Chapter 16
Supporting USDL by a Governance Framework

Christian Janiesch and Michael Niemann

Abstract The previous chapter introduced service marketplaces as fundamental tool that enables and benefits service ecosystems. The application of service marketplaces for enterprise resource planning is a growing market. The operation of such an online marketplace requires a governance approach that lies adjacent to the requirements of a SOA and the more general governance of IT. It also has requirements of its own, especially when it comes to the description of services with languages such as USDL. In this chapter, we propose four building blocks as a basis for a governance framework that is capable of supporting the operation of a service marketplace. The research is based on existing frameworks and also takes into consideration the particularities of emerging SOA Governance approaches. We emphasize the processes required for the management of service descriptions.

16.1 Introduction

Service marketplaces in the Internet of Services are an approach to enable and facilitate the trading of services. In the case of trading pure software services, the goal is to make software ubiquitously available as services which can be licensed for use. The aim is to reduce hardware cost and maintenance at the customer side and make (complex) software a commodity. The application of marketplaces for enterprise resource planning is a growing market.

Christian Janiesch
Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology, Englerstr. 11, Geb 11.40, 76131 Karlsruhe, Germany,
e-mail: christian.janiesch@kit.edu

Michael Niemann
KOM - Multimedia Communications Lab, Technische Universität Darmstadt, Rundeturmstr. 10, 64283 Darmstadt, Germany, e-mail: michael.niemann@kom.tu-darmstadt.de

A. Barros and D. Oberle (eds.), *Handbook of Service Description: USDL and Its Methods*,
DOI 10.1007/978-1-4614-1864-1_16, © Springer Science+Business Media New York 2012

The operation of an online service marketplace requires a governance approach that lies between the requirements of an SOA and the more general governance of IT. IT Governance aims to reduce the risk of fraud, data inconsistencies, and resulting damages for stakeholders by defining regulations concerning the organizational model (roles and responsibilities) and general procedures (cf. Section 16.2 for an elaborated overview of related work). We use the IT Governance frameworks of COBIT [6] and ITIL [7] as a basis for research not only because they provide insights from unbiased organizations rather than individual enterprises but because they are at both ends of the governance spectrum: COBIT focuses on strategically important tasks (main processes) and ITIL focuses on management tasks (support processes), which are often subject to outsourcing and, thus, the ideal blueprint for managed third-party processes.

Based on an analysis of related work, we propose four building blocks in Section 16.3 to instantiate a governance framework that is capable of supporting the operation of a service marketplace. We base our conceptual considerations on existing frameworks and also take into consideration the particularities of emerging SOA Governance frameworks. We highlight its usefulness and applicability to USDL in Section 16.4. We conclude with a summary and outlook (Section 16.5).

16.2 Related Work

According to a survey conducted among companies that use SOA as enterprise architecture, 79 % of the respondents stated that they feel a large negative risk by taking services into production that are not effectively "governed." On top of that, 88 % of the companies consider their current SOA Governance approach insufficient — only 12% implemented a sufficient approach according to their own estimation [5]. Although companies are aware of the high risk of a governance lack, they have not installed sufficient mechanisms to address it. The need for appropriate governance approaches is high.

In recent years, a number of models and frameworks for SOA Governance have been proposed. While proceeding from diverging challenges and definitions, most of them address similar goals. They propose varying techniques and differing combinations of them to reach these goals. As the awareness of the need for SOA Governance is quite young, only few accepted standard procedures, goals and techniques exist. One reason might be the fact that there is no common definition of SOA Governance that could form a foundation for the different approaches.

We investigated and compared 22 SOA Governance approaches, developed in 35 publications at companies and research institutions. We divided them into three groups: Scientifically published approaches cover reviewed publications such as journal articles, conference papers, as well as book chapters and books. Many governance approaches have been made available by software vendors, published as company whitepapers which target governance for the SOA system aligned with proprietary software products (e.g., SOA infrastructure). The third group is formed by authors from the consulting industry that published their expertise in whitepapers based on achieved experience. During the examination, ten major aspects have been identified.

The approaches, however, show different quality. Approaches which formulate a clear view and opinion backed with arguments concerning a criteria, i.e., whose recommendation of the integration of a corresponding criterion is backed with arguments, are considered a *founded recommendation* in this analysis. Suggested building blocks of SOA Governance are fully integrated, concrete suggestions are made and even examples are given (marked with • in Table 16.1). In contrast to this, some approaches are characterized by a more narrow view of the topic. They show a lack of clear instantiation, explanation, level of detail, or specification. They point out an aspect considered important, however, lack the required level of detail, or precision. These cases are considered as *proposal* of the integration of the given criteria, equally to "partially integrated, mentioned" (marked with ○ in Table 16.1).

Governance Policies

In almost all approaches, governance policies are informally defined as "means to define what's *right*." Generally, governance policies represent general guidelines, conventions, rules, and best practices that support the controllable and efficient operation of the SOA system. They are often applied in the administration of a service lifecycle, or during a SOA procedure model.

Generally, governance policies are considered distinct from *service performance-related policies* as described by standards such as WS-Policy [47, 24]. Main aspects of governance policies are their application to *roles*, *service design and operation*, and *service documentation*. Some approaches, however, leave the specified policies unclassified. Concerning policy handling, procedures for policy exception handling, as well as recognition of too restrictive policies are suggested.

As a consensus of all authors, policies are considered mighty instruments that combine various application aspects. They represent the most important and quite complex aspect of SOA Governance. Application aspects are roles-related, service design and operation-related, and, explicitly, service documentation-related policies. The latter are to ensure useful retrieval processes that are performed by, e.g., service requesters. Important aspects of policy handling are policy lifecycle management, policy exception regulation, and recognition of inappropriate (too restrictive) regulations.

Organizational Structure

Due to the changed conditions of SOA systems compared to other IT systems, the majority of authors considers to adjust organizational structures. The approaches outline and introduce new boards, councils, and institutions for special accountability around SOA.

Table 16.1: Detailed survey results.

	Governance Policy Catalogue	Organizational Structure	Artefacts Management	Role and Responsibilities	Service Lifecycle	Strategic Aspects	SOA Procedure Model	Governance Processes	Metrics	SOA Maturity Measurement
Books and journal articles										
Schepers et al. [44]	●	○	○	○	○	●	●	-	-	●
Bernhard and Seese [17]	●	●	○	-	●	-	●	●	●	-
Derler and Weinreich [23]	○	-	●	○	●	-	-	-	-	-
Kohnke et al. [31]	●	●	-	○	○	●	-	○	○	-
Bieberstein et al. [18, 19]	○	●	●	●	-	●	●	●	○	○
Marks and Bell [32]	●	●	●	○	○	●	●	●	●	-
Brown et al. [22]	●	●	●	●	●	●	●	●	●	-
Schelp and Stutz [43]	○	●	-	○	-	○	○	-	○	-
Rieger and Bruns [42]	●	●	●	●	○	○	-	-	-	-
Josuttis [29]	○	●	○	○	●	○	○	○	-	-
Software Manufacturers										
Brauer and Kline [21]	●	●	●	-	●	○	○	-	-	-
Hewlett Packard [8]	●	●	●	-	●	○	○	-	-	-
Systinet [3]	●	●	●	-	●	○	○	-	-	-
WebMethods [4]	○	-	●	○	○	-	○	○	-	-
Matsumura [33]	○	-	●	○	○	-	○	○	-	-
Software AG [1, 9]	●	○	●	○	●	○	●	-	○	●
BEA Systems Inc. [2]	○	-	●	-	●	-	○	-	-	-
Afshar [13]	●	●	●	○	○	●	●	-	-	●
Holley et al. [26]	-	●	●	○	●	○	●	-	○	-
McBride [34]	-	●	●	○	●	○	●	-	○	-
Mitra [35]	-	●	●	○	●	○	●	-	○	-
Muriankara [37]	-	●	●	○	●	○	●	-	○	-
Woolf [52]	-	●	●	○	●	○	●	-	○	-
The Open Group [11]	○	●	-	●	●	●	●	●	○	-
Consulting Industry										
Everware-CBDI: Allen [14]	●	●	●	○	-	●	-	●	-	●
BearingPoint: Rane and Lomow[41]	●	○	○	○	○	○	●	-	-	-
ZapThink: Bloomberg [20]	●	○	●	●	-	●	-	○	-	-
Windley [50, 49, 51]	●	●	○	○	●	-	●	-	-	-
Berlecon: Quantz [40]	○	●	●	-	-	-	-	-	-	-

All approaches that give *founded recommendations* concerning organizational changes (15 out of 22), recommend setting up a *SOA Centet of Excellence* (SOA CoE). This institution has convinced in theory (7 mentions) as well as in practice (8 mentions). It can be considered a crucial organizational institution for the operation of an SOA system.

Summarizing, the presented organizational entities (SOA CoE, SOA Board, and SOA Governance Board) are the three most frequently integrated ones. Competencies, however, are not clearly attributable. The approaches give different recommendations especially concerning the question of how decision and consulting competencies are to be distributed among the entities. The majority, however, agrees on the SOA CoE bundling many of the discussed competencies — in some cases even all of them.

In contrast to organizational entities that could also be named *group roles*, the precise definition of (single) roles and responsibilities has been a major aspect of SOA Governance approaches.

Roles and Responsibilities

Almost 80 % of the approaches mention the adjustment of roles and responsibilities for the operation of a SOA system. All these authors consider implementing and operating an enterprise architecture as an SOA to have impact on the organizational structure of the entire company. Besides the introduction of new organizational entities, this covers the definition of new roles and accountability. In order to assign clear and non-overlapping definitions of competencies, a solid concept for roles and accountability is commonly considered to be advantageous for all involved persons and the operation of the SOA system.

In the context of SOA Governance, an important aspect is the targeted *impact on behavior*. The IT Governance goal to "achieve desirable behavior in the use of IT" [48] is an important goal for SOA Governance as well.

Methods such as *RACI charts*, *impact on behavior* [19, 32], *SOA Education Plan* [18], and the *Capability Assessment Method* [22] are considered central components of the discipline SOA Governance.

Artifact Management and Software Support

Clearly more than half of the approaches (14 out of 22) name software support or artifact management a central building block of SOA Governance. Most of them come from the software industry.

During the development process of a SOA, many artifacts are created, e.g., *services*, *meta data*, *service descriptions*, *interface descriptions*, and *message format specifications*. Services in operation are bounded by *policies* and *service contracts*. Further meta data are SOA Governance artifacts such as *roadmaps*, *process descriptions*, and *reference architectures* (cf., e.g., [11]). The approaches suggest the operation of a service registry or service repository, and a Web service management system. They recommend structuring all kinds of artifacts as a meta-model to clarify relationships as well as the establishment of additional data and artifact related roles and responsibilities.

As main function of a service registry, most authors refer to publishing and discovery of services, while the service repository is considered to serve meta data storage. However, none of the approaches recommends operating both of these. Nevertheless, the understanding of service registries and repositories in terms of functionalities astonishingly diverge among the authors (cf., e.g., definitions by [41, 50]).

Service Lifecycle

Service lifecycle management (SLM) is a central aspect of SOA Governance. More than 75% of the approaches mention a service lifecycle to be an integral part of SOA Governance. The majority of approaches emphasizing the service lifecycle are from the author group software vendors.

Lifecycle models, in general, are widely used additives for design, development, operation, and maintenance of software (e.g., in [46]). As a purpose of SOA Governance, the design, implementation, operation, and version management of services can be improved by comprehensive and reasonable regulations in service lifecycles [21, 33, 50]. Their planning and implementation is part of SOA governance. However, the notions of definition and distribution of activities in lifecycle phases vary in wide ranges.

Using lifecycles, many artifacts beyond services can be controlled. Additionally to guidelines, applications composed from services, as well as business processes can be controlled by lifecycles (as proposed by [13]). Also, readjustments of SOA goals to changed business requirements, or frequent transposition of the SOA Governance model are performed using lifecycles (cf., e.g., [22, 26]). Using lifecycles is a powerful instrument of control, i.e., a powerful instrument of governance.

Strategic Alignment

The conception of a strategic plan as well as business-IT alignment, are both considered a further central element of SOA Governance by the experts. 15 out of 22 approaches refer to strategic alignment, the majority with concrete suggestions. Especially authors from the practitioner's domains (consulting industry and software vendors) consider this point crucial. Four aspects of strategic alignment considered most important are *formalization of SOA goals*, *identification and prioritization of services*, *adequate financing of service development*, and *SOA commitment of the management*.

SOA Procedure Model

Besides the management and effective administration of governance methods, the strategy and procedure of adopting and introducing a SOA as enterprise architecture — a procedure model — is considered a crucial part of a SOA Governance approach. 16 of 22 approaches point out the importance of a procedure model, most of them from the software industry and academia.

What is referred to as *SOA procedure model*, are many different variations of procedures for regulated SOA introduction and operation that are called, e.g., *SOA Lifecycle*, *SOA Governance Roadmap*, or *SOA Adoption Model* by the respective approaches. Generally, SOA procedure models act as a global guideline for the future development of the SOA system. They designate and communicate planned future developments of a SOA system and describe the phases from plan to realization.

Governance Processes and Policy Enforcement

Nine out of 22 approaches formulate governance processes and policy enforcement to be crucial aspects of SOA Governance, four of them are founded recommendations from academia. *Governance processes* are the actual implementation of governance. They define the business and IT-internal processes that are required to operate an IT system from the perspective of governance. They provide the activities and accountability for the operation of a SOA on a meta level. Mechanisms for automated policy conformance checks are summarized by the term *policy enforcement* used by the approaches. They target the monitoring of adherence to policies and their operational enactment and are integrated in processes.

Many approaches mention the category *Processes* as a central point of their approach. However, the classification types vary from governing vs. governed processes [18, 22, 12], runtime vs. design time governance [32, 4], policy-related vs. review-related processes [17], organizational structures vs. employees [31], processes vs. organization, infrastructure vs. maturity [14], and architecture review processes [20]. The classification that is mentioned most frequently is *governing vs. governed processes*. *Governing* processes cope with performing and realizing governance methods and structures. They serve as a means for the governance approach. *Governed* processes are subject to governance. They represent activities such as service development, process management, and service operation. Further, all authors agree that concise definition and structuring of governance processes is crucial to the successful operation of a SOA system.

Concerning policy enforcement (as part of *governing* processes), all approaches propose control points that reside in (cyclic) governance processes. Techniques or concrete examples for automated policy enforcement (other than manual revision of artifacts) are provided by none of the approaches.

SOA Maturity Measurement

According to Windley [49], implementations of governance that are not adjusted to the scope and maturity of a SOA system cannot display its full effect: either they exercise too few control, or they limit the involved persons by an overdose of regulation in their freedom of action and have a demotivating effect [50, 49]. Governance methods and procedures are to be planned proactively, in order to keep up with the development of the SOA system and enable controlled growth. Documentation of the planned development as a roadmap is an often proposed method to keep track of the state and development direction of the SOA system. In order to assess the current maturity of a SOA system, SOA maturity models have proven useful [44, 1, 13, 14, 28].

Overall, SOA Maturity Models are explicitly considered in four out of 22 approaches (ca. 23%), where three mentions come from the practitioner's domain, and one from scientific work. As SOA Maturity Models are already widespread and well-known instruments of SOA Governance, it seems astonishing that the integration of maturity models into SOA Governance is proposed by a minority of authors. Obviously, only few authors recognize the benefits of maturity measurement in the context of SOA Governance. However, several additional authors proclaim SOA Maturity Models-related methods. So it might be a lack of awareness which causes the little assignment of maturity models to governance.

SOA Metrics

Almost half of the approaches mention a *metrics system* as an important building block for SOA Governance. Metrics, in general, are defined along with goals and make processes and parameters of the SOA system more transparent. The measurement of goals, combined with a supporting management structure, supports the judgement on the effectiveness of the adoption of an IT system such as SOA. For the implementation of SOA Governance in a company, a set of goals is usually defined that are to be achieved. Metrics, in general, report on the performance of the SOA system as a whole, by measuring the goals set by the governance initiative (cf., e.g., [17, 22]).

Improving the assessment of achievement of SOA goals by the definition of a metrics system is considered an important aspect of SOA Governance by all authors mentioning this issue. Most of the authors especially emphasize the management of service operation, service statistics, project performance, and the relationship to employee behavior to be important in the context of metrics for SOA Governance. Further, the measurement of service reuse is an important aspect.

Summary

We compared the structure and core aspects of several approaches that first structured SOA Governance. As a result, ten components have been identified, that most of the authors make use of to compose their approaches.

The approaches do not usually adhere to consistent criteria, as done by the presented analysis. Most approaches use either organizational means, SOA goals or governance guidelines as a main criterion. In most cases, one important aspect is selected, and other (equally important) ones are presented in a cross-sectional way. The inherently multi-dimensional nature of this area is simplified and reduced to a few structuring criteria in most cases. However, there seems to be no reason for the selected and presented outlines — the choices of main criteria seem arbitrary.

In unison, the authors agree on the necessity of SOA. Based on common characteristics of SOA systems and the emerging challenges, the installation and operation of governance approaches for SOA is considered essential, on the one hand, regarding the management and unification of SOA-inherent heterogeneity and complexity, and, on the other hand, on the regulation and exploitation of new capabilities such as cross-organizational service deployment. SOA Governance turned out to be an area that is structured in various dimensions, e.g., goals and strategy, organizational structures, roles and employee behavior, software support.

Only few proposals [32, 22, 11] present holistic approaches that tackle all or most of the identified components. The overall comparison shows that most approaches are characterized by a *tunnel perspective*, limiting the focus on selected issues. However, the majority of authors agree that a holistic governance approach is crucial for SOA Governance. In the remainder of this chapter, we specify building blocks for a governance approach tailored to the needs of the Internet of Services. In particular, we adopted the results from *Organizational Structure* and *Role and Responsibilities* in our stakeholder map, while the process framework reflects insights from the *SOA Procedure Model* as well as *Governance Processes*. The component *Metrics* is adopted in the Measurement Framework, and *SOA Maturity Measurement* in the Maturity Model and Capability Profile. The service description management for USDL has been influenced by the insights of *Governance Policies* and *Governance Processes*.

16.3 Building Blocks of a Service Governance Framework

Based on an analysis of related work, we propose four building blocks to instantiate a governance framework that is capable of supporting the operation of such a platform. We base our conceptual considerations on existing frameworks and also take into consideration the particularities of emerging SOA Governance frameworks.

First, the *Process Framework* defines tasks and activities required to manage the ISM and its lifecycle. Especially, the areas of “service portfolio management,” “service lifecycle management” as well as “broker operations” are not adequately represented by current frameworks and are developed in this component. Roles and responsibilities of the processes, tasks are focused on in the second building block, namely, the *Stakeholder Map*. The third building block, viz., the *Measurement Framework*, describes corresponding key performance indicators and other result measures, which are used to evaluate process quality as well as the compliance with internal, normative, and legal regulations. The fourth building block is a *Maturity Model*. The application to a service-oriented IT system allows the evaluation concerning system maturity and identification of potential gaps, which need to be covered by additional governance processes.

16.3.1 Stakeholder Map

Generally, in the Internet of Services, which we consider as the basis for (future) service marketplaces, several main stakeholders have been identified: *service provider*, *service broker* or *intermediary,* and *service consumer* [16]. While the service consumer and the service provider are actual persons acting as a specific stakeholder, the service broker is a virtual entity, a marketplace, or a piece of software. Nevertheless, it is operated by actual persons who act as a certain stakeholder.

With the emphasis on the complete service lifecycle, including the inception of a service and its after-sales, i.e., the community around the platform, these roles need to be extended. As outlined above, the service broker itself is not a stakeholder in that sense. That means its role cannot be taken by any person but is, instead, a piece of software. However, a supporting stakeholder, such as the platform host, needs to be established. Fig. 16.1 shows all roles. Note that multiple instances of each role but the operating platform host communicate via the service marketplace.

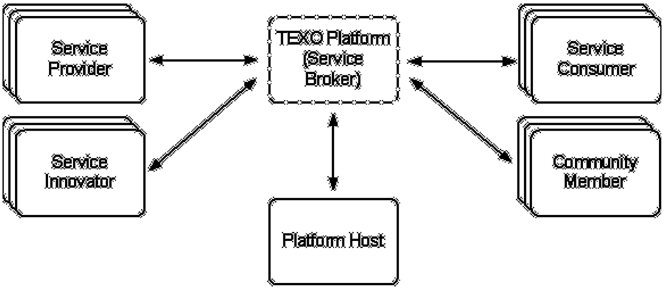


Fig. 16.1: General Service Marketplace Roles [16].

In the following we describe and detail the five *stakeholders*. For each stakeholder, associated activities are also outlined. A *role* is a subordinated entity of a stakeholder. Thus, each stakeholder may have several roles. Primarily a governance framework for marketplace platforms does focus on the platform host's activities.

But the framework also needs to take into consideration that services are neither produced nor consumed by the platform host but a provider and consumer. As such these roles have to be taken into consideration.

16.3.1.1 Service Provider

The service provider supports agencies that hold governance and operational responsibility for a service, including organizational structures and other business aspects, as well as systems and other implementation artifacts. The service provider represents the role of a development party, producing and publishing services ready for execution. Largely, they are the service owners, responsible for the service implementation as well as maintenance. Unlike traditional software producers, service providers develop services that remain in the same organization, rather than being delivered to software clients (what is also possible). Therefore, during requirements engineering, it is the duty of service providers to not only analyze the objective, functionality, interface, and quality of service, but to also consider accessibility, retrievability, how to manage service level agreements (SLA), and define policies, etc. At runtime the provider may have to provide second level support and appropriate change management. Possible roles include the service manager, service clerk, service producer, content provider, service aggregator, service integrator, business expert, service engineer, service designer, and service programmer (for more detail on roles cf. [27]).

16.3.1.2 Platform Host

The platform host administrates the marketplace platform, including the user management, and the maintenance of all running management services, as well as platform governance, risk, and compliance. For the platform host several roles are distinguished. The platform host has to support agencies that specialize in taking services out of markets and driving up their consumption through competitive pricing models. These agencies provide a further intermediation, managing the front-desk delivery of services to customers without encroaching on back-office responsibility. The platform host may have to certify service providers and their offers, since not all offers are acceptable. A role certifier is needed (perhaps even an additional stakeholder). Another point is billing and payment tracking, which also is a role often held by additional stakeholders.

With an increasing number of services, registries are becoming more and more important. They serve as a central location for tracking and managing services. The reusability of services depends on these registries, as these provide a way to share services across organizational borders. The platform host has to keep his registry and search index current as the central information database and its timeliness is crucial to the success of the whole system. Additionally, registry maintenance is of importance. For instance, a service that is updated while being in use should not be interrupted in execution, the removal of services that were never or seldom invoked should be considered, a rating system could be included, etc. During service delivery, a controlled service provisioning has to be ensured. In order to guarantee that clients can be charged by the providers and to ensure security, services are only accessible by authorized users. SLAs are used as contracts and authorization. These usually define costs, assured availability, performance, etc. As soon as services start being executed, the platform host begins the service monitoring process in order to ensure SLAs as well as policies and keeps track of the behavior of published services.

The role platform support gives support (at least second level support) to all marketplace platform processes, assisting various other stakeholders that are interacting with the platform including the service consumer, service innovator, service provider, or community member. Platform support is, e.g., a call center agent or a support consultant.

Further roles can include a business manager, governance officer, ontology and standards engineer/ expert, host architect, hardware admin, and software admin.

16.3.1.3 Service Consumer

The service consumer finds services, based on his functional and non-functional requirements and selects from offered variants (e.g., SLA variants) via the marketplace of the SaaS platform, buys or licences them and then may request and invoke them. For the service consumer several roles are distinguished such as business user, expert user, or manager, and perhaps administrators (each with approval rights). Additionally, a guest (of the platform) may only browse the offers. He has to register and login for ordering.

16.3.1.4 Community Member

Community members are registered and non-registered users of the marketplace. Most other stakeholders can act as community member stakeholders: Roles attached to the platform host are excluded, as they do not take part in the community in this sense but only from an administrative side. Community members — in addition to their possible other stakeholder roles — provide feedback for tradable services (e.g., problem reports) and use wikis, web logs, and forums provided by the marketplace platform to discuss tradable services.

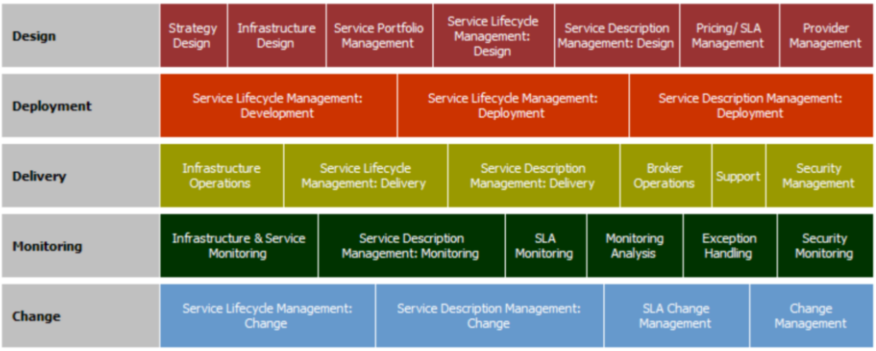
16.3.1.5 Service Innovator

Service innovators use the marketplace platform to innovate on tradable services. The innovator derives new ideas from direct feedback from service consumers, query logs, or other data (e.g., wikis, blogs, etc.) or creates new ideas for services from scratch. Service innovators collect, aggregate, store service ideas in an idea repository, and rank these collections of ideas. Service innovators need to be registered in the marketplace platform, for use of the service browser and service discovery, as well as the community portal to browse consumer feedback.

16.3.2 Process Framework

All relevant governance processes have been grouped in five phases to increase the accessibility of the framework: design, deployment, delivery, monitoring, and change. In each of these phases, several processes constitute the process framework. Figure 16.2 provides an overview. As the framework has been compiled on the basis of existing frameworks, some processes have already been considered in existing frameworks. Most of the time, the existing processes will need to be extended to cater for the specific needs of marketplace platform governance.

The design phase contains all sorts of strategic aspects of the use or operating of the marketplace platform and its traded services. The development and deployment of services, as well as the selection of third-party services are components of the deployment phase. The delivery phase contains all aspects of service and infrastructure operations. It is closely coupled with the monitoring phase as they are executed concurrently. The monitoring phase contains all aspects of service and infrastructure monitoring. It is closely coupled with the delivery phase as they are also executed concurrently. The change phase contains all processes and tasks needed to adjust and change the infrastructure and software traded as service.



Design	Strategy Design	Infrastructure Design	Service Portfolio Management	Service Lifecycle Management: Design	Service Description Management: Design	Pricing/ SLA Management	Provider Management
Deployment	Service Lifecycle Management: Development	Service Lifecycle Management: Deployment	Service Description Management: Deployment				
Delivery	Infrastructure Operations	Service Lifecycle Management: Delivery	Service Description Management: Delivery	Broker Operations	Support	Security Management	
Monitoring	Infrastructure & Service Monitoring	Service Description Management: Monitoring	SLA Monitoring	Monitoring Analysis	Exception Handling	Security Monitoring	
Change	Service Lifecycle Management: Change	Service Description Management: Change	SLA Change Management	Change Management			

Fig. 16.2: Process Framework.

16.3.2.1 Design

The design phase includes organizational and infrastructure aspects as well as the service portfolio planning and the alignment of business requirements with the IT. It also comprises legal issues concerning general terms and conditions as well as SLAs. Furthermore, a process of provider management has to be introduced (including certification) because a marketplace platform is essentially a supplier/service provider enabling approach.

Strategy design covers all activities that are related to the creating and reviewing a strategic IT plan. This mainly involves strategic alignment, dependency analysis as well as the specific consideration of SOA capabilities. Strategy design also involves financial management tasks. *Infrastructure design* processes comprise activities which ensure a sound architecture specification. Accordingly, general processes on standards and development plans, as well as service marketplace-specific processes on running processes, organizational units, and their relationships are covered. This involves, e.g., the identification of system owners, data owners, and services owners. *Service portfolio management* encompasses all available processes on the infrastructure. Their management is of high importance to smooth operations. Thus, the governance of the composition of services, their granularity, their description as well as portfolio development are the key tasks. In addition to that, capacity planning is of essence as is the management of service continuity. While *service lifecycle management* as such covers all phases of the framework, there are specific tasks which have to be conducted in every of its phases. The design phase is the first. In order to properly deploy and deliver a service, the service has to be configured and validated before a transition strategy for service operations can be designed. Similarly, *service description management* covers all phases of the framework as services have to be described properly according to a certain schema in order to be discovered and used. Since services are traded, the governance of the underlying *pricing models and SLA* becomes important. The design of an SLA framework, the design of standard terms and conditions as well as the design of payment/ pricing models has to precede the service offering. As most of the services can be provided by third-party service providers, it is important to include a process to specifically manage their involvement. The process of *provider management* is a portfolio management process to evaluate and engage or disengage providers. Also, the governance of third-party design processes is part of provider management.

16.3.2.2 Deployment

The deployment phase comprises all processes that surround the deployment of services. This covers the service catalogue management, service continuity management, service validation and testing, and the definition/ negotiation of SLA. *Service development* has been consciously separated from this framework as we only cater for run-time governance. Service development and engineering is a broad topic of its own.

Service lifecycle and *service description management* for *deployment* cover all governance processes that are executed when a service is to be deployed. This involves service catalogue management, service continuity management as well as the execution of validation and testing. In order to create a service offering, the corresponding SLA, operation level agreements (OLA), and general terms and conditions have to be chosen, too.

16.3.2.3 Delivery

The delivery phase contains all aspects which guarantee the delivery of services. Thus, it includes service and infrastructure maintenance. Here, both the *operation* of mere infrastructure software and hardware as well as the assurance of service performance is managed. In addition to that all business functions concerning the brokerage of services have to be governed properly. Thus, the process *broker operations* has to be introduced which contains sub-processes which stem from the common phases of business transactions to reflect all phases of a purchase: Initiation, Agreement, Settlement, After Sales. Furthermore, all support operations (such as help-desk) and the management of the organization are part of this phase. Finally, security and compliance issues are also addressed here. Processes that deal with monitoring have been grouped in a separate phase.

Managing the marketplace platform involves several governance processes which are tightly related to IT management operations. Data as well as the physical environment, i.e., servers, data storage, network, need to be properly protected and/or backed up. In addition, the operation of the infrastructure needs to be safeguarded, i.e., regular maintenance has to be conducted and software updates have to be applied. Besides infrastructure operations the service operation, too, needs to be governed but during *service lifecycle management*. All services which were designed, described, and deployed need to be operated so that performance and capacity requirements can be met. A continuous service must be ensured according to SLAs. The central component of a marketplace platform is the service broker which facilitates the communication between service provider and service consumer. It ensures that a service can be searched for and discovered, contracts can be negotiated, and services can be bought and payed for. Also after-sales routines need to be governed. The structuring of this new component is based on the common phases of transactions for e-commerce purchases on a marketplace [45, 38]. Both, the operation of the infrastructure and of all services, needs to be supported by a proper (multi-level) *support* help-desk. It provides support services and incident management to customers and providers alike. All interaction on the marketplace platform needs to be *secure* as it involves business transactions. In order for the infrastructure to run secure, measures have to be taken to ensure authorization and authentication. Measures for security breaches and constant vulnerability assessments have to be in place. This includes data privacy issues.

Execution is both, the delivery and the monitoring of services and infrastructure. As both topics are closely related but focus on different parts of the execution they are separated logically in the framework. Monitoring covers the observation of services and infrastructure concerning performance, (future) capacity, and fulfilment of SLAs. Data analysis, exception handling, and security specific tasks such as logging are also within the scope of this phase. Error logs have to be analyzed for preparing error corrections.

16.3.2.4 Monitoring

Monitoring is an important governance process to ensure that the infrastructure and services are delivered according to plan. In order to do so, both, the *infrastructure* as well as the *services* including their *descriptions*, need to be closely *monitored*. Governance tasks involve the actual setup of a monitoring organization which specifies the approach, measures, objects to be measured etc. Areas to be monitored are, e.g., third-party services from service providers, overall service performance as well as infrastructure capacity and thresholds. Besides monitoring the infrastructure and service functionality, the contracted service levels need to be safeguarded, otherwise compensation routines have to be executed. This entails that governance processes for the specific monitoring of *SLAs* have to be in place to constantly monitor and review the execution and compensate for violations. Monitoring is no end in itself. The as-is data has to be correlated with planned/ predicted performance, so that weaknesses can be identified and proposals for improvement can be derived. This *monitoring analysis* process usually involves the generation of reports and the use of descriptive data analysis techniques (e.g., Online Analytical Processing). However, in order to allow not only ex post analysis, more intelligent ex ante analyzes are desirable to enable the broker to predict the impact change will have on future operations (architecture management), e.g., when replacing services. While monitoring data and their analysis usually means to compile reports and analyze aggregated data, the *exception handling* processes deals with singular events and is designed to single out irregularities and provide patterns so that events can be correlated and appropriate responses can be selected. This in turn allows the categorization and prioritization of incidents to escalate and recover. Similar to security in the delivery phase, *security monitoring* is necessary to ensure a consistent behavior. Also, in order to comply with legal requirements, certain tasks may need to be monitored and logged.

16.3.2.5 Change

The change phase contains all processes and tasks needed to adjust the infrastructure and services traded on the platform in order to ensure compliance and quality of service. That comprises processes which deal with the change and retirement of the actual services, change management processes for SLAs as well as change management from an organizational perspective.

Within *service lifecycle management*, the deployment of each new release of a service needs to undergo a specified change management governance process in order to ensure the continuous service provisioning. These activities involve, e.g., test plans and deployment verification. In case of a service retirement, contracted warranties have to be enforced, contracts may have to be terminated or changed, and equal functionality may have to be offered as a replacement. Also maintenance requires the removal of defunct services. Service functionality may change, business models may change over time. This needs to be reflected in the contracts which were signed on the platform. Changes within *service descriptions*, *SLAs*, pricing models, and general terms and conditions are governed by these processes to ensure due diligence and traceability. This can be conducted either in conjunction with functionality change or without. All activities in the change phase need to be managed by a proper *change management* organization. All change requests and all managed change need to be documented, prioritized, and evaluated before the change ticket is closed. The change itself — at least in complex cases — will be followed by design, development, deployment, i.e., here the overall process may start again.

16.3.3 Measurement Framework

We propose a multi-stage measurement framework composed of three layers to assess the performance of the governance framework processes. Fig. 16.3 depicts the different stages: Company Scorecard, IT Balanced Scorecard, ITIL/COBIT-based Processes and KPI (key performance indicators).

The first layer describes the company scorecard including the vision, mission, and strategy of the company. The most generic stage is the vision of a company describing the mission statement. It explains the reason for a company to exist. The second stage is the mission of the company describing specific goals in terms of performance, costs, ROI or market goals. The strategy follows as stage three and defines the specific way to achieve the company goals. These three stages are defined by the top management and do not have a standardized way of measuring the achievement of the goals. Instead they are discussed by the responsible managers in person.

The second layer refers to the IT Balanced Scorecard pointing out the strategy of the business area. The IT Balanced Scorecard is composed of six perspectives including financial management, process management, provider management, employee management, innovation management, and product management. Referring to Kaplan and Norton, the four perspectives of the classical Balanced Scorecard (financial, customer, internal process, and innovation & learning perspective) are extensible and should be adjusted to the own needs [30]. Objectives are deduced by concentrating on the main characteristics of the strategy defined on the third stage. These more clearly outlined objectives serve as a basis for the critical success factors which are determined for specific perspectives of the IT Balanced Scorecard.

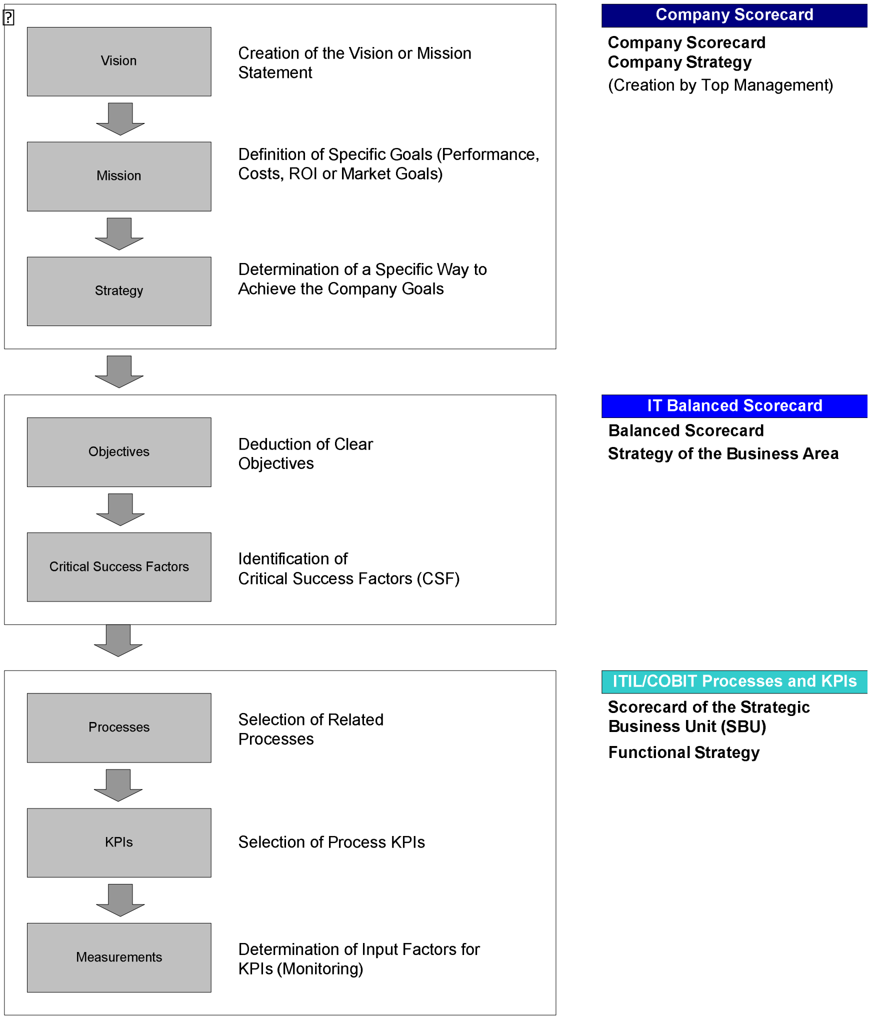

Vision
Creation of the Vision or Mission Statement
Mission
Definition of Specific Goals (Performance, Costs, ROI or Market Goals)
Strategy
Determination of a Specific Way to Achieve the Company Goals
Company Scorecard
Company Scorecard
Company Strategy
(Creation by Top Management)
Objectives
Deduction of Clear Objectives
Critical Success Factors
Identification of Critical Success Factors (CSF)
IT Balanced Scorecard
Balanced Scorecard
Strategy of the Business Area
Processes
Selection of Related Processes
KPIs
Selection of Process KPIs
Measurements
Determination of Input Factors for KPIs (Monitoring)
ITIL/COBIT Processes and KPIs
Scorecard of the Strategic Business Unit (SBU)
Functional Strategy

Fig. 16.3: Measurement Framework.

Each critical success factor is measured by few KPIs belonging to one of the six perspectives.

The third layer comprises processes. A suitable selection of processes from the process framework makes up the sixth stage. For each process there exist several KPIs measuring the performance of the specific process, related to the KPIs of the CSF. A proper selection has to be made for the specific case. The final part of the framework comprises monitoring measures serving as input factors for each KPI on the eighth and last stage.

While the distribution and transformation of governance requirements to lower levels follows a top-down approach, the requirements and goals can be measured and controlled on a bottom-up basis by a KPI system. Therefore, the monitoring data on the lowest stage serves as input for KPIs on the next higher stage. The KPIs feed measures on the stage above and so on.

16.3.4 Maturity Model and Capability Profile

A maturity model is a method for evaluating and measuring the current state of service adoption of an organization. Each organization can class in one level based on different characteristics. The maturity model reveals the organization's weaknesses and helps to develop transition plans to achieve the next maturity level [10]. Our maturity model is presented in Table 16.2.

We base our considerations on the following characteristics (cf., e.g., [15, 36, 39]):

Technology/ architecture Level of the underlying architecture, level of integration of for example databases or legacy systems, or the implementation of monitoring and optimization tools

People/ organization The employee's knowledge, characteristics of enterprise culture, and employees' motivation

Adoption scope Organizational focus of SOA, inter-departmental adoption of SOA, supply chain scope of SOA

Process Orchestration of services, business processes

Standards Technical standards, eBusiness standards, standardized approaches

SOA development Maturity of the SOA development process, existence, acceptance, documentation, and communication of an organization-wide standardized SOA approach

Besides these aspects, further dimensions are suggested for a maturity model, for example, the level of tool support, information management as well as lifecycle maturity or governance maturity. Most SOA maturity models consist of five levels such as the Capability Maturity Model (CMMI) [10]. However, their labels are different due to their different focus. The level labels are named according to the CMMI, as various dimensions are considered and the CMMI labels can cover all described aspects.

Table 16.2: Maturity Model.

Dimension	Level 1 – Initial	Level 2 – Repeatable but Intuitive	Level 3 – Defined	Level 4 – Managed and Measurable	Level 5 – Optimized
Technology/ Architecture	Platform-dependent point-to-point services, no standardized or centralized SOA technology, small number of services	Distributed systems, integration of applications, databases, and legacy, planned architecture vision, versioning and security	Reusable and discoverable services, long-running transactions, service versioning	SOA monitoring, event-driven dash-boards and alerts	Business process tools, easy assimilation of new technologies, composite applications, highly flexible architecture
People/ Or-ganization/ Stakeholder	Some self-taught SOA-skills, disconnected SOA project teams, no SOA specific organizational occurrence	SOA leadership and sponsorship through CIO, SOA Competence Centre, SOA skills developed	Incentives established for to encourage SOA adoption, executive commitment for SOA, strong SOA skills	Reuse and measurement culture, CFO sponsorship for SOA	Agile and continuous improvement culture, all responsibilities assigned and defined
Adoption Scope	Intra-departmental adoption	Business unit level adoption	Cross-business unit level adoption	Enterprise-level adoption	Value net respectively supply chain adoption
Process	SOA knowledge available via individual competence	Modeling of business processes with service components, first reusable processes implemented on a project basis	Modeling, documentation, and implementation of business processes based on SOA components across business areas and organizational units	SOA framework and service components are systematically and proactively managed	Systematic approach established for identifying new requirements and detecting gaps, continuous improvement
Standards (sample)	SOAP/REST, XML, WSDL, J2EE	UDDI, WS-Security	WS-BPEL, ebXML	Project management standards, business activity monitoring	Business process modeling standards
SOA Development	Minimal documentation of architecture, no formal service development process, no communication across project teams	Some level of architectural documentation, reusable architecture within project teams, as hoc communication across project teams	Standardized architecture defined, project teams are encouraged to use architecture, support levels are established	Standard approach for SOA, including processes, technologies, and components	Architectural framework in place for each team to expose and consume services including external partners, continuous architecture improvement
SOA Governance	No formal SOA Governance concept in place	Regular SOA Governance practices take place, identified problems are tackled by project teams that are formed when necessary	An organizational and process framework is defined as a basis for SOA Governance, specific procedures for SOA management in place	Target-setting has developed, integration of business BSC	SOA Governance is sophisticated approach using effective and efficient techniques

In addition, capability profiles represent the application of the maturity model on a SOA system and outline the overall abilities of the system compared with the planned targets. Commonly, the purpose of capability profiles is to provide a blueprint of a system's current respective abilities related to specific domains [25]. In the case of IT Governance, a capability profile is created by the assessment of an IT system using a maturity model — it illustrates the situation as-is [6]. Along with a governance framework, adoption models or best practices are often provided, that, in some cases, are part of the framework itself [13]. One kind of adoption support or recommendation is to provide an assessment of reference processes concerning their importance in implementation. This provides an order as well as a benchmark that can be applied later on.

We built a process maturity matrix that includes recommendations for which processes to address with high priority, depending on the targeted maturity level. Our method visualizes capability profiles by emphasizing the importance of specific processes. Thus, it allows a weighting concerning the ordering of process adaptation in order to achieve given maturity levels when implementing the governance approach. It represents an adaptation tool for planning support, especially concerning the implementation details of reference processes, using a *percentage completion*-assessment.

Once a process matrix is defined, generic capability profiles, one for each maturity level, are generated. These capability profiles are aligned along the five phases: design, deployment, delivery, monitoring, and change.

During the generation process, each of the process adoption steps preparation, implementation, and consolidation is weighted concerning expected effort, as well as the respective processes are weighted inside the process domain. This way, the expected percentage value of implemented processes per governance phase is computed. For the following configuration, the resulting radar chart is outlined in Fig. 16.4: process domains: uniformly weighted. Each axis represents one governance phase of the governance framework: Design, Deployment, Delivery, Monitoring, and Change.

Each of the five axes indicates the implementation progress achieved per governance phase on a percentage scale. The diagram shows that the capability profile for maturity level 1, *initial*, poses no requirements concerning any monitoring processes, touches design, change, and delivery-related processes, and demands almost 25% of deployment-related processes to be implemented. This is due to the fact that processes of the deployment phase are considered important for the second level and need to be considered in the first instance when adopting reference processes.

Maturity level 2, *repeatable but intuitive*, demands a solid basis of implemented processes in each of the five phases. Deployment is once more considered far more important than monitoring. The diagram outlines the importance of level 3, *defined*, that covers over 50% of implementation progress of all five phases. In particular, it requires the complete realization (including consolidation) of all deployment processes.

Levels 4 and 5 perform the optimization of processes. For Level 4, *managed and measurable*, the realization of all processes of the phases *Delivery* and *Change* is re-

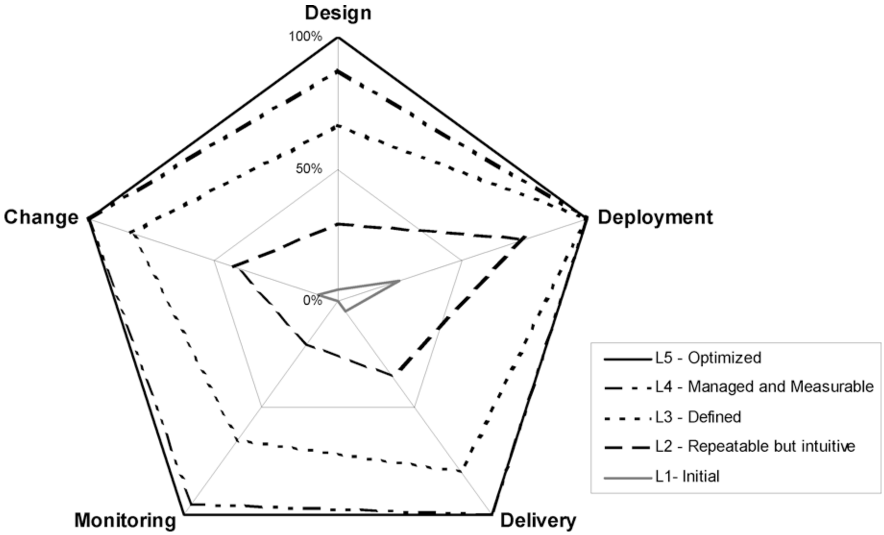


Fig. 16.4: Capability Profiles for Maturity Levels.

quired. Level 5, *optimized*, basically consists of process improvement in *Monitoring* (Monitoring analysis) and *Design* (Strategy design).

This approach provides an overview and visualization of the overall governance process in connection with system maturity. Vice versa, it provides an easy means to estimate the maturity level of the system based on the measured or estimated implementation progress of reference processes by the comparison of respective capability profiles. This analysis allows the combination of the governance framework and the maturity model and hence supports management decision-making.

16.4 Service Description Management for USDL

16.4.1 Service Description Management Processes

In addition to the creation of the initial service description, descriptions are constantly subject to change due to service upgrades, changes in service functionality, changes of the used technical terms, changes in word usage, and many more and lead to different service description variants or configurations (see Chapter 17). In order to reduce the risk of fraud, data inconsistencies, and resulting damages for stakeholders, it is important to define regulations concerning the organizational model (roles and responsibilities) and general procedures. The goal is to organize the handling of descriptions by assuring consistency of the used service description meta-model.

From a software engineering point of view, service descriptions can be seen as *traditional* development objects. Change processes are common and for specific cases (create, update, remove, distribute, ...), there is a number of appropriate (technical) ways to address these issues. However, for the traceable and controllable operation of a service marketplace, it is essential (in the sense of adhering to a general guideline or law, e.g., the Sarbanes Oxley Act) to assure guideline-consistent behavior in the general processes (here: handling service description). In addition to the automation of concrete data handling activities, the according processes need to be observed in order to globally assure efficient control and, in the end, compliance. Governance approaches introduce structures and guidelines aiming at achieving these goals.

We incorporate processes to complement the governance framework (throughout the governance phases) for the regulation and standardization of service description processes, as well as the setup and maintenance processes for service description meta-models (here: USDL). Overall goals are data consistency, sustainability, traceability, reliability and transparency of IT processes. Processes that define the extension of the framework concerning governance of service description are twofold as you need to manage instances, i.e., concrete service descriptions, and the meta-model. Consequently, for all phases of the Service Description Management, we distinguish the processes of *Setup and Maintenance of Service Description Meta-model* and *Service Instance Description*.

The framework defines the phases as a control cycle from an IT system operation perspective. The activities are organized along the phases in Table 16.3.

In the design phase, the host architect (HA) and the business manager (BM) define the standards that will be used to design the actual service description meta-model for the platform. USDL could be one of the standards. The HA and software administrator (SA) define the concrete repository as well as the maintenance and versioning procedures. Also, the BM and the HA define the concrete organizational model to support these maintenance processes for their platform. The processes may be adapted from other governance frameworks such as the TEXO Governance Framework [27]. Similarly, the supporting tools for meta-model design and maintenance have to be chosen. The BM and HA define description guidelines in order to achieve uniformity. This way, they make sure that the same elements in different meta-models are named in a uniform way. Finally in the design phase, the BM then designs the concrete meta-model and the HA checks and implements it from a technical perspective.

The concrete deployment of the meta-model management tools as well as the meta-model itself, takes place in the deployment phase and is executed by the SA.

The delivery of the description is automated, all related governance processes are related to its monitoring.

Accordingly, in the monitoring phase, the BM verifies and certifies that all meta-models conform to the guidelines. He is supported by the meta-model management tools. But ultimately, he is accountable for the data. The SA performs periodic consistency checks to eliminate technical inconsistencies, e.g., after deletions. Advanced monitoring tasks are the monitoring of word changes over time,

Table 16.3: Service Description meta-model Setup and Maintenance Processes. BM — business manager (host), HA — host architect (host), SA — software admin (host), SM — service manager (provider), R stands for responsible, A for accountable.

Phase	Processes	Roles
Design	- Define description standards and meta-modeling language	HA, BM
	- Define repository maintenance and versioning	HA, SA
	- Define organizational model	BM, HA
	- Determine tooling support	BM, HA, SA
	- Define general description guidelines	BM, HA
	- Create meta-model	BM, HA
Deployment	- Meta-model deployment (Establish repository and load meta-model in repository)	BM
Delivery		
Monitoring	- Verify and certify adherence to description meta-model guidelines	BM
	- Perform frequent consistency check	SA
	- Monitor word usage and meaning development over time	BM (A)
	- Monitor changes and perform development trends analysis	BM (A)
	- Analyse feedback from monitoring of semantic applications	BM (R)
Change	- Change description standards	BM (A), HA (R)
	- Change maintenance processes	BM (A), HA (R)
	- Change maintenance roles and responsibilities	BM (A), HA (R)
	- Change peripheral (non-core) modules of description meta-model	BM (A)
	- Change core modules in description meta-model	BM (A)

trends, and semantic checks. Terms change over shorter periods of time, so that meta-model changes may be necessary. Although this more apparent on the instance level, changes of the meta-model may appear. Sometimes, a trend can be calculated from these changes to anticipate necessary future modification. Semantic applications can also provide feedback on the use of technical terms or tags in the meta-models. Again, we assume the latter two are more important on an instance level but still, in a large deployment the may be enough data to perform this kind of monitoring.

In the change phase, standards as well as the associated governance may be adapted if the monitoring of the marketplace platform suggests this. Change may impact the description standard, e.g., the addition of core data types from CCTS to USDL, the roles and responsibilities, e.g., the reassignment of BM tasks to and ontology engineer, the change of processes, e.g., the addition of a community liaison process to bring the service community up to speed about changes in the meta-model, as well as implementing changes in the actual core meta-models or their modules.

For more details on the roles cf. [27]. We also included accountability (A) and responsibility (R) information in the table according to RACI matrices.

Table 16.4 lists essential reference guidelines that are required for successfully designing, setting up, and maintaining a repository used for service description in an environment such as the Internet of Services. Guidelines described in Table 16.4 focus on the usage of the service description repository, the service ontology, throughout the governance lifecycle.

Table 16.4: Service Description Setup and Maintenance Processes. BM – business manager (host), HA – host architect (host), SA – software admin (host), SM – service manager (provider), R stands for responsible, A for accountable.

Phase	Processes	Roles
Design	- Determine tooling support	HA, SA
	- Define general description guidelines	BM
	- Set up repository for service descriptions	HA, SA
Deployment	- Verify description instances	BM
Delivery		
Monitoring	- Verify and certify adherence to description meta-model and description guidelines	SA, BM
	- Monitor word usage and meaning development over time	BM (A)
	- Monitor changes and perform development trends analysis	BM (A)
	- Analysis of semantic applications and usage feed-back	BM, SA
Change	- Change description guidelines	BM, HA
	- Change description process	BM, HA
	- Change service description	BM, HA, SM

Similarly, to the governance of meta-models, the instances of service descriptions have to be governed. In the design phase, approved tooling for actual descriptions needs to be chosen and modeling guidelines have to be published so that the description files are uniform and comparable for the service consumers.

When providers deploy their services so that they can be sold via the marketplace platform, their descriptions have to be checked for compliance to the guidelines set forth in the design phase. This entails a semantic as well as a syntactical evaluation.

Naturally, the actual delivery takes place automatically, similar to the meta-model.

Just like in the case of the meta-model, word changes are tracked, trends are estimated and semantic checks are performed. In addition user feedback is evaluated by the BM and factored into the verification of service descriptions to their meta-mdoel.

In the change phase, the BM and HA adapt processes and guidelines for service descriptions as deemed necessary through monitoring and communicate the change to the users. If necessary, the concrete description of services has to be changed here as well.

16.4.2 Exemplary Application

When designing the service description meta-model, one of the first decisions will have to be on the used standards. If we assume USDL is taken as a basis, then (with a look at the running example) a meta-model for an agent's address might look as follows (rendered as XML for the sake of readability in Listing 16.1):

Listing 16.1: Agent address meta-model.

```
<PhysicalAddress xmi:id="">
  <Street></Street>
  <StreetNumber></StreetNumber>
  <City></City>
  <Postcode></Postcode>
  <State></State>
  <Country></Country>
  <GeographicalPoint xmi:id=" ">
    <Latitude></Latitude>
    <Longitude></Longitude>
   </GeographicalPoint>
</PhysicalAddress>
```

It is a design decision to adopt this meta-model or to add or delete information (e.g., remove the state for an exclusively German platform). Also, the meta-model designers need to decide on code lists for the attributes such as country codes or a format for geo-spatial coordinates. USDL is quite comprehensive but changes will be necessary as there may be new forms of communication for virtual addresses or if a country decides to introduce a different system to specify physical addresses. Also, the platform host could decide to implement CCTS and use core data types for addresses which would look significantly different as they carry more detailed data.

On an instance level, the agent's address might look like Listing 16.2.

Listing 16.2: Service description showing a concrete agent's address.

```
<PhysicalAddress xmi:id="PhysicalAddress_27437204">
  <Street>Hacks Cross Road</Street>
  <StreetNumber>3620</StreetNumber>
  <City>Memphis</City>
  <Postcode>38125−8800</Postcode>
  <State>Tennessee</State>
  <Country>USA</Country>
  <GeographicalPoint xmi:id="GeographicalPoint_26101118">
    <Latitude>35.05107</Latitude>
    <Longitude>−89.792703</Longitude>
   </GeographicalPoint>
</PhysicalAddress>
```

Here, the change of address is probably the most striking use case. In order to keep addresses (and service interfaces up to date) monitoring and feedback information has to be evaluated to make sure all data is current. Also, legal requirements on reporting the accountability for services might change and require different data which used to be optional. Here a change in the meta-model will cascade to changes in the actual service descriptions.

16.5 Conclusion

Based on evidence from existing governance frameworks and research from academia and practice we derived processes, stakeholders, measurements, and maturity levels as the four building blocks for a governance framework that can support operations of a service marketplace.

In our understanding transactions on a service marketplace involve three entities: a service provider, a service broker and a service consumer. Innovators and communities support this process indirectly. The platform host is the key stakeholder to enable this interaction via the broker. However, the provider and the consumer are also involved directly or indirectly in governance processes in the different phases of the service lifecycle and the generation of KPIs about these processes. We distinguish five phases that cover design, deployment, delivery, monitoring, and change of services. The KPIs are broken down from a company-level to an IT-level to these processes. Furthermore, in order to grow and evaluate the overall architecture we propose a maturity model and capability profile. These four building blocks are necessary to form the core of a Governance framework for the Internet of Services.

Furthermore, this kind of Governance has to pay attention to cross-company legal aspects, e.g., data protection/ security. It must comprise contract management over country borders, country-specific laws for data transmission and protection, and laws concerning the fulfilment of online contracts. It also must cover different service monitoring aspects and includes the interests of multiple stakeholders. Operating a service marketplace platform involves much more stakeholders than common SOA approaches. Being a cross-company setup, a framework must consider the interests of all stakeholders of the marketplace platform.

We acknowledge that there may be more aspects to include as related work points out. However, they can most likely be related to the above four building blocks. In this chapter we provided an initial framework that needs to be instantiated according to the application and its process, roles, maturity levels, and metrics need to be detailed for application. At this stage it is also intended as an overview of the challenges governance faces in the Internet of Services such as brokerage as well as an inspiration for further research in this emerging area.

References

1. SOA Governance: Beherrschen Sie Ihre SOA. Business white paper, Software AG, 2005. http://www.softwareag.com/de/Images/WP_SOA_Governance_D_tcm17-22130.pdf.
2. Service Lifecycle Governance: Timely Policies and Enforcement Help Companies Reap the Full Benefits of SOA. BEA white paper, BEA Systems, Inc., 2006. http://www.itworldcanada.com/Admin/Pages/Assets/DisplayAsset.aspx?id=e0a24263-a10a-4887-8d45-582261587176.
3. SOA Governance: Balancing Flexibility and Control Within an SOA. A systinet white paper, Systinet, 2006. http://www.webservices.org/content/download/83830/

1225383/file/SOA_Gov906.pdf.
4. SOA Governance: Enabling Sustainable Success with SOA. Technical report, webMethods, Inc., 2006. http://www.cioindex.com/nm/articlefiles/44428-SOA_Governance.pdf.
5. 2007 Survey on SOA Governance. Technical report, WebLayers SOA Forum, 2007. http://www.bitpipe.com/detail/RES/1209665671_580.html.
6. CObIT 4.1: Control Objectives for Information and Related Technology. Technical report, IT Governance Institute, Rolling Meadows, IL, USA, 2007.
7. ITIL v3: Information Technology Infrastructure Library Version 3. Core OGC titles, Office of Governance Commerce, The Stationery Office, London, 2007.
8. SOA Governance: Balancing Flexibility and Control within an SOA. Technical report, Hewlett-Packard, 2007. http://hp.com/go/soa.
9. Five Steps for Building the Business Case for SOA Governance. Business white paper, Software AG, 2008. http://www.softwareag.com/de/images/BusCase_SOA_WP_Nov08-web_tcm17-45795.pdf.
10. CMMI Models and Reports. Technical report, Software Engineering Institute, 2009. http://www.sei.cmu.edu/cmmi/models/.
11. SOA Governance Framework. Technical standard, The Open Group, 2009.
12. TOGAF 9: The Open Group Architecture Framework. Technical report, The Open Group, 2009. http://www.opengroup.org/architecture/togaf9-doc/arch/index.html.
13. M. Afshar. SOA Governance: Framework and Best Practices. An Oracle White Paper. An Oracle white paper, 2007. http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf.
14. P. Allen. SOA Governance: Challenge or Opportunity? *CBDI Journal*, 7:20–31, 2008.
15. A. Arsanjani and K. Holley. Increase Flexibility with the Service Integration Maturity Model (SIMM). Technical report, IBM developerWorks, 2005.
16. A. P. Barros and M. Dumas. The rise of web service ecosystems. *IT Professional*, 8(5):31–37, 2006.
17. J. Bernhardt and D. Seese. A conceptual framework for the governance of service-oriented architectures. In G. Feuerlicht and W. Lamersdorf, editors, *Service-Oriented Computing - ICSOC 2008 Workshops, ICSOC 2008 International Workshops, Sydney, Australia, December 1st, 2008, Revised Selected Papers*, volume 5472 of *Lecture Notes in Computer Science*, pages 327–338. Springer, 2008.
18. N. Bieberstein, S. Bose, M. Fiammante, K. Jones, and R. Shah. *Service-oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. IBM Press, Upper Saddle River, NJ, USA, 2005.
19. 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.
20. J. Bloomberg. *SOA Governance: IT Governance in the Context of Service Orientation*. ZapThink, LLC, Baltimore, MD, USA, 2004.
21. B. Brauer and S. Kline. SOA governance: a key ingredient of the adaptive enterprise. Whitepaper, HP, 2005.
22. W. Brown, R. Laird, C. Gee, and T. Mitra. *SOA Governance: Achieving and Sustaining Business and IT Agility*. IBM Press, Upper Saddle River, NJ, USA, 2008.
23. P. Derler and R. Weinreich. Models and Tools for SOA Governance. In D. Draheim and G. Weber, editors, *Trends in Enterprise Application Architecture, 2nd International Conference, TEAA 2006, Berlin, Germany, November 29 - December 1, 2006, Revised Selected Papers*, volume 4473 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2006.
24. T. Erl. *Service-oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, Englewood Cliffs, NJ, 2005.
25. D. Herron and D. Garmus. Identifying your organization's best practices. *CrossTalk: Journal of Defense Software Engineering*, 19:22–25, 2005.

26. K. Holley, J. Palistrant, and S. Graham. Effective SOA Governance. IBM White Paper, 2006. ftp://ftp.software.ibm.com/software/uk/flexible/wp/effective_soa_governance.pdf.
27. C. Janiesch, M. Niemann, and R. Steinmetz. The TEXO Governance Framework. Technical report, SAP Research White Paper, 2011. http://www.internet-of-services.com.
28. W. Johannsen and M. Goeken. *Referenzmodelle für IT-Governance: Strategische Effektivität und Effizienz mit COBIT, ITIL & Co.* dpunkt, Heidelberg, 2007.
29. N. Josuttis. *SOA in Practice: The Art of Distributed System Design*. O'Reilly Media, Sebastopol, CA, USA, 2007.
30. R. Kaplan and D. Norton. *Translating Strategy into Action: The Balanced Scorecard*. Harvard Business School, Boston, MA, 1996.
31. O. Kohnke, T. Scheffler, and C. Hock. SOA-Governance — Ein Ansatz zum Management serviceorientierter Architekturen. *Wirtschaftsinformatik*, 50(5):408–412, 2008.
32. E. Marks. *Service-Oriented Architecture Governance for the Services Driven Enterprise*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008.
33. M. Matsumara. The Definitive Guide to SOA Governance and Lifecycle Management. Technical report, 2007. http://www.scribd.com/doc/7056416/Guide-to-SOA-Governance.
34. G. McBride. The Role of SOA Quality Management in SOA Service Lifecycle Management. IBM developerWorks, 2007. ftp://ftp.software.ibm.com/software/rational/web/articles/soa_quality.pdf.
35. T. Mitra. A Case for SOA Governance. IBM developerWorks, 2005. http://www.ibm.com/developerworks/webservices/library/ws-soa-govern/.
36. B. Moreland and M. Afshar. The Path to SOA (Part I of III). http://www.ebizq.net/topics/systems_management/features/7193.html, 2006.
37. M. Muriankara. SOA Governance Framework and Solution Architecture. IBM developerWorks, 2008. http://www.ibm.com/developerworks/webservices/library/ws-soa-govframe/.
38. A. Picot, R. Reichwald, and R. Wigand. *Die grenzenlose Unternehmung: Information, Organisation und Management*. Gabler, Wiesbaden, 4th edition, 2001.
39. A. Pugsley. Assessing your SOA Program. HP White Paper, Hewlett Packard, 2006.
40. J. Quantz. SOA in der Praxis: Wie Unternehmen SOA erfolgreich einsetzen. Berlecon report, BerleCon Research, Mar 2006. http://www.berlecon.de/studien/downloads/Berlecon_SOA.pdf.
41. D. Rane and G. Lomow. SOA Governance: More than just Registries and Repositories. Technical report, BearingPoint, 2008. http://www.digitalgovernment.com/media/Knowledge-Centers/asset_upload_file77_1753.pdf.
42. I. Rieger and R. Bruns. SOA-Governance und -Rollen: Sichern des Mehrwerts einer serviceorientierten Architektur. pages 20–24, 2007.
43. J. Schelp, O. Schmitz, J. Schulz, and M. Stutz. Governance des IT-Sourcing bei einem Finanzdienstleister. *HMD - Praxis Wirtschaftsinform.*, 250, 2006.
44. T. G. J. Schepers, M.-E. Iacob, and P. van Eck. A lifecycle approach to SOA governance. In R. L. Wainwright and H. Haddad, editors, *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, pages 1055–1061. ACM, 2008.
45. B. Schmid. Elektronische Märkte: Merkmale, Organisation und Potentiale. In A. Hermanns and M. Sauter, editors, *Management-Handbuch Electronic Commerce: Grundlagen, Strategien, Praxisbeispiele*. Vahlen, Munich, 1999.
46. I. Sommerville. *Software Engineering*. Addison-Wesley, Harlow, 8th edition, 2006.
47. A. S. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, and U. Yalcinalp. Web services policy 1.5 - framework. W3C Recommendation, http://www.w3.org/TR/ws-policy/ 2007.
48. P. Weill and J. Ross. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Press, Boston, MA, USA, 2004.

49. P. Windley. Governing SOA. InfoWorld.com, Jan 2006.
50. P. Windley. SOA Governance: Rules of the Game. InfoWorld.com, 2006.
51. P. Windley. Teaming up for SOA. InfoWorld.com, 2007.
52. B. Woolf. Introduction to SOA Governance. IBM developerWorks, 2007. http://www.ibm.com/developerworks/library/ar-servgov/.