

SOA-GovMM: A Meta Model for a Comprehensive SOA Governance Repository

Jan Königsberger, Stefan Silcher and Bernhard Mitschang
Graduate School of Excellence advanced Manufacturing Engineering
University of Stuttgart
70569 Stuttgart, Germany

Email: {jan.koenigsberger, stefan.silcher, bernhard.mitschang}@ipvs.uni-stuttgart.de

Abstract—In recent years, the paradigm of service-oriented architecture (SOA) has more and more found its way into many organizations. The SOA principles of loosely coupled and reusable services has convinced decision makers in many organizations to start SOA initiatives. Yet, the lack of proper governance mechanisms has doomed many projects to fail. Although some SOA governance frameworks exist, they differ highly in scope and none of them covers the whole spectrum necessary to properly govern a SOA. In this paper we identify and discuss eleven core areas the governance of a SOA has to cover in order to realize the intended benefit in flexibility and agility. We then analyze and evaluate existing SOA governance frameworks with regard to those requirements. Subsequently, we present a meta model composed of four parts: *Service Provider*, *Service Consumer*, *Organizational Structure* and *Business Object*. We show, that those four parts cover all requirements for a comprehensive SOA governance repository. This allows an organization to leverage the information integrated in the repository to better govern their SOA and therefore improve the chances of its success.

Keywords—*Service-Oriented Architecture, SOA Governance, Meta Model, Governance Repository*

I. INTRODUCTION

Increasing globalization and the rising competition as well as mass customization nowadays require companies, especially manufacturing companies, to quickly adapt their IT systems to changing market conditions. The paradigm of service-oriented architectures (SOA), which has been around for quite a while now, is more and more finding its way into many companies, because of the benefits it presents. Establishing a SOA within an organization enables it to improve the flexibility and agility of its business processes and to reuse and adapt them quickly to changing market conditions.

Yet establishing a SOA also introduces new challenges, that is why in recent years, many books, academic literature or blog entries have been produced in order to help new SOA adopters to avoid many of the known mistakes [1], [2], [3].

Still, many SOA initiatives fail or never accomplish the desired improvements in flexibility and agility. A study conducted by Burton Group (now a part of Gartner) back in 2008 comes to the conclusion that only 1 in 5 SOA projects actually succeed [4], a problem that still exists today [5]. Reasons for those failures are diverse and range from lack of executive support, to forgetting that SOA is an architecture and not an IT project.

SOA never was about web services (they are just one

technology to implement services), but most companies understood it that way and focused their efforts mainly on the technical aspects and ignored the equally important process-related and organizational aspects. Yet, only implementing some web services does not make a SOA, it creates only what is known as JABOWS (Just a bunch of web services, [4]). To truly achieve a SOA, one needs to view and comprehend SOA as a mix of people, processes and technologies [6].

To address all aspects regarding SOA and manage it appropriately, its governance is absolutely essential. As a matter of fact, the lack of governance is often stated as the main reason SOA initiatives fail [7].

There are many definitions of what SOA governance actually is, and most of them differ in focus. To name only a few, Ramakrishnan of Oracle defines it as "the creation and administration of policies for the purpose of influencing and enforcing actions and behaviors that align with business objectives" [8]. Allen & Wilkes define it as "The part of IT governance that refers to the organizational structures, policies and processes that ensure that an organization's SOA efforts sustain and extend the organization's business and IT strategies, and achieve the desired outcomes" [9]. Holley et al. see it as "an extension of IT governance that focuses on the life cycle of services and composite applications in an organization's SOA" [10].

Since relevant definitions don't completely cover all aspects we see as part of SOA governance, we have redefined it as follows:

SOA governance serves to effectively control all technical, strategical, organizational and staff requirements of a service-oriented architecture. It is responsible for the introduction of directives and control mechanisms as well as processes to supervise and enforce them. It also is responsible for the alignment of all activities within a SOA according to a company's goals.

Even though the wording has shifted in recent months from pure SOA to cloud computing and from services to the full range of APIs, the notion of SOA governance is now more important than ever. The SOA paradigm still is valid if services are deployed in the cloud, and with the integration of external APIs into an organization or the provisioning of APIs to the outside world, the governance of those APIs is equally important. Regarding this development, Gartner

recently coined the term *Application Services Governance* [11], which can be seen as the evolution of SOA governance. Since *Application Services Governance* currently is even more unclearly defined than SOA governance, and this paper is focused on SOA governance within organizations, we will stick with the term *SOA governance* for the remainder of this paper.

The next section of this paper will discuss the aspects a governance framework for SOA needs to cover, to enable an organization to achieve the benefits SOA offers. Section III will present several existing SOA governance frameworks and analyze them with regard to the requirements from Section II. Section IV will present our meta model (dubbed SOA-GovMM) for a comprehensive SOA governance approach that represents the structured foundation for our SOA governance repository discussed in Section V. The paper concludes with an outlook and plans for future work regarding the realization of the governance repository.

II. REQUIREMENTS FOR A SOA GOVERNANCE FRAMEWORK

Without SOA governance, several problems arise when trying to implement a SOA. Based on those problems, we identify the following eleven key requirements for SOA governance frameworks, both from an extensive literature review as well as from application experience:

- Service Life Cycle Management
- Consumer Management
- Meta Data Management
- Organizational Structure
- Portfolio Management
- Architectural Standards
- Governance Hierarchy
- Funding Model
- Service Monitoring
- Maturity Measurement
- Business Object Management

These requirements will be discussed in detail in the following subsections. There are, of course, further aspects that have to be considered when implementing SOA governance, such as risk assessments, domain management or training of employees. Since they are part of larger programs within an organization and not specific to SOA, we don't see them as key requirements and therefore did not include them in our analysis.

A. Service Life Cycle Management

Service life cycle management is one of the most important things you need to do when governing a SOA. Keeping track of a service might seem simple at first, but gets complicated the more services you add to your portfolio or when you need to keep multiple versions of the same service active to stay backwards-compatible with your consumers.

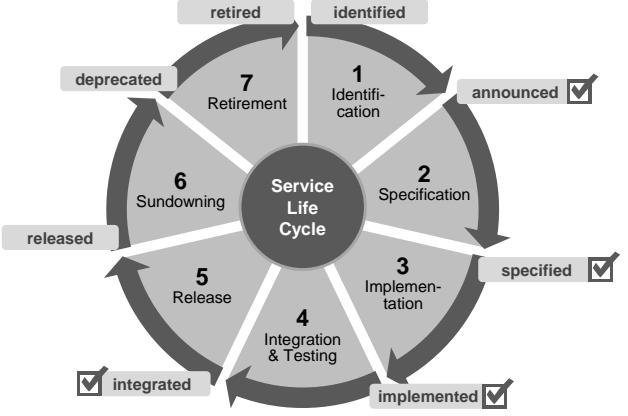


Fig. 1. Service Life Cycle with Service States and Checkpoints

The management of a service's life cycle gets even more complicated, when factoring in, that companies nowadays often use more than one environment, which may hold different versions of the same service or hold the same version of a service in different life cycle states. So, while a service might be in the state *announced* on the production environment, it might at the same time be in the state *released* on the staging environment and in the state *deprecated* on the test environment.

While you could manage the services on each environment separately, managing them together allows you to leverage a lot more information. For example, you could introduce restrictions so a service can't be released on production if it hasn't been released on the staging environment yet.

A service life cycle usually consists of several phases from its initial identification to its final retirement. We propose a life cycle consisting of seven phases, as show in Fig. 1: In Phase 1 (*Service Identification*), the requirement for a new service or service version is identified and then checked against the organizations business portfolio, to determine whether it is aligned with business need. During Phase 2 (*Specification*) the business service is specified according to the existing architecture policies. After passing an architecture review checkpoint, the service will be implemented (Phase 3, *Implementation*). After implementation, the service will undergo integration testing during Phase 4 (*Integration & Testing*). After passing the integration test, the service will be released into production. When the service is no longer needed or when a new version is being developed, the service will enter Phase 6 (*Sundowning*). During this phase all consumers are informed that the service is deprecated and is going to be retired. After the sundown period, the service is retired in the last phase of its life cycle (*Retirement*).

B. Consumer Management

Before a consumer can use a service, it has to negotiate a formal contract with the service provider. This might be accomplished using a standardized contract or by negotiating a new contract for each consumer. When providing standardized contracts, this process could be automated.

This contract (or service level agreement) defines the conditions under which the consumer may use the service as

well as the conditions under which the provider operates the service. This may include payment models (e.g. flat fee or usage-based), service usage constraints (e.g. average service calls per hour) or service response times.

Consumer Management also includes the Management of the consumers themselves, i.e. the tracking of all consumers using services. Those consumers stakeholders have to be managed as well, so that they can, for example, be notified if a service is being replaced by a newer version and therefore being retired. This also includes consumers outside of the organization. For example, if services have been deployed in the cloud and provisioned by business partners.

C. Meta Data Management

SOA governance not only needs to keep track of all services and their life cycle state, but also of a number of other artifacts. Those include first and foremost all documents describing a service interface (i.e. design document, test documentation, interface descriptions as well as data models), the people responsible for the service (service owner, test coordinator, etc.) and also information about service consumers and contract details for every consumer.

While some data might be available as unstructured data (like design documentation) only, most of the data should be held as structured data in a central repository to allow tools to leverage the information as well as automate certain tasks, such as service monitoring and policy enforcement. OASIS' SOA Repository Artifact Model & Protocol (S-RAMP) provides a standardized data model for the management of SOA artifacts [12].

D. Organizational Structure

As already stated previously, SOA is a mix of people, processes and technologies. A successful SOA adoption inherently brings changes in established team and role structures [22], [15].

SOA will bring new processes and services that span across different departments or even the whole enterprise. This moves the focus away from system-specific thinking to a more global view and is bound to lead to intra-organizational tension. It needs to be met with clearly defined roles and responsibilities as well as the promotion of the understanding what SOA and its benefits are on all levels of the organization.

Especially a clear definition of those responsibilities is important, as each stakeholder needs to know who owns a service and is responsible, for example, in case of any outages. To accomplish this, existing roles might be enhanced with new responsibilities or completely new roles have to be introduced. This extends also to boards, as sometimes a council of people might be more suited to address certain issues.

To be able to realize the establishment of new roles, boards and changed responsibilities, support from the (upper) management is extremely important. The lack of sponsorship from (upper) management is as a matter of fact often stated as one major reason for SOA initiatives to fail [20], [23], [24], [25].

Organizational changes are one of the hardest tasks during a SOA initiative, because of people's resistance to change [26].

It is therefore important to clearly outline to each and every employee the benefits the organization as well as the employees themselves gain from this changes. The establishment of virtual teams might help to ease the transition.

E. Portfolio Management

To prevent the "wrong" services from being developed, portfolio management makes sure, that service development is aligned with an organizations business strategy and need.

When a new service is being developed, portfolio management also ensures that there are no similar services already in development. It also includes other stakeholders that might benefit from the service to improve reusability.

This can be achieved by implementing checkpoints in the established project quality processes that trigger a review of newly proposed services. Checkpoints can also be tied to funds for the development of new services that are only cleared after a checkpoint has been passed (see Section II-H).

F. Architectural Standards

The establishment of architectural standards helps service architects and developers in the development of better services. This is usually done as part of Enterprise Architecture Management (EAM) and might include the definition of SOA patterns that require the use of specific middleware, such as a central Enterprise Service Bus (ESB) as well as guidelines for service interface, e.g. standardized authentication methods, business objects or best practice catalogs for service development.

Architectural standards also go hand in hand with establishing (or extending) architecture boards, who are tasked to review new or changed services.

G. Governance Hierarchy

SOA governance is not a stand-alone concept, but has to be integrated with other governance undertakings within an organization, namely IT- and corporate governance.

Literature differs on how SOA governance is linked with them: Rieger & Bruns see it as a subset of IT governance [15], whereas Allen and Wilkes as well as Oracle, see it as being directly influenced by both corporate and IT governance [9], [16].

Because SOA crosses organizational boundaries and has a huge impact on the business itself, it cannot be seen as just a subset of IT governance, but must be seen in a broader perspective, directly being influenced by (and influencing) both IT and corporate governance.

H. Funding Model

With services and processes spanning across several business units, new funding models have to be explored, too. This could mean the implementation of usage-based payment models or flat fee models, allowing for unlimited service use. This helps to distribute the costs evenly between all service consumers.

In early stages of SOA initiatives, this often is a problem, because costs for the development of a service are billed to a

TABLE I. COMPARISON OF SOA GOVERNANCE FRAMEWORKS

	Kohnke et al. [13]	Niemann et al. [14]	Rieger & Bruns [15]	Oracle [8], [16], [17]	HP/Systinet [18], [19]	IBM [10], [20]	Everware -CBDI [9]	PwC [6]	Open Group [21]
Service Life Cycle Management	●	●	●	●	●	●	●	●	●
Consumer Management	○	●	○	●	●	●	●	○	●
Meta Data Management	○	●	●	●	●	○	○	○	●
Organizational Structure	●	●	●	●	○	●	●	●	●
Portfolio Management	●	○	○	●	●	●	●	●	●
Architectural Standards	●	○	○	●	○	●	○	●	●
Governance Hierarchy	●	○	●	●	●	●	○	○	●
Funding Model	●	●	○	●	○	●	●	●	●
Service Monitoring	●	●	○	●	○	●	○	○	●
Maturity Measurement	○	●	○	●	○	●	●	○	●
Business Object Management	○	○	●	○	○	○	○	○	○

● Detailed Concept

● Defined

○ Mentioned

○ Not mentioned

single department or stakeholder, and the subsequent operation and change costs stay with them, even if more consumers start using the service [6].

I. Service Monitoring

Implementing monitoring mechanisms to keep track of services during their usage is another important requirement of SOA governance. These mechanisms help to keep track of which consumers use which services, how often they use it and if they abide by the contracts negotiated (policy enforcement). This is important, because it might impact the fee consumers have to pay (if on a usage-based model), and also could impact the response times of services towards other consumers, if they are being flooded with too many requests.

Another crucial monitoring task is the observation of the services running state. Because services are integrated deeply into an organization's business process, the outage of a seemingly unimportant service might have a huge impact.

J. Maturity Measurement

A SOA program cannot be installed once and then works for all times. Instead it is a process, that has to be checked upon regularly and, if necessary, has to be adjusted accordingly to changes within the organization.

A maturity model helps an organization to assess its current SOA maturity as well as show a path to improve it, by providing a roadmap and best practices.

The same of course holds true for SOA governance, it has to be checked upon regularly as well and if necessary, be adjusted.

K. Business Object Management

One of the advantages of SOA is the abstraction of functionality into services as well as the standardization of these services' interface.

Business Objects further abstract and standardize these service interfaces by decoupling the data model from a service

interface, and managing it separately. The data models are then subjected to their own life cycle, similar to the service life cycle presented above (cf. Section II-A, Fig. 1), which transforms them into business objects.

This allows the development of organization-wide standardized business objects that are specific to certain domains. They can then be used by multiple services, and therefore further improve information integration and reuse. To manage the business objects, they can be subjected as architectural patterns to an organization's enterprise architecture management (cf. Section II-F).

The introduction of Business Objects further increases the agility of service providers, allowing them to adapt their data model for new consumers, while at the same time keeping those changes transparent to existing consumers due to the added abstraction.

III. ANALYSIS OF SOA GOVERNANCE FRAMEWORKS

Multiple SOA governance frameworks have been developed and published. Between the frameworks there are huge differences regarding their scope and capabilities. Most of them focus on single aspects, while ignoring others, whereas few have a broad view of the topic. The following section gives an overview over those frameworks and subsequently analyses them on the basis of the requirements presented in Section II.

A. Existing Frameworks

The following SOA governance frameworks have been published in magazines or journals (1 - 3), were developed by software companies (4 - 6), published by consulting companies (7 - 8) or industry consortia (9).

1) *Kohnke et al. [13]*: Kohnke, Scheffler and Hock, all employed by SAP, present a SOA governance framework aligned along several 'fields of action', grouped into the three major areas 'structures', 'processes' and 'employees'.

2) *Niemann et al. [14]*: Niemann et al. describe a governance model consisting of two parts, the 'SOA Governance Control Cycle' and the 'SOA Governance Operational Model'. The control cycle is built up of four phases describing a process to develop and improve an organization's SOA governance. The operational model consists of six 'main elements' that are targeted to achieve certain predefined SOA goals.

3) *Rieger & Bruns [15]*: Rieger & Bruns provide a governance framework by highlighting several key aspects of SOA governance. They also develop a detailed role model and an overview on how those roles have influence on meta data management.

4) *Oracle [8], [16], [17]*: Authors at Oracle define a governance framework consisting of a 'SOA Governance Reference Model', grouped into five 'key areas' and a 'SOA Governance Continuous Improvement Loop' providing a roadmap to keep SOA governance active and on track. They also provide a SOA Maturity Model detailing the areas an organization needs to leverage in order to achieve SOA success.

5) *HP/Systinet [18], [19]*: HP presents a SOA governance framework structured into the four key aspects 'SOA policies', 'service contracts', 'lifecycle management' and 'metadata'. They also present a list of eight best practices.

6) *IBM [10], [20]*: IBM define their SOA governance framework as an extension of IT governance with focus on the service life cycle. The framework describes a service life cycle and a service governance life cycle, both consisting of four phases. They also list a catalog of best practices and questions that need to be answered when addressing SOA governance.

7) *Everware-CBDI [9]*: Allen & Wilkes present Everware-CBDIs take on SOA governance by outlining a framework consisting of four dimensions (what, how, who, when) around a central SOA policy hierarchy.

8) *PricewaterhouseCoopers [6]*: PricewaterhouseCoopers (PwC) propose a SOA governance model based on the three dimensions 'people', 'process' and 'technology' that are split into different governance categories. They also list several *key governance policies* as well as outline a four-step process to implement SOA governance within an organization.

9) *Open Group [21]*: The Open Group provides a comprehensive SOA governance framework, consisting of two parts: The SOA Governance Reference Model (SGRM) and the SOA Governance Vitality Method (SGVM). The reference model describes multiple 'Governance Guiding Principles' as well as processes that need to be considered for governance. The vitality method describes a process following four phases to monitor and improve SOA governance.

B. Analysis

As already stated above, the frameworks presented in the last section all differ in scope. Most of them focus on a few select aspects (e.g. Rieger & Bruns and HP/Systinet), for which they present a more detailed concept. Beyond that, they do not provide a complete view on SOA governance. This holds especially true for the frameworks developed by software companies. Their definition of SOA governance is only based on the scope of their software portfolio.

There are a few aspects nearly all of the frameworks agree upon. This is of course the general need for SOA governance, as well as the aspects *Service Life Cycle Management* and *Organizational Structures*. Some of the frameworks considering the organizational structures also provide a detailed role concept (e.g. Rieger & Bruns).

Among the least regarded aspects is *Service Monitoring*, which is defined by less than half the concepts. An aspect that all frameworks except one ([15]) ignore is the management of business objects, which is, as discussed above (cf. Section II-K), very important to reach business agility and stable interfaces.

A full analysis and comparison of the existing frameworks is shown in Table I. The table shows whether the framework neglects a requirement (○), simply mentions it (◐), defines it (◑) or contains a detailed concept (●).

To properly govern a SOA, we propose a comprehensive SOA governance repository, that covers all requirements presented in the previous section. It should especially cover areas of SOA governance, that are currently untouched by existing frameworks, such as the service life cycle management within a multi-environment setting and the management of business objects and their life cycle.

The notion of a central SOA registry or repository is already present in some frameworks (e.g. Everware/CBDI or Oracle). In case of repositories this is always limited to the management of service artifacts, whereas registries deal with the management and discovery of the services themselves. There are also several other tools from different vendors labeled as "SOA Governance Tools". Unfortunately they are either lacking required functionality, or the functionality is split between different tools that can't be integrated or overlap [27]. Splitting functionality up, deprives the user of leveraging the information contained in those systems. E.g., it is not possible to automatically inform consumer stakeholders after a new service version has been announced, if the services and the stakeholders are managed in different tools.

A central repository also has the advantage, that information has only to be managed in one place and users only have to learn how to use one tool instead of multiple, increasing the tools' adoption rates.

IV. META MODEL FOR A SOA GOVERNANCE REPOSITORY

The framework analysis above shows that existing governance frameworks only regard parts of the complete SOA governance picture. Therefore a complete view of all aspects is necessary.

The following section presents the SOA Governance Meta Model (SOA-GovMM) that defines our comprehensive SOA governance repository. It is structured into four parts: *Service Provider*, *Service Consumer*, *Organizational Structure* and *Business Object Management* (cf. Fig. 2).

A. Service Provider

Part A of the SOA-GovMM models the service provider, including the service itself, its life cycle and its deployment. It is shown in Fig. 3.

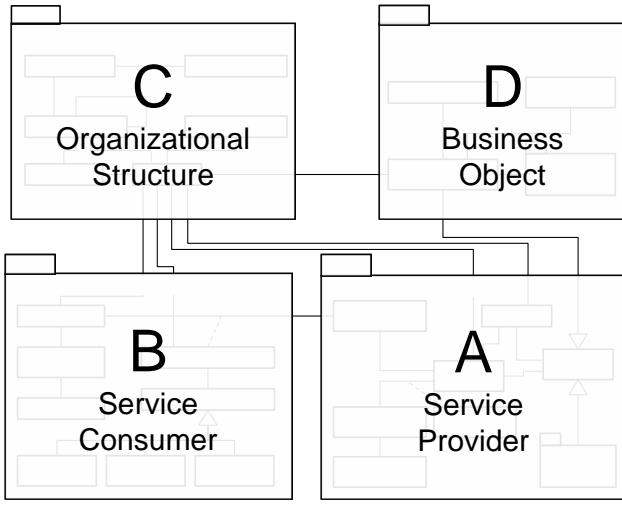


Fig. 2. Parts of the SOA-GovMM

As already stated in Section II-A, the management of a services' life cycles is one of the most challenging tasks within SOA governance. The model takes care of the fact, that a service may exist in multiple versions at the same time, by allowing each *Service* to have multiple *Service Versions*. A *Service Version* can be assigned to multiple *Environments* to model the parallel deployment of the same service version on different environments as described in Section II-A. Since the service version deployed does not necessarily have the same state in each environment, each *Service Version/Consumer*-pair has an associated *Life Cycle State*. Each *Environment* is also assigned *Operating Resources*. They represent, for example, the physical server hardware.

Service meta data, as introduced in Section II-C, is modeled as a collection of *Service Artifacts*, that can be assigned to either the *Service* (and are then valid for all *Service Versions*) or directly to a single *Service Version*. There may exist specialized *Service Artifacts* such as the exemplary modeled *Service Description* (e. g. the services' WSDL) or the *Business Objects* assigned to this particular *Service (Version)*.

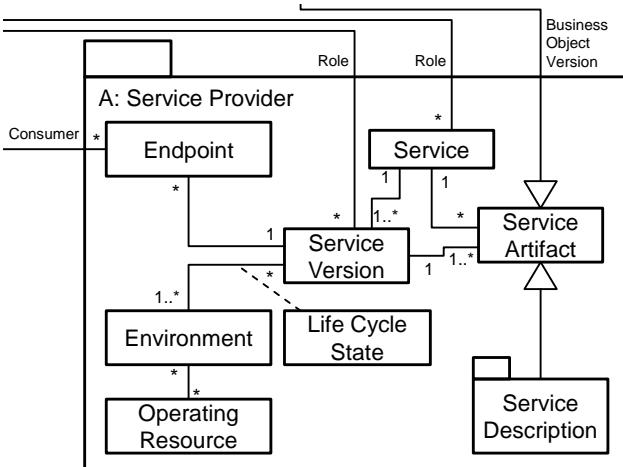


Fig. 3. SOA-GovMM Part A: Service Provider

Each *Service Version* also has one or more *Endpoints* associated, that represent the connection point to the *Service Consumer* (cf. Section IV-B).

Services as well as *Service Versions* are linked to one or multiple *Roles* (cf. Section IV-C).

B. Service Consumer

Part B of the SOA-GovMM describes the service consumer. It is shown in Fig. 4.

Each service provider's *Endpoint* is connected to a *Consumer*. The *Consumer* represents the link between a specific consumer system and a service provider. The *System Version* and *System* represent this consumer system (respective system version) and help to enable better portfolio management (cf. Section II-E) by making visible which consumer system uses which services.

Each *Consumer/Endpoint*-pair is also associated with a *Consumer Contract* detailing this pairs' consumer contract (cf. Section II-B). The details of this contract are modeled as generic *Contract Properties*. Those might be specialized as the (exemplary) modeled *Contract Document*, the *Quality of Service (QoS) Property* or the *Funding Property*, which holds the payment/funding information as introduced in Section II-H. The contract details stored in these properties are used as a basis for service monitoring and policy enforcement (cf. Section II-I). For example: The negotiated number of maximum service calls per day can be checked against the actual service calls. If this shows that the consumer is overusing its quota, appropriate steps can be taken to either enforce the policy or re-negotiate the contract with the consumer.

The *Contract*, as well as the *Consumer* are linked to the organizational model (cf. Section IV-C) via one or multiple *Roles*. Those might be the owner of a system or the contact person that negotiated a specific service contract.

C. Organizational Structure

As already stated, a SOA is highly dependent on clearly defined organizational structures and responsibilities (cf. Section II-D). This is modeled in part C of our meta model and shown in Fig. 5.

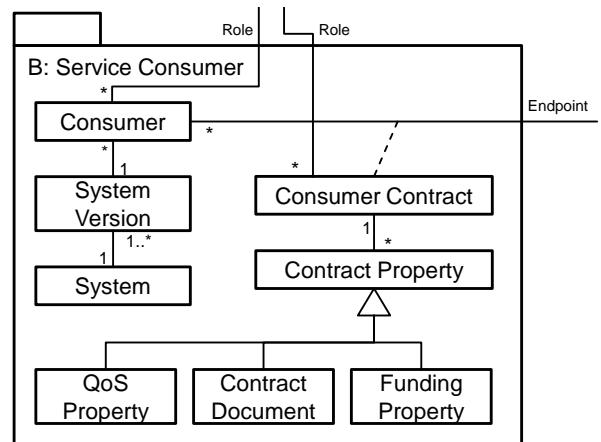


Fig. 4. SOA-GovMM Part B: Service Consumer

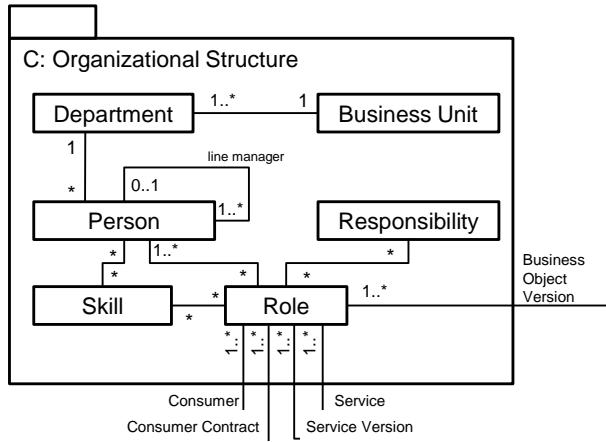


Fig. 5. SOA-GovMM Part C: Organizational Structure

Core element of this part is the *Role* entity. It is the link into the organizational structure for the previously described *Service* and *Service Version* (cf. Section IV-A), as well as the *Contract* and *Consumer* (cf. Section IV-B).

Each *Role* is associated with certain *Responsibilities*, the bearer of this role holds. A *Role* is held by a certain *Person* that in turn is part of an organizations' *Department*. To model the organizations hierarchical structure, a *Person* is assigned a line manager (who again is a *Person*).

Both *Person* and *Role* are associated with certain *Skills*. In case of a *Role*, those represent the skills necessary to fill this particular role. In case of a *Person*, those represent the skills this person has to offer. This construct can be used to identify qualified candidates to fill a role.

D. Business Object

Part D of the SOA-GovMM models the business objects as introduced in Section II-K. The model is shown in Fig. 6.

As already stated, business objects are subject to their own life cycle and are therefore modeled similar to services (cf. Section II-K). As there might exist several versions of a business object, each *Business Object* can have multiple specialized *Business Object Versions* assigned to it. *Business Object Versions* are linked as *Service Artifacts* to a *Service Version* (cf. Section IV-A). To model its life cycle, each *Business Object Version* is furthermore connected to its respective *Life Cycle State*.

The entity *Data Model* represents the specific data model of a *Business Object Version*. This might be an XML-based (or otherwise represented) description of the actual data model.

V. ASSESSMENT OF THE SOA GOVERNANCE META MODEL

To show how the meta model presented covers our SOA governance requirements (cf. Section II), this section provides an assessment of the model. It also outlines our first ideas on how to realize our SOA governance repository.

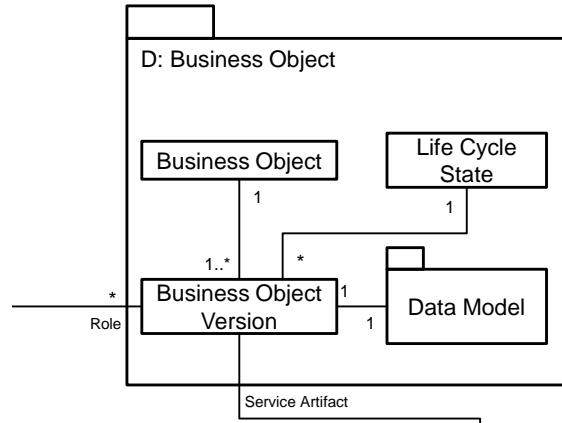


Fig. 6. SOA-GovMM Part D: Business Object

A. Assessment

As stated in Section III-B, current SOA governance frameworks do not cover the life cycle management of services that are deployed within a multi-environment setting. To support this, our SOA governance meta model decouples services into an arbitrary number of service versions, each subject to their own life cycle in relation to different environments.

Another requirement that current frameworks don't consider are business objects. This is, as Section IV-D and Fig. 6 show, explicitly modeled within our meta model.

The model furthermore clearly fulfills the aspects required by meta data management, organizational structure, consumer management and funding model.

Some aspects are not as easily recognizable from the distinct parts of the model, instead they can only be leveraged when considering the complete model. This is for example true for portfolio management: The model allows to get a complete overview over all consumers and providers (including their versions), as well as their interfaces, design documentation and life cycle. This enormously simplifies portfolio management, as the complete service portfolio is available within one model.

The model also provides the overview over service descriptions and artifacts as well as business objects, which are all required for architectural standards as introduced above. Furthermore, the model can be used to assess an organizations SOA maturity (cf. Section II-J) by allowing to produce statistics about service reuse and interface standardization.

A requirement that currently is not covered completely by the model is service monitoring. The model already contains the complete consumer contracts, that build the basis for monitoring. To actually be able to use this data for service monitoring it needs to be compared to actual service usage data. This usage data is not yet included in the model as it depends mostly on the actual technical infrastructure used.

B. The SOA Governance Meta Model as semantic foundation for the SOA Governance Repository

Our SOA-GovMM is, as previously stated, the foundation for a comprehensive SOA governance repository. The repository will build on and implement the meta model, as well as provide the operations necessary to manage data stored in it.

C. Realization of the SOA Governance Repository

The straight-forward approach to realize the governance repository is to exploit any off-the-shelf Database Management System, install the meta model as database schema and to provide the necessary functionality by means of an appropriate database API.

However, alternate approaches seem to be more beneficial. A flexible and extendable solution based on the *Resource Description Framework (RDF)* [28] seems especially promising. One main advantage of this approach is the flexibility of a RDF data model. Since it does not have a fixed schema as relational databases do, the underlying data model can be changed as needed, even at run time. For example: This allows to define new specialized properties for services or contracts (cf. Sections IV-A and IV-B) as they are needed. As RDF is a graph per definition, the modeling of complex data and its dependencies is also much easier in comparison to traditional approaches.

As the governance repository will be covering a huge functional range for several different roles a well-thought-out software concept and user-guidance are indispensable.

VI. CONCLUSION AND OUTLOOK

In this paper we proposed a meta model for a comprehensive SOA governance framework covering eleven core requirements we identified both from literature review and our own application experience. The analysis and evaluation of existing governance frameworks yields the result that they are not able to cover our complete requirements catalog. This supports our notion that a comprehensive SOA governance repository is necessary for an organization to be able to fully leverage SOA benefits. We show, that our proposed meta model is able to cover all of our requirements and therefore is suitable to be used as foundation for the implementation of a comprehensive SOA governance repository. Using such a repository will enable organizations to quickly adapt their IT infrastructure to changing market conditions and turbulences.

We currently work on the realization of the SOA governance repository using the RDF-based approach outlined in the paper. Future work will also include the validation of the repository, possibly within an enterprise context.

ACKNOWLEDGMENT

The authors would like to thank Martin Schaaf for providing valuable insights into SOA and governance processes within large enterprises.

REFERENCES

- [1] T. Erl, *Service-oriented architecture: Concepts, technology and design*. Upper Saddle River [etc.]: Prentice Hall PTR, 2005.
- [2] N. Bieberstein, *Service-oriented architecture (SOA) compass: Business value, planning, and enterprise roadmap*, ser. Developer works. Upper Saddle River and NJ [u.a.]: IBM Press, 2006.
- [3] T. Erl, *SOA: Principles of service design*, ser. The Prentice Hall service-oriented computing series from Thomas Erl. Upper Saddle River and NJ: Prentice Hall, 2008.
- [4] M. Meehan, “SOA adoption marked by broad failure and wild success,” 2008. [Online]. Available: <http://searchsoa.techtarget.com/news/1319609/SOA-adoption-marked-by-broad-failure-and-wild-success> [Accessed: 19.04.2014]
- [5] P. Swabey, “Divide & Conquer: The science of splitting large IT projects into manageable chunks,” *InformationAge magazine*, vol. June, 2012.
- [6] PriceWaterhouseCoopers, “SOA Governance: More than just registries and repositories,” 2009.
- [7] Gartner, “Bad Technical Implementations and Lack of Governance Increase Risks of Failure in SOA Projects,” 2007. [Online]. Available: <http://www.gartner.com/newsroom/id/508397> [Accessed: 23.04.2014]
- [8] M. Ramakrishnan, “SOA Lifecycle Governance: Right from the Start,” 07.2013.
- [9] P. Allen and L. Wilkes, “SOA Governance: Challenge or Opportunity,” *CBDI Journal*, no. 03/2010, 2010.
- [10] K. Holley, J. Palistrant, and S. Graham, “Effective SOA Governance,” 03.2006. [Online]. Available: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/soagov-mgmt.pdf> [Accessed: 19.04.2014]
- [11] P. Malinverno, “Govern Your Services and Manage Your APIs With Application Services Governance,” 15.11.2012.
- [12] OASIS SOA Repository Artifact Model and Protocol (S-RAMP) TC, “S-RAMP Version 1.0. Part 1: Foundation,” 2014. [Online]. Available: <http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/s-ramp-v1.0-part1-foundation.html> [Accessed: 20.04.2014]
- [13] O. Kohnke, T. Scheffler, and C. Hock, “SOA-Governance: Ein Ansatz zum Management serviceorientierter Architekturen,” *Wirtschaftsinformatik*, vol. 8, pp. 408–412, 2008.
- [14] M. Niemann, J. Eckert, N. Repp, and R. Steinmetz, “Towards a Generic Governance Model for Service-oriented Architectures,” in *Proceedings of the Fourteenth Americas Conference on Information Systems*, I. Benbasat and A. R. Montazemi, Eds. Madison: Omnipress, 2008.
- [15] I. Rieger and R. Bruns, “SOA-Governance und -Rollen: Sicherung des Mehrwerts einer Service-orientierten Architektur,” *OBJEKTSPEKTRUM*, 2008.
- [16] S. G. Bennett, “A Framework for SOA Governance: Oracle Practitioner Guide,” 2012-02. [Online]. Available: <http://www.oracle.com/technetwork/topics/entarch/oracle-pg-soa-governance-fmwrk-r3-2-1561703.pdf> [Accessed: 27.02.2014]
- [17] B. Hensle and M. Deb, “SOA Maturity Model - Guiding and Accelerating SOA Success,” 09/2013.
- [18] Systinet, “SOA Governance: Balancing Flexibility and Control Within an SOA,” 09.2006.
- [19] Hewlett Packard Development Company, “The eight most important best practices in SOA governance,” 2008.
- [20] W. A. Brown, G. Moore, and W. Tegan, “SOA Governance: IBM’s approach,” Somers and New York, 08.2006.
- [21] The Open Group, “SOA Governance Framework,” Berkshire, 2009-08. [Online]. Available: <http://www.opengroup.org/soa/source-book/gov/index.htm>
- [22] S. Poi, B. Sleight, J. Viezel, and D. Snavely, “Enabling SOA through organizational change and governance,” 2007.
- [23] Oracle and Jyothi Swaroop, “Get Serious About SOA Governance: A Five-Step Action Plan for Executives,” 2012-09.
- [24] M. Kavis, “Top 10 Reasons Why People are Making SOA Fail,” 2008. [Online]. Available: <http://www.cio.com/article/438413/> [Accessed: 10.04.2014]
- [25] G. Klebanov, “Why Enterprise SOA Fails,” 2014. [Online]. Available: <http://theenterprisearchitecture.com/2014/01/18/why-soa-fails/> [Accessed: 10.04.2014]
- [26] T. Kokko, J. Antikainen, and T. Systä, “Adopting SOA - Experiences from Nine Finnish Organizations,” in *13th European Conference on Software Maintenance and Reengineering*, 2009, R. Ferenc, Ed. Piscataway and NJ: IEEE, 2009, pp. 129–138.
- [27] P. Malinverno, D. C. Plummer, and G. van Huizen, “Magic Quadrant for Application Services Governance,” 2013-08-08.
- [28] World Wide Web Consortium, “RDF 1.1 Concepts and Abstract Syntax,” 2014-02-25. [Online]. Available: <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/> [Accessed: 24.04.2014]