buy and provisioning process [Er07]. Since one accepted the evolutionary characteristic of SOC it is clear that we need metrics to define the actual level of the SOA evolution process of an enterprise. Such metrics a provided by SOA maturity models which deduce different maturity levels based on various technological as well as organizational criteria [RG08]. Important in the context of a coSIM is the fact that maturity models not only allow to detect the current evolution level, they also offer the opportunity to define general set of required SOA-roles, (associated) artifacts, and most important information requirements and governance processes that are necessary but also sufficient for the current level of evolution. The maturity of an enterprise SOA has a deep influence on the way service management should be organized and therefore is an important SOA dimension that should be represented in- and influence the characteristics and structure of an coSIM.

4 Information Model

The following section presents an initial structure of the proposed coSIM. In Fig. 1 we present a simplified view of the set of top-level elements, addressing the most relevant categories of service management artifacts. Each category (*BaseElement*) contains a complex and extensible structure of related elements, which either contain or reference properties and artifacts.

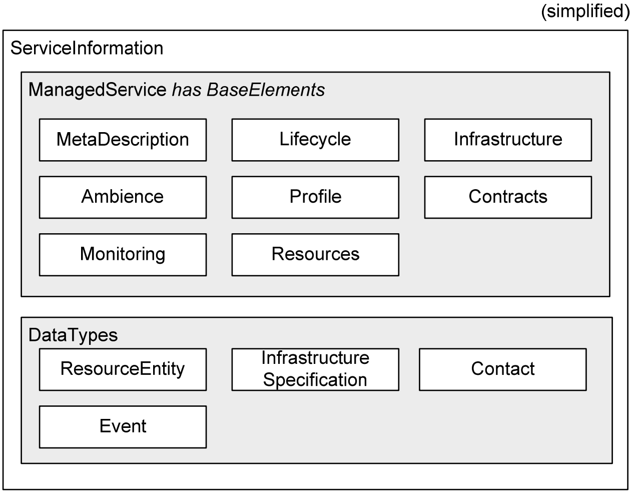

Fig. 1. Top-level elements of the coSIM

The entities of the coSIM structure can be divided into BaseElements which are designated to encompass information elements and are represented either by references to any arbitrary data type or as *DataTypes* defined by the coSIM model.

MetaDescription contains any kind of meta information of the service resource (comparable to the common information in service registries) like e.g. textual descriptions of the service functionality, address information, registry location, business context, associated organizational units, persons in charge (BSO, SD, TSO in stored as *Contact* data type), etc.

Lifecycle contains and archives information regarding the service-lifecycle history, the current lifecycle-stage as well as processed and planned stage transitions.

The *Infrastructure* element contains predominantly information of hard- and software related prerequisites, required technical services as well as deployment specifications (*InfrastructureSpecification* data type). This includes detailed information to version and configuration of used application- and web containers, dependency specification to backend systems (like databases) or other supporting infrastructure components, and any kind of technical condition this service resource needs for its operation.

Ambience is designed to hold information about the current services context with respect to its associated business/technical workflows. Furthermore it contains information about dependencies to and from other services. The primary information sources for ambience are e.g. references to BPEL documents created during the lifecycle-phases creation and deployment as well as monitoring information from the services operation phase.

Profile contains aggregated information that can be classified as key performance indicators on various behavioral aspects of the service. Service profiles are an up-to-date description of non-functional properties of the service. Profiles can be utilized for service comparison, service selection and measuring of business requirement alignment and fulfillment [AKMZ06]. The primary information sources are business activity monitoring (BAM) services within the infrastructure and log files of the service, which combined with specific calculus defined in the profile section of the coSIM, prepare and feed the indicators.

Contract contains instances or references to service level agreements and contracts that define the service related business-driven non-functional requirements. Depending on the architectural style of the service (atomic or composed) either a single or a composed service level agreement resource is referenced and monitored [LF08].

Monitoring primarily contains references to related logging resources of different types. Such types can be e.g. references to structured text document, BAM services or specific management information services that present aggregated historical logging information as outcome of management processes or event correlation systems [MHSV08, Lu01] e.g. as *Event* data type.

The information model element *Resources* is a general store for instances or references (*ResourceEntity* data type) to any kind of resources of the services lifecycle that can be not mapped to any of the above named categories. This includes e.g. different kind of design- and specification documents from the initial service design phases as well as change requests and change documentations from the operational phase.

Artifacts and information elements within the coSIM are represented according to the following rules:

1. All artifacts either if their structure is defined by a schema or any arbitrary binary data are stored in an external, infrastructure-wide reachable information store (e.g. as WS-Resources) and referenced by the coSIM instance.

2. Every information element based on defined DataElement-Types (properties, references, meta information) in the coSIM structure is directly stored in the coSIM instance.

The infrastructure maturity dimension is not directly addressed with the information model. Nevertheless maturity influences the complexity and the required elements which differ on each maturity level. The elements *ambience, profile* and *contracts* for example can be neglected on an early level. Other elements like *lifecycle, infrastructure*, and *monitoring* will be cut to a subset of required properties.

5 Service Management Research

Since the presented approach of a service management infrastructure and information model is based on research of a variety of domains this section presents an overview of the related work in the involved areas.

Papazoglou et al. presented the current state-of-the art and forthcoming challenges in the area of service infrastructure management [PTDL+06]. They clarify that service management is essential in SOC and encompass the control and monitoring of service applications through their complete lifecycle. They propose a management architecture concept and define the grand challenges for forthcoming SOC management infrastructures like self-configuring, self-adapting, and self-optimizing services.

A Web Service Manager introduced by Casati et al. focuses on involving business driven requirements in the management of service infrastructures by creating automatically controlled services driven by actual or predicted values of business metrics that possibly can be represented in the coSIM.

In the area of performance monitoring and anomaly detection Ghezzi et al. [GG07] present a monitor modeling language to define service workflow objects of interest which in combination with a proposed monitoring manager offer possibilities to monitor the execution of BPEL processes in detail and react in defined circumstances.

Berbner et al. [BGRH+05] propose an infrastructure that enhances the SOA concept with management functionalities and presents a service proxy-based monitoring prototype. Their work is concentrated on aspects of service discovery, service rating, composition, and monitoring of SLA requirements. Beside a valuable management approach, a service information model instance for each managed service is not in the scope of their paper. Furthermore, the management of the service lifecycle is not directly addressed.

Sahai et. al. [SG05] discuss required management functionalities for service infrastructures as well as possible tasks and metrics for service management. They propose approaches to align the business context to service management. However, they do not present a distributed solution lifecycle spanning information management solution to support the management of services as well as the business/IT alignment.

An interesting approach in the area of service description, extending the service description with semantic information is presented by Schröpfer et al. [SSOA07]. Their article proposes an OWL-S based service description that extends the regular UDDI service repository which possibly can be utilized in semantic annotations in a forthcoming version of the coSIM.

Müller et al. propose a conceptual framework for SOA management. Their technical-driven work combines a plug-in architecture with the concept of management workflows and primarily addresses the technical implementation of a service management system with management workflows that possibly can utilize the coSIM as source for required management processes according to the current lifecycle stage as well as common information storage for the resulting artifacts [MHSV08].

Rathfelder and Groenda [RG08] introduce a independent SOA Maturity Model (iSOAMM) which describes possible challenges, benefits and risks associated with different SOA maturity levels. It also reflects the implications on organizational structures and governance requirements and therefore provides an interesting source for relevant information requirements, management processes and involved stakeholders the coSIM should support in each maturity level.

6 Summary and Future Work

Current work in industry and research in the area of service management is primarily focused on technical aspects and solutions to support a service management infrastructure but the foundation for such supporting system – a common foundation for relevant information – is neglected. We proposed an initial conceptual approach in establishing a coSIM which can build the foundation to support the governance of a service infrastructure including different dimensions, stakeholder informational requirements and types of artifacts. Our future work on this concept will concentrate on a more detailed description of the information categories and their possible extensions, too. Furthermore we will evalutate technical opportunites for information collection during the lifecycle phases and will investigate possible technical representation alternatives that build the coSIM instances either as a decentralized federated information resource system based on e.g. WS-Resource Framework [CFFF+04, FFGT+04] or alternatively by facilitating existing repository systems to support the described service information model [So08, St07].

References

[ADEF+07] Arrott, M., Demchak, B., Errnagan, V., Farcas, C., Farcas, E., Kriiger, I.H., Menarini, M.: Rich Services: The Integration Piece of the SOA Puzzle. IEEE International Conference on Web Services (ICWS 2007), 2007, pp. 176-183

[AKMZ06] Abramowicz, W., Kaczmarek, M., Kowalkiewicz, M., Zyskowski, D.: Architecture for Service Profiling. Modeling, Design, and Analysis for Service-oriented Architecture Workshop: (mda4soa'06) in conjunction with the 2006 IEEE International Conferences on Services Computing (SCC 2006) and Web Services (ICWS 2006), Vol. 0. IEEE, Chicago, 2006, pp. 121-130

[BGRH+05] Berbner, R., Grollius, T., Repp, N., Heckmann, O., Ortner, E., Steinmetz, R.: An approach for the Management of Service-oriented Architecture (SoA) based Application Systems. Enterprise Modelling and Information Systems Architectures, Proceedings of the Workshop in Klagenfurt. GI, 2005, pp. 208-221

[Bi08] Biske, T.: Implementing SOA Governance, 2008

[CFFF+04] Czajkowski, K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W.: The WS-Resource Framework, 2004

[Du08] Dubray, J.-J.: Establishing a Service Governance Organization. InfoQ.com, 2008

[Er07] Erl, T.: SOA Principles of Service Design. Prentice Hall, 2007

[FFGT+04] Foster, I., Frey, J., Graham, S., Tuecke, S., Czajkowski, K., Ferguson, D., Leymann, F., Nally, M., Sedukhin, I., Snelling, D., Storey, T., Vambenepe, W., Weerawarana, S.: Modeling Stateful Resources with Web Services, 2004

[Fi00] Fielding, R.: Architectural styles and the design of network-based software architectures. 2000

[GG07] Ghezzi, C., Guinea, S.: Run-Time Monitoring in Service-Oriented Architectures. Test and Analysis of Web Services, 2007, pp. 237-264

[Jo06] Jones, S.: Enterprise SOA Adoption Strategies. C4Media Inc., 2006

[KBS04] Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series). Prentice Hall PTR, 2004

[LF08] Ludwig, A.; Franczyk, B.: COSMA - An Approach for Managing SLAs in Composite Services. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.): 6th International Conference on Service Oriented Computing (ICSOC 2008). LNCS, vol. 5364, Springer, Berlin, Heidelberg, 2008, pp. 626–632

[Lu01] Luckham, D.C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co. Inc., 2001

[MHSV08] Mueller, I., Han, J., Schneider, J.-G., Versteeg, S.: A Conceptual Framework for Unified and Comprehensive SOA Management, 6th International Conference on Service Oriented Computing (ICSOC 08), Sydney, 2008

[PTDL+06] Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.J.: Service-Oriented Computing: A Research Roadmap. In: Cubera, F., Krämer B., Papazoglou M. (Ed.): Service Oriented Computing (SOC). Dragstuhl Seminar Proceedings, Dagstuhl 2006, pp. 1-29

[RBHS07] Repp, N., Berbner, R., Heckmann, O., Steinmetz, R.: A Cross-Layer Approach to Performance Monitoring of Web Services. Emerging Web Services Technology, 2007, pp. 21-32

[RG08] Rathfelder, C., Groenda, H.: iSOAMM: An Independent SOA Maturity Model. Distributed Applications and Interoperable Systems, 2008, pp. 1-15

[SG05] Sahai, A., Graupner, S.: Web services in the enterprise: Concepts, standards, solutions, and management. Springer, New York, 2005

[So08] Software AG: The SOALink Cookbook: A guide to extending and integrating with CentraSite, 2008

[SSOA07] Schröpfer, C., Schönherr, M., Offermann, P., Ahrens, M.: A Flexible Approach to Service Management-Related Service Description in SOAs. Emerging Web Services Technology, 2007, pp. 47-64

[St07] Strnadl, C.: Bridging Architectural Boundaries Design and Implementation of a Semantic BPM and SOA Governance Tool. Service-Oriented Computing – ICSOC 2007, 2007, pp. 518-529

[Ti07] Tilkov, S.: Roles in SOA Governance. InfoQ.com, 2007

[VS08] Virili, F., Sorrentino, M.: The enabling role of Web services in information system development practices: a grounded theory study. Information Systems and E-Business Management, 2008