

Service Network Modeling: A Generic Approach

Aneta Kabzeva^(✉), Joachim Götze, and Paul Müller

Integrated Communication Systems Lab (ICSY), University of Kaiserslautern,
Kaiserslautern, Germany
{kabzeva,j_goetze,pmueller}@informatik.uni-kl.de

Abstract. With the broad adoption of service-orientation for the realization of business applications and their provisioning and usage over open cloud infrastructures, the topology of the resulting service networks is becoming extremely complex. Due to the composition of services for value-added business capabilities and the reusability of a service in multiple compositions, the execution of one service often depends on other services and changes in its provisioning can affect the health of large parts of the service network. The lack of insight on the relationships between the network components makes the management of the service network's health and change impact hard and error prone tasks. This paper proposes a service network modeling approach for capturing the topology of a service network at design time. The model is used to validate the health of the service network to ensure the operability of its services. At run time the model can be applied for analyzing the effects of evolutionary events such as service modification or withdrawal. Our major contributions are a generic and adaptable modeling structure and a classification of service network entities and relationships. The applicability of our approach is demonstrated on an example service network scenario.

Keywords: Service network · Service network model · Adaptability · Service network health analysis

1 Introduction

With the adoption of service-orientation as paradigm for the development of business applications and the introduction of cloud infrastructures, the organization of today's IT landscapes is changing. Organizations are focusing on their core competency. Supporting non-core functionality is consumed on demand as third-party services in the cloud [16]. The resulting sophisticated relations between services of different vendors and consumers connect the IT landscape in a complex value co-creation service network [15].

Since the structure of a network always affects its functionality [23], the modeling of service networks as an instrument for understanding the complex interactions between the actors and the resources in a service network is an emerging research field in service science [22]. Based on a comprehensive service

network model capturing the relationships between the service network components, analysis approaches can be used to study and optimize the value creation in the underlying service network [6]. However, due to the maturity level of the modeled service network and the different perspectives of the heterogeneous network actors, the scope of the entities relevant for a service network model can differ. The semantic of the relationships between the entities can also differ widely according to the role and the analysis needs of the actor [14].

To provide a model able to support these different service networks and the needs of the various stakeholder roles, the underlying modeling approach needs to (i) be *adaptable* to the heterogeneous modeling concepts for service network components, (ii) be *applicable* for diverse analysis scenarios, while (iii) providing a *uniform* representation of the core modeling concepts. Hence, while allowing to capture private and public service networks with all relevant entities and multiple relationships with different semantics between them, the approach should be able to present different aspects of the model for each network and each stakeholder to address their different requirements and expectations.

The DAME solution proposed in this paper is a metamodel approach for graph-based network models comprised of entities and relationships of customizable types and extensible set of properties. This abstraction provides flexible adaptation of the models and their applicability for heterogeneous networks.

The remainder of the paper is structured as follows: Sect. 2 introduces a motivation scenario for our approach. Related work on service network modeling is addressed in Sect. 3. The modular structure of our approach is described in Sect. 4. Following this structure, Sect. 5 presents a classification of service network entities and relationships identified as initial preset for modeling service networks. The applicability of the approach for impact analysis based on the selected scenario are discussed in Sect. 6. A summary and an outlook to future work in Sect. 7 conclude the paper.

2 Motivation Scenario

The following scenario presents a simplified service network. It presumes a platform-as-a-service (PaaS) cloud infrastructure where actors (A) interested in a collaboration can publish or consume services (S). Figure 1 shows a segment of the network with the interactions of three actors: the private person *John Doe* interested in a *Credit Card Payment* capability (P), which is provided by the *Credit Card Management* service of *Smart Billing Ltd.*, and *Safe Shopping Inc.* providing a *Shipping Management* and *Product Order Management* services on the platform. *Safe Shopping Inc.* acts further as a consumer within the service network, since it uses its own *Product Order Management* service for the realization of its *Product Delivery* capability. The *Product Order Management* service on its side is a value-added service combining the functionality of the *Credit Card Management* service and the *Shipping management* service to check the validity of a card before shipping the goods listed in an order.

In this scenario a number of properties of the network participants and a variety of relationships between them can be observed. Information on properties like

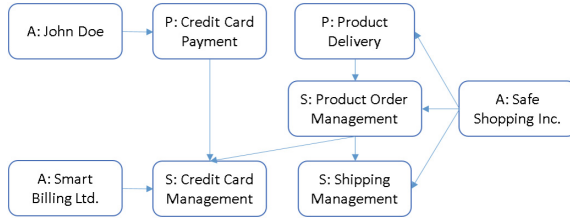


Fig. 1. A service network involving multiple services

the different roles of the actors or the complexity of the provided services is valuable for estimating the productivity of the service network. Together with the non-functional ownership of processes or services we observe functional dependencies on existing services for providing the operability of a composite service, or resource dependencies between the participants in a composition (e.g., Shipping Management awaits an acknowledgment on the validity of a credit card from Credit Card Management to start processing). Already in this small scenario one can see that making the existing relationships explicitly visible will ease the analysis on the service network's health and the management of evolutionary changes. The information could be applied for estimating the area of effect of the different participants and for the identification of hot spots within the topology.

3 Related Work

Numerous research works are focusing on the definition of service network models covering a wide scope of application scenarios. Some of them apply the models for value flow calculation [2, 11], others for the derivation of possible service compositions [5], or providing an overview on business relationships [8]. Cause-effect correlations analysis and service market share [7], service performance tracing [24], and documentation of collaborations for risk-aware management [15, 20] are also considered as motivation for the design of service network models. Depending on the application domain for a model, the scope of the considered model components and the nature of the relationships between them vary greatly.

The value-based model in [2] remains completely on the business entities level of service networks, modeling services, knowledge, and intangible value exchange between organizations. Business tasks are added to the business entities level in the models in [8, 11]. The relationships in [11] remain on the value flow level, while [8] focuses on the ownership of offerings and requests, and the provisioning relations between them.

The model in [7] is based on rich relationships with multiple properties referred to as multi-layer relationships. A multi-layer relationship connects only services and is defined along six dimensions: role, level, involvement, comparison, association, and causality. Dependencies between services are also the focus of

the model in [20]. Here however, the service level agreements of a service are separately modeled and build their own dependency topology.

The approach in [5] proposes a formalism for modeling service networks as demand-driven, ad-hoc interplay of providers and their services. Each model targets the satisfaction of one complex service request and describes all feasible combinations of services that may satisfy the request. The best alternative can be selected by calculating the minimal costs noted on the relationship edges. The scope of our work is different, in that we support the modeling of existing relationships rather than possible ones. Existing collaboration relationships and competition relationships between services are considered in [15]. Provisioning and consumption relationships to the business entities are also defined as part of the approach. However, compared to our approach no possibility for adaptation or weighting of the relationships is considered. The modeling concept in [24] is the only one considering all three abstraction levels of service networks including the business entities, the process, and the service level. However, the approach does not specify concrete relationships between the layer participants. The information on customer involvement, participant interaction, granularity, human operation involvement, software service involvement, and dependency are modeled as part of a service-centered assessment.

4 Generic Modeling Approach for Service Networks

The foundation of the DAME service network management approach is the conceptual service network model. A service network model comprises a number of different entities and relationships between them. A conceptual model defines the types of entities and the types of relationships between them that have to be captured to provide an efficient support for service network analysis. A formal representation of the conceptual service network model can be used then to derive processes for model retrieval, validation, and scoping, and to develop a framework implementing these processes.

The modeling language is defined according to the metamodeling principles of linguistic and ontological metamodeling [4]. As illustrated in Fig. 2, it comprises three ontological metalayers (O_i) which avail themselves of the linguistic constructs (L_j) of XML and XML Schema: the metamodel layer O_2 defines the language for the specification of service network models; the model layer O_1 defines the specification concepts for service network models conform to the language from the upper layer; the instances layer O_0 contains concrete models of service networks using the concepts from the upper layer.

Inspired by [1], the DAME conceptual model consists of two modules – the *DAME Schema* and the *DAME Preset* (cf. Fig. 2). The *Service Network Metamodel* of the schema provides the basic language elements for the description of service network models. The *Model Base* of the schema defines the representation of the elements needed for the specification of a service network model conform to the metamodel. The *Core Concepts* of the preset contribute to the representation, capturing, and analysis of service network models through the

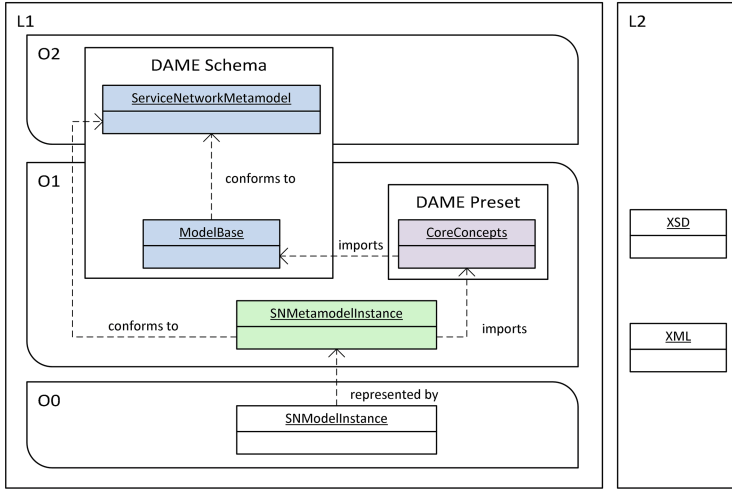


Fig. 2. An overview of the model components

definition of entities and relationships identified as reasonable for the structuring of service network models, their extraction from established entity modeling notations and analysis functions respectively.

Via the modular structure of the approach the uniformity and adaptability properties of the solution are supported. Providing generic metaconcepts for the service network structures, the schema facilitates the dynamic addition of new types of entities and relationships, while the definition of universal and necessary concepts for modeling relationship information in the preset allows for uniform representation. Addressing the required applicability property for the solution, a subset of the core concepts relevant for the needs of the modeled environment can be selected.

4.1 Service Network Metamodel

The Service Network Metamodel defines the language elements for specifying service network models (cf. Fig. 3). The root element `SNModelDefinition` describes the data types available for the description of a service network model.

An *Artifact Type* is a generalization for the types of entities and relationships comprising the service network model. Each artifact type is identifiable by a unique name. It should be well-chosen and describe the nature and the purpose of the artifact in the network.

An *Entity Type* is a specialization of artifact type comprising entities with common properties. The definition for entity types is kept generic. It does not require any mandatory information on the properties of an entity type.

A *Relationship Type* is correspondingly a specialized artifact type for the description of relations with common properties. Each relationship type connects a source entity type with a target entity type. With the provisioning of

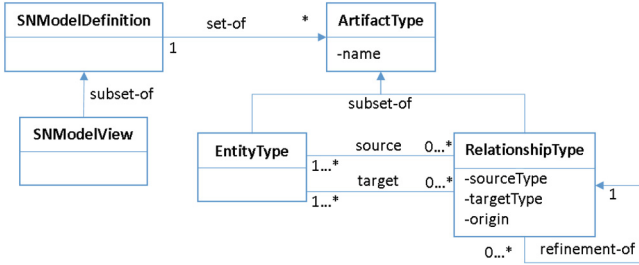


Fig. 3. The Service Network Metamodel

attributes for the specification of permitted source and target types, the meta-model allows the restriction of allowed entities for a specific relationship on the model level O1. An additional origin property is included to indicate if the relationship information was introduced manually in the model or extracted through automated processing of artifact descriptions.

Additionally, the metamodel introduces the concept of views. A view is a specification of a service network model definition with a restricted subset of entity types and relationship types, allowing a simplified representation of a model for specific analysis purposes [9].

4.2 Model Base

This part of the model instantiates the Service Network Metamodel and defines the structure of the elements in a service network model. Figure 4 provides an overview on the elements.

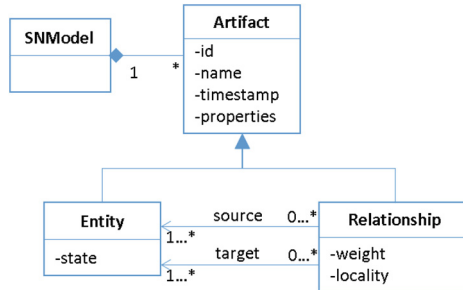


Fig. 4. The model base

An *Artifact* is an abstract element representing the common features of entities and relationships in the model. Additionally to its unique *id* within the model and a *name*, each artifact should save a *timestamp* indicating the insertion of the artifact into the service network model in order to be able to trace

the evolution of the network over time. The *properties* attribute provides an extensible storage structure for artifact-specific properties.

An *Entity* is an instantiation of an entity type used to represent the different members of a service network. The generic representation of an entity inherits all properties of an artifact. Entity-specific fragments can be included in the description via the properties attribute. Additionally, an entity entry is defined by its *state* providing information on the operability of the entity, which depends on the validity of its interconnection within the network.

A *Relationship* is an instantiation of a relationship type used to represent relations between entities. It references the unique identifiers of the two entities which it connects. Additionally to the properties inherited from artifact, a relationship is specified by two additional attributes. The *weight* attribute allows the specification of the occurrence frequency of links that can be caused by the reusability of resources in a service network [10]. Considering the third-party consumption of resources as one of the main characteristics of service networks [17], the value of the *locality* attribute is applied as indicator for relationships crossing domain or even network borders.

4.3 Core Concepts

The preset part of the model comprises an initial information model for the uniform representation of service network models, extraction concepts for their capturing from established modeling languages in the context of service orientation, and analysis concepts providing help functions for the calculation of diverse service network health factors. The structure of the initial information model is discussed in the next section. The initial extraction concepts for BPMN, BPEL, and WSDL and the analysis functions considering the segmentation of the model in different views on the network level and single actor level are work in progress and not part of this paper.

5 Core Representation Concepts

The information model defined by the DAME preset comprises three types of entities and eight types of relationships. The selection of the entity types mirrors the socio-technical character of service networks [17]. The process of emergence of service networks from the composition of third-party services and the risks involved have served as inspiration for the relationship type identification.

Entity Types. With the erosion of the people/system boundary and the new paradigms for simultaneous acquisition and operation of a services [17], the development, operation, and evolution of service network are a cooperative initiative of multiple actors. Thus, the set of entities carving the landscape has to include not only the services responsible for the automation of capabilities, but also the

human actors and the processes defining the requirements on services and inducing their collaboration [21]. Therefore, the preset defines an initial set \mathcal{E} of entity types including *actors*, *processes*, and *services*.

An *actor* \mathcal{A} indicates a physical entity supporting the evolution of a service network through the provisioning and/or consumption of resources. Thus, each network actor should own at least one process or provide at least one service. Each actor is characterized with two properties. The *role* property is used to specify if an actor participates in the network as a provider, a consumer, or both. A *type* property can be specified to indicate the nature of the actor [8], i.e., a private human or a business organization.

A *process* \mathcal{P} is a sequence of activities performed together to realize a business capability [3]. Thus, a process entity represents the non-executable description of a business capability. Each process belongs to exactly one actor. One or more services are responsible for the realization of a process, and thus define the need for consumption of services and their interlinking into a network. Additionally to its operability *state* inherited from the model schema, a process entity is characterized with two additional properties. Its *version* indicates variations of the same capability. *Type* is used to specify the complexity of the process, i.e., *atomic* or *composite*.

A *service* \mathcal{S} is a technical entity enabling the access to one or more capabilities [18]. A service consists of a service description and an execution component provided by an actor. The service description provides an interface in the form of available operations and contract specification needed for the usage of a service. The service logic implemented by the execution component is a black box for the service network.

A service entity is characterized with five properties. The *state*, *version*, and *type* attributes correspond to the ones defined for process entities. A *life-cycleStage* attribute allows the specification of the development stage of the service, e.g., initiation, design, testing. An *access* property indicates the target audience of a service [13]. A differentiation between domain-internal and domain-external services with respect to access [19] allows for better estimation of the QoS requirements and customization effort on a service.

Relationship Types. Considering the variety of entity types involved in the evolution of service networks, the model can be separated in different abstraction levels. Therefore, relationships can be differentiated into *vertical* relationship connecting entities from different type, and *horizontal* relationships connecting entities from the same type.

The vertical relationships are further classified into realization, ownership, and consumption relationships. *Realization* relationships \mathcal{R}_r indicate the refinement of a source entity by the target entity. The source has to fulfill at least one requirement of the target. This relationship provides the basis for expressing the interconnection between the business level and IT level of the service network.

An *ownership* relationship \mathcal{R}_o is used to express the responsibility of a source entity for the target entity. This type of relationship connects the social level

of the service network with the resources of business and IT level. Thus, the source of an ownership relation is always an actors. The set of target entity types includes the set of processes and services. One actor can own multiple processes and services. Each process and service interconnected in a service network belongs exactly to one actor. This actor is responsible for the provisioning of their specified quality.

A *consumption* relationship \mathcal{R}_u expresses the usage of resources within the network. It does not specify how the source makes use of the target but specifies that the target is of interest for the source. This relationship connects the social network layer with the technical services layer. The information is deduced from the realization relationships of the actor's processes. Each actor can consume multiple services. The more of the actor's processes are realized by a service, the higher the weight of the consumption relationship for the actor.

The horizontal relationships in a service network can be further separated into equivalence and associative relationships. Equivalence relationships connect similar entities as competition or evolution parties. A *competition* relationship \mathcal{R}_c connects entities providing the same communication interface, i.e. services providing the same functionality. The actors owning competing services are correspondingly also competitors. A service can have multiple competitors, but not every service has a competitor, e.g., services providing niche functionality. The strength of a competition relationship between actors providing competing services depends on the number of the competing service pairs they own. Although possible in case of intra-domain redundancies, a competition relationship between services of the same provider will not lead to an actor competition relationship.

An *evolution* relationship \mathcal{R}_e indicates that a source entity is a descendant from the target entity. Thus, it connects two versions of the same entity with a modified interface existing at the same time in the service network. Multiple versions of a process or a service may be present within the network with slightly different scope of the provided capabilities or quality. Yet, an evolution relationship is allowed only between resources provided by the same actor.

Associative relationships include dependency relations between entities from the same type. These are divided into cooperation, flow, and resource relationships. A *cooperation* relationship \mathcal{R}_v expresses the collaboration of entities to provide a more complex capability. The result of a collaboration is a value-added entity combining the capabilities of existing entities. The composition of entities in a complex value-added entity can take place on the process level and on the service level. A composite entity can combine multiple existing entities. A cooperation relationship is included from the composite entity to each of its sub-entities. Via deduction from the cooperation between entities from different providers, cooperation relationships can be included on the social level of the model between all actors participating in the provisioning of the value-added entity.

A *flow* relationship \mathcal{R}_f expresses a temporal dependency from the source entity regarding the provisioning of the target entity. Flow relationships occur between entities participating in the composition of a value-added entity and represent

the control flow defined by the value-added entity. Thus, a flow relation connects either process or service entities. The weight of the relationship between two entities increases with the number of value-added entities defining the same control flow between both entities.

A *resource* relationship \mathcal{R}_d expresses a data dependency from the source entity regarding the execution of the target entity. Resource relationships occur also between the participants of a composition and represent the exchange of data objects provided by the source and further processed by the target. The weight of the relationship relates to the number of objects exchanged between the connected entities.

6 Application of the Model for Impact Analysis Support

To validate the usefulness of the model a demonstration of its applicability for impact analysis support is considered. Impact analysis in general is the assessment process of the possible consequences of pursuing a course of action¹. In the context of service-based applications, impact analysis is focused on determining the scope of modifications needed on the different modeling layers of the application to accomplish a change [12]. Impact analysis for service networks broadens the scope of the process to consider the consequences of a change across all applications referencing the resource under modification.

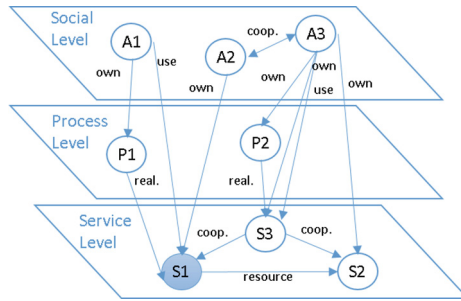


Fig. 5. An exemplary service network model

The **objective** of the impact analysis process is to determine the inter-connectivity of a certain entity within the service network. This is important for identifying hot spots within the service network, i.e., entities that will immensely influence the health of the network in case of modification. In order to determine these entities all incoming and outgoing relationships to all entities have to be considered.

The **problem** of identifying the critical entities in a complex service network landscape is twofold. First, since services do not know for which capabilities they

¹ <http://www.businessdictionary.com/definition/impact-analysis.html>.

are applied in the network. Second, neither services nor processes know in what value-added entities they are composed. This knowledge is only implicitly available in the descriptions of the more complex entities, which also know only their direct successors.

A **solution** for the problem is provided by the capturing and analysis of an integrated service network model following the approach proposed in this paper. Figure 5 shows an example service network model for the motivation scenario from Sect. 2. Analyzing the number and nature of the incoming and outgoing relationships for all eight entities, we identify S1 as the entity with the greatest impact in the network. Following its cooperation and resource relationships on the service level we see that S1 will affect the operability of all other services within the network. Also both processes on the process level will be influenced by a change of S1: P1 via its direct realization relationship to S1 and P2 via its relationship to S3, which is in the impact set of S1.

7 Conclusion and Future Work

The explicit availability of information on the relationships between the artifacts in a service network is important for analyzing the operability state of the service network and for the estimating the impact of changes that can occur during the evolution of the network. As illustrated in this paper, relationships can be observed vertically between the social, process, and services abstraction layers of the service network model, as well as horizontally between entities of the same type. Moreover, relationships can be of different types.

With the purpose to provide a definition of an extensible and uniform service model, this paper proposed a meta-model based modeling approach providing a generic language schema and a preset of entities and relationships following that schema. The applicability of the resulting model is demonstrated for the discovery of critical entities in an impact analysis scenario.

As part of our future work we plan the evaluation of the model with more complex service networks. Additionally, we will provide a set of analysis functions as part of our core concepts preset providing useful calculations on the basis of the initial information model presented here.

References

1. Adersberger, J.: Modellbasierte Extraktion, Repräsentation und Analyse von Treceability-Informationen. Herbert Utz Verlag (2013)
2. Allee, V.: Reconfiguring the value network. *J. Bus. Strategy* **21**(4), 36–39 (2000)
3. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: *Web Services: Concepts, Architectures and Applications*. Springer, Heidelberg (2004). ISBN 3-540-44008-9
4. Atkinson, C., Kuhne, T.: Model-driven development: a metamodeling foundation. *IEEE Softw.* **20**(5), 36–41 (2003)
5. Blau, B., Kramer, J., Conte, T., Van Dinther, C.: Service value networks. In: *IEEE Conference on Commerce and Enterprise Computing 2009. CEC 2009*, pp. 194–201. IEEE (2009)

6. Cardoso, J., Miller, J.A., Bowman, C., Haas, C., Sheth, A.P., Miller, T.W.: Open service network analysis. In: 1st International IFIP Working Conference on Value-Driven Social, pp. 81–88 (2011)
7. Cardoso, J., Pedrinaci, C., De Leenheer, P.: Open semantic service networks: modeling and analysis. In: e Cunha, J.F., Snene, M., Nóvoa, H. (eds.) IESS 2013. LNBIP, vol. 143, pp. 141–154. Springer, Heidelberg (2013)
8. Danylevych, O., Karastoyanova, D., Leymann, F.: Service networks modelling: an SOA & BPM standpoint. *J. Univers. Comput. Sci. (J.UCS)* **16**(13), 1668–1693 (2010)
9. Danylevych, O., Leymann, F., Nikolaou, C.: A framework of views on service networks models. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 21–34. Springer, Heidelberg (2011)
10. Erl, T.: SOA Principles of Service Design. Prentice Hall PTR, Upper Saddle River (2007)
11. Gordijn, J., Akkermans, H., Van Vliet, J.: Designing and evaluating e-business models. *IEEE Intell. Syst.* **16**(4), 11–17 (2001)
12. Hirzalla, M., Zisman, A., Cleland-Huang, J.: Using traceability to support soa impact analysis. In: IEEE World Congress on Services, pp. 145–152. IEEE (2011)
13. Josuttis, N.: SOA in Practice. O’reilly, Sebastopol (2007)
14. Kabzeva, A., Götze, J., Müller, P.: Service network modeling approaches: overview, classification, and analysis. In: Proceedings of the 40th Euromicro SEAA Conference, Verona (2014)
15. Liu, Y., Fan, Y., Huang, K.: Service ecosystem evolution and controlling: a research framework for the effects of dynamic services. In: Proceedings of the International Conference on Service Sciences (2013)
16. Motahari-Nezhad, H.R., Stephenson, B., Singhal, S.: Outsourcing business to cloud computing services: opportunities and challenges. HP Laboratories, HPL-2009-23 (2009). www.hpl.hp.com/techreports/2009/HPL-2009-23.html
17. Northrop, L., Feiler, P., Gabriel, R.P., Goodenough, J., Linger, R., Longstaff, T., Kazman, R., Klein, M., Schmidt, D., Sullivan, K., et al.: Ultra-Large-Scale Systems: The Software Challenge of the Future (2006)
18. OASIS: Reference model for service oriented architecture 1.0, October 2006
19. Repp, N., Schulte, S., Eckert, J., Berbner, R., Steinmetz, R.: Service-inventur: Aufnahme und bewertung eines services-bestands. In: MDD, SOA und IT-Management (MSI 2007) Workshop, Oldenburg, April 2007. p. 13. GITO mbH Verlag (2007)
20. Schulz, F., Caton, S., Michalk, W., Haas, C., Momm, C., Hedwig, M., McCallister, M., Rolli, D.: Integrated modeling of technical and business aspects in service networks. *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, pp. 119–128. Springer, Heidelberg (2013)
21. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., Paige, R.: Large-scale complex it systems. *Commun. ACM* **55**(7), 71–77 (2012)
22. Spohrer, J., Maglio, P.P.: Service science: toward a smarter planet. In: Karwowski, W., Salvendy, G. (eds.) *Introduction to Service Engineering*, pp. 1–30. Wiley, Hoboken (2010)
23. Strogatz, S.H.: Exploring complex networks. *Nature* **410**(6825), 268–276 (2001)
24. Wang, Y., Taher, Y., van den Heuvel, W.J.: Towards smart service networks: an interdisciplinary service assessment metrics. In: Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 94–103. IEEE (2012)