Describing Services for Service Ecosystems

Gregor Scheithauer[1,2], Stefan Augustin[1], and Guido Wirtz[2]

[1] Siemens AG, Corporate Technology, Knowledge Management
Otto-Hahn-Ring 6, 81739 Munich, Germany
[2] University of Bamberg, Distributed and Mobile Systems Group
Feldkirchenstrasse 21, 96047 Bamberg, Germany

Abstract. Service ecosystems are electronic market places and emerge as a result of the shift toward service economies. The aim of service ecosystems is to trade services over the internet. There are still obstacles that impede this new form of market places. Two of these challenges are addressed in this paper: (1) identification of appropriate service properties to specify service descriptions, and (2) a need of a clear classification for service description notations. Therefore, service properties and their relationship are introduced and an adaption for the Zachman Framework is presented to classify service description notations depending on the relative perspective.

Keywords: Service description, Zachman Framework, service ecosystems.

1 Introduction

Tertiarisation describes a structural change in developed countries concerning the sectoral composition. Countries shift from an industry economy toward a service economy. Sources of this change include globalization, technological change, and an increasing demand for services [21]. Considering this trend, it becomes clear that services and the service economy play an important role in today's and tomorrow's business. In line with this trend, service ecosystems [4] emerge, such as eBay, Google Base, Amazon.com, SalesForce.com, and SAP Business by Design. Such market places allow to trade services between different legal bodies.

One major challenge for service ecosystems is the fact that services are different to goods. According to Booms and Bitner [5] services are intangible, and thus, can neither be stored, transported, nor resold. Goods are produced at some point, *stored*, and eventually consumed. In contrast, production and consumption of services take place at the same time. Goods can be *transported* from one point to another. Services, on the other hand, are consumed at customers' location, thus, production and consumption happen in one place. Whereas goods can be *resold*, services' outcome cannot be sold to another party. Additionally, services can hardly be *standardized*, since service experience is unique and depends on individual expectations. Moreover, no established language exists to define, agreeing on, and to monitor service properties [12]. For service ecosystems, service descriptions abstract from concrete services and provide a tangible artifact,

which can be stored and transported, and therefore relax some of Booms and Bitner's arguments.

Another challenge is that notations for service descriptions depend on perspective. Service descriptions on a business operational perspective and a technology perspective differ in concepts and semantics. A system to categorize different service description notations and realization languages would support a common understanding. Service engineering would benefit from such a classification, since it comprises different stakeholders and different phases such as innovation, strategic planning, service design, service implementation, and market launch [11]. It would support the work to derive service description from service provider's business models, and to implement them with web service related technology.

This paper presents (1) service properties and their relationship to specify a service description from a service provider's viewpoint, and (2) an adaptation for the Zachman Framework [31] in order to categorize service description notations for various perspectives.

The remainder of this paper is structured as follows: section two introduces service ecosystems. The Zachman Framework is summarized in section three. The service properties and their relationships are explained in section four, and section five presents the service description adaptation for the Zachman Framework. Section six concludes this paper. Related work is presented in the relative sections.

2   Service Ecosystems

Web services [34] for heterogeneous application integration and communication between companies gained popularity during the last years [27]. Recently, companies, such as Amazon.com, acknowledged web services beyond integration as a means to create value for customers. In consequence, the service ecosystem concept gained momentum. Since it is a fairly new field of research, various names exist for service ecosystems, which include service systems [32] and internet of services [25]. The vision of service ecosystems is an evolution of service orientation and takes services from merely integration purposes to the next level by making them available as tradable products on service delivery platforms [4].

Service ecosystems are market places for trading services in the business sense and involve actors from different legal bodies. Service trade involves the following steps: service discovery, service selection, service contracting, service consumption, monitoring, and profiling. During discovery and selection, service providers advertise their services toward potential consumers, whereas service consumers specify their service preferences toward providers. During service contracting, providers and consumers negotiate and finally agree on service levels (SLA) which are monitored (for billing & payment) throughout service consumption. In the event service levels are not met, compensations must be triggered. During service profiling, valuable information on services' performance is stored, which is gathered during consumption and monitoring.

On the technical side, service ecosystems refer to a logical web service collection [4], with more than one service provider. On the business side, service
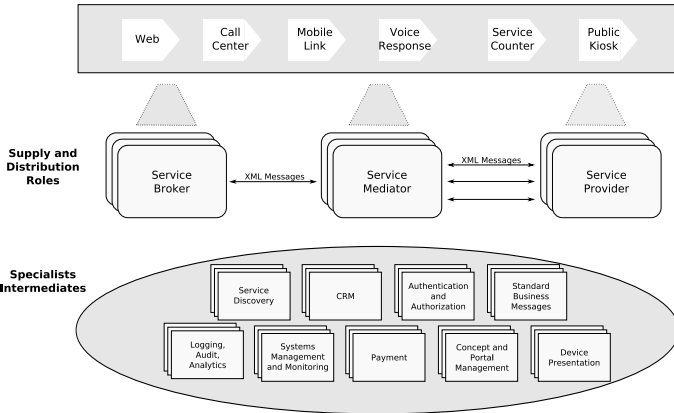
Fig. 1. Top-level Architecture of a Service Ecosystem [4]

ecosystems bring together shared information, people, and technology [32]. They comprise service innovation, service design, service engineering, service marketing, and service provisioning [11]. These systems provide core services such as payment and monitoring, domain-specific services such as eco-value calculations [9], and complex services such as travel services. These service are leveraged by others to implement end-to-end business processes [28], which cross companies' borders, to create value for end-customers. Business models such as Business Webs [33] foster this idea of *coopetition*.

Barros and Dumas [4] (cf. figure 1) identify next to service consumers three different roles for actors in service ecosystems. *Service providers*, who provide services in the first place. *Service brokers* offer services from different providers. Their business model is to bring providers and consumers together, or enhance services with delivery functions for convenient service provisioning. *Service mediators* generate value by customizing provider's standard services toward consumer's needs. All the same, service ecosystem actors may play more than one role.

Services that are traded in service ecosystems are so-called *e-services*. Baida et al. [2] offer a terminology comprising the terms service, e-service, and web service. Services refer "...to business activities which often result in intangible outcomes or benefits ..." [2]. E-services, in contrast, refer to service provisioning by means of electronic network protocols, such as the Internet. E-services are technically implemented with web services. For the rest of this paper, the term service refers to the e-service definition.

In conclusion, service ecosystems aim at (1) trading services over the internet, (2) composing complex services from existing ones, and (3) supporting service provisioning with IT [9].

3 Zachman Framework

The Zachman Framework [35] provides a taxonomy to relate real world concepts to Enterprise Architecture [31]. Zachman describes Enterprise Architecture as

means to flexibly react to business changes and to manage the varied resources of an enterprise. The Zachman Framework embodies vital artifacts to describe, create, operate, and change an object. The term "Object" is used consciously, since it may relate to practically anything, e.g., an enterprise, a project, a solution, and in this case a service.

The Zachman Framework distinguish between six perspectives and six descriptions which are orthogonal to each other. The first six columns (internal view) in figure 2 depicts the framework's matrix. Each column of the matrix offers a basic model for the description in question from a certain perspective. It is important to note that the framework does not specify the order of descriptions. Each intersection is a placeholder for a basic notation which satisfies a columns' basic model.

3.1   Perspectives

The six different perspectives are organized into corresponding layers [31]. It is important to note that the various perspectives are different with respect to nature, content, and semantics and not only in their detail level [35].

The *scope layer* represents the planner's perspective. The purpose of this layer is to identify "... the size, shape, spatial relationships, and final purpose of the final structure." [31] and thus, the scope. On this basis a planner decides whether to invest in the architecture.

The *business layer* symbolizes the owner's perspective. Architects describe the requirements from the owner's perspective, whereas the intention is to "... enable the owner to agree or disagree with the ..." [35] description.

The *system layer* corresponds to the designer's perspective. The purpose of this layer is to transform the enterprise model's artifacts into detailed specifications. The owner can use these specifications to negotiate with builders to implement the system.

The *technology layer* represents the builder's perspective. The rationale of this layer is that the detailed specifications must be adapted into builder's plans to take into account the "... constraints, of tools, technology, and materials." [31].

The *component layer* symbolizes the perspective of a sub-contractor. Builder's plans are translated into shop plans. Shop plans "... specify details of parts or subsections ..." [31] of builder's plans.

The *operations layer* represents the system itself.

3.2   Descriptions

The six descriptions depict an enterprise from different angles. Though, each of them is unique and addresses a different purpose, they relate to each other [35]. Descriptions are the answers to the basic questions: What (Data Description), How (Process Description), Where (Location Description), Who (People Description), When (Time Description), and Why (Motivation Description). It is important to note, that for each description exists a set of terms (description model) which are valid for all perspectives. Nonetheless, these terms differ essentially in semantics for each perspective.

The *data description*'s model consists of entities and relationships between entities. The basic intention is to identify enterprises' inventory.

The *process description*'s model embodies processes and arguments (input and output to processes). The purpose is to make out enterprises' processes and business functions.

The *location description*'s model uses the concepts of locations and connections in order to discover enterprises' network.

The *people description*'s model is that of roles and work. The description's intention is the "... allocation of work and the structure of authority and responsibility." [31].

The *time description*'s model embodies event and cycle. The description's intention is to "... produce a schedule of events and states that maximizes the utilization of available resources while at the same time satisfying the external commitment." [31].

The *motivation description*'s model uses the concepts of ends and means. The motivation description's intention is to describe the motive of an enterprise, where ends equal objectives and means equal strategies.

4   Service Properties

This section introduces service properties to describe services. Initially, Scheithauer and Winkler [26] introduced these properties. They collected properties from different sources: (1) PAS 1018:2002 [14], (2) IEEE 830:1998 [29], (3) IEEE 1061:1998 [30], (4) ISO 9126 [8], (5) O'Sullivan's service properties [20], (6) quality attributes [3]. The actual set presents some adapted properties as well as a different and a more coherent property grouping. The taxonomy from Avizienis et al. [1] was added to the analysis for quality of service. Additionally, a hierarchy is introduced which depicts the parent property in combination with cardinalities. Nevertheless, the presented properties are not intended to be complete. They rather show the current state of the work. Once service ecosystems' requirements toward describing services are discovered, an exhaustive analysis is possible. Another limitation is that the presented properties lack appropriate metrics. This results from the fact that depending on the perspective, properties are interpreted differently, e.g., a capability is differently expressed on the scope layer than on the technical layer. This challenge is beyond the scope of this paper.

These properties are valid for all perspectives of the Zachman Framework (cf. section 5). It is important to note that these properties do not intend to describe services' behavior, implementation, nor how to technically integrate a service into various software environments. They rather serve to propose a service on a market place toward potential customers in terms of functionality, financial, legal, marketing, and quality aspects. The following subsections will shortly introduce each property.

Table 1. Functional Service Properties

Property Name	Parent Property	Cardinality
Capability	Service Description	1...*
Classification	Service Description	0...*

4.1 Functionality

Functionality provides the service consumer with an understanding of what the service is actually providing and thus, what the consumer can expect from the service. Properties include capabilities and service classifications (cf. table 1).

A *Capability* is the major function that services provide. A capability allows a service consumer to access services' functionality. Often, services' functionality is divided into several capabilities. This allows service consumers to access particular subsets of services' functionality. Additionally, a service's outcome might be different, depending on capabilities to perform in what order. E.g., a flight booking service offers different capabilities, such as to browse different flights, to plan a flight route, to book a flight, and to pay for it. In some cases just some of these capabilities are necessary for service consumers to achieve their goals. However, to book a flight, one of more specific capabilities must be invoked in a predefined way. A service has one or more capabilities. This property is represented by a formal naming system. This would include a capability's name, involved parties, data which is processed by the service, and the outcome, which address also pre- and postconditions.

A *classification* allows to apply the service into one or more classification systems. A classification is a system of interrelated terms which generally form a hierarchical structure. The terms allow to specify the kind of service, an unique identifier, and a reference to a classification standard, such as eCl@ss and UN-SPSC [7]. While the classification property is optional, it may be the case that a service is classified according to multiple classification standards. For that reason it is necessary to model service classification as a tuple of a reference to a classification standard and a unique identifier.

4.2 Financial

This section comprises monetary related properties (cf. table 2).

The *price* property represents an economical numerical value for services. PAS 1018:2002 [15] and O'Sullivan [20] list this property. PAS 1018:2002 depicts two price properties. The first price property describes service providers' price conception. The second price property specifies the service consumers' price idea. O'Sullivan, however, offers a more holistic approach. His work includes four different types of price. It is possible to relate all price types to entities such as *time*, *area*, etc. This allows to specify different prices for different time or areas of service usage. Additionally, tax information can be included as well. Four price types are explained briefly [20]. An *absolute price* specifies a specific amount of money and a currency. E.g., booking a flight costs EUR 10. A *proportional*

Table 2. Financial Service Properties

Property Name	Parent Property	Cardinality
Absolute Price	Capability	0...*
Ranged Price	Capability	0...*
Proportional Price	Capability	0...*
Dynamic Price	Capability	0...*
Discount	Price	0...*
Early Payment	Discount	0...*
Payment Instrument	Discount	0...*
Coupon	Discount	0...*
Volume	Discount	0...*
Location	Discount	0...*
Age Group	Discount	0...*
Student	Discount	0...*
Membership	Discount	0...*
Shareholder	Discount	0...*
Payment	Service Description	1...*
Payment Option	Payment	1...1
Payment Schedule	Payment	0...*
Card Instrument	Payment	1...*
Cheque Instrument	Payment	1...*
Cash Instrument	Payment	1...*
Voucher Instrument	Payment	1...*

price depicts a percentage with respect to a given value. E.g., a life insurance monthly rate is 1% of one's yearly income. A *ranged price* allows to specify a price range with a minimum and maximum *absolute price* or *proportional price*. Service providers may use this price type in case it is impossible to set an absolute price. Fixing the final price is part of the negotiation phase between service providers and service consumers. E.g., a rental car's price per day ranges from EUR 50 to EUR 70. The final price depends on the final car configuration. A *dynamic price* covers auctions, where the price matching is based on natural supply and demand. E.g., a service provider offers train tickets and potential service consumers bet an amount of money they perceive as their value. The metric for currencies is the ISO 4217:2001. The price amount is represented by a numerical data type.

The *payment* property specifies feasible options to fulfill service consumer's payment liability. PAS 1018:2002 [15] and O'Sullivan [20] list this property. Where PAS 1018:2002 depicts only a placeholder for payment, O'Sullivan offers a more thorough approach. However, they do not contradict each other. According to O'Sullivan [20], payment is complementary to the *price* property. He subdivided this property into four models: payment options, payment schedules, payment instruments, and payment instrument types. A *payment option* constitutes, whether a particular payment option is the preferred one, whether there is a charge connected to the payment option, where a payment option is available, specific conditions for a payment option, and currencies. A *payment schedule* depicts when a

payment is due. This property has two dimensions. Firstly, it is possible to specify a percentage of the whole price with respect to services' provisioning moment (before, during, and after). Secondly, percentages together with concrete dates can be specified. Four *payment instrument types* are available: *card based instruments*, *cheques*, *cash*, and *vouchers*. A service has one or more payment properties. Each payment property has exactly one option, none or more schedules, and at least one instrument. Payment is a mandatory property. Dates are represented with ISO 8601, currencies with ISO 4217, and regions with ISO 3166.

The *discount* property specifies possible price reductions. Only O'Sullivan [20] lists this property. In general, *discount* properties can be offered within a specified time segment (temporal), for a specific location (locative), or a given condition. The discount property is differentiated between payment related discounts and payee related discounts. *Payment related discounts* group types of discounts that refer to how payment is done. This includes *early payment*, *type of payment instrument*, *coupons*, *location of payment*, and *volume invocation*. *Payee related discounts* relates to the service consumer, who pays for a service. This includes *age group*, *student*, *membership*, and *shareholder*. A service has no or more discounts for a *price*. Dates are represented with ISO 8601, and regions with ISO 3166.

4.3 Legal

The legal category embodies properties which state terms of use. The properties are rights, obligations, and penalties (cf. table 3).

Rights state what service consumer are allowed or expected to do with the service. For service ecosystems, re-selling, and re-bundling with other services needs to be considered. Rights are represented with semantically defined terms.

Obligations determine and settle the commitment for a service provider and a service consumer. This includes what a service provider must deliver. Obligations are represented with semantically defined terms.

The *penalty* property dictates compensation in case an obligation was not met by one party. Each obligation property relates to one penalty property to cover the effects. Penalties are represented with semantically defined terms.

Table 3. Legal Service Properties

Property Name	Parent Property	Cardinality
Right	Service Description	0...*
Obligation	Service Description	0...*
Penalty	Obligation	1...1

4.4 Marketing

The marketing category allows to promote the service toward potential customers. Properties in this category should both attract customers and establish

Table 4. Marketing Service Properties

Property Name	Parent Property	Cardinality
Certification	Service Description	0...*
Expert Test Rating	Service Description	0...*
Benefit	Service Description	0...*

a trusted relationship. A certification would provide a rather neutral view on a service provided by a third party. On the other hand, expert test ratings provide a subjective view on the service from an expert perspective. Service benefits are the gained outcome of the service with respect to the potential service consumer. These properties are summarized in table 4.

The *certification* property represents a declaration issued by trusted institutes or by the service platform itself. This property tells whether a service is certified by a known and trusted party. This party issues a certificate in case one or more requirements regarding services are met. An analogous concept is the certification for secure websites. The certificate is represented with a formal system or a common standard, such as the *X.509*.

Expert test rating represents a rating from autonomous parties which are experts in the service domain. Potential service consumers might consult the expert test rating to decide whether to use the service or not. The expert test rating is determined by thorough tests, where domain-specific criteria are applied to services and then, depending on the performance, are rated. This property may be represented via a scale of values ranging from a minimum to a maximum value (e.g., scale from 1 to 10 as described before).

The *benefit* of a service is the gained outcome of the service for the service user. This information is needed for a potential service consumer to determine whether this particular service has the potential to suit its needs.

4.5   Quality of Service

As aforementioned, service provisioning in service ecosystems is conducted over the internet, technical properties of the network and the service itself can be of importance for service discovery and selection. Properties include: (1) performance and (2) dependability (cf. table 5).

Performance represents a service's responsiveness with respect to events and time [3]. Performance is expressed with *latency* and *throughput*. Latency describes a fixed time interval in which an expected event's response must be fired. Throughput specifies how many responses to a given event during a time interval will be completed.

Avizienis et al. define [1] dependability as "...the ability to deliver service that can justifiably be trusted." For this analysis, *availability* and *reliability* are regarded. A service's availability represents the systems readiness to be triggered [3]. Reliability, on the other hand, express the service's capability to keep performing over time [3]. Both, availability and reliability are expressed with a percentage.

Table 5. Quality Service Properties

Property Name	Parent Property	Cardinality
Latency	Service Description	0...1
Throughput	Service Description	0...1
Availability	Service Description	0...1
Reliability	Service Description	0...1

5 Service Description for the Zachman Framework

This section presents a seventh description for the Zachman Framework to create a classification for service description notations and realization languages. The general logic for *perspectives* (cf. section 3) is used as the basis.

The existing six descriptions in the Zachman Framework address the solution itself, e.g., an enterprise architecture or a service. These descriptions can be considered as an *internal view* on the solution. Considering service ecosystems as market places to trade services, an internal view is not appropriate for two reasons: (1) service customers might not be interested in how services work or are implemented and (2) service providers do not want to expose this information to customers and competitors. Thus, a description must become available which depicts the *external view* of the solution, which promotes the service toward potential customers. This *external view* has different semantics depending on the different perspectives during service engineering.

Hence, the Zachman Framework is used to align the service description notations and realization languages with the six different perspectives. This allows to find appropriate service description notations and realization languages for each perspective and to identify the need for further notations. Furthermore, this classification supports the arrangement of the perspectives and their appropriate notations to foster business-IT alignment.

One limitation to this approach is that this adaptation works for service ecosystems. In other environments than service ecosystems, different requirements might apply, and therefore other adaptations than the one presented become necessary.

It is important to note that the service properties introduced in section 4, are valid for all perspectives of the Zachman Framework.

The *service description*'s model is that of properties and values. The service description intention is to describe a service's proposition [6] a company offers toward its customers. *Properties* refer to service elements which describe a certain aspect of services. *Values* refer to the characteristic of service elements. This model is valid for each perspective, though the semantic differs for each of them. Figure 2 depicts the resulting framework.

On the *scope layer* properties have a strategic semantic and take into account the service final purpose and context by listing the important properties. For example, the price property refers to a price strategy, and price values refer to a specific strategy, such as Porter's generic price strategies. Suitable models for this perspective are the business model ontology [19,25] and the service bundle [22].

	Internal View						External View
	Data	**Process**	**Location**	**People**	**Time**	**Motivation**	**Service**
Scope Layer	List of important Entities	List of important Processes	List of important Locations	List of important Organizations	List of important Events	List of important Goals and Strategies	List of important Properties
Business Layer	Semantic Model	Business Process Model	Business Logistic Model	Workflow Model	Master Schedule	Business Plan	Value Proposition
System Layer	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model	Service Model
Technology Layer	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	Service Profile
Component Layer	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification	Service Component
Operations Layer	DATA	FUNCTION	NETWORK	ORGA-NIZATION	SCHEDULE	STRATEGY	SERVICE

Fig. 2. Service Adaptation for the Zachman Framework

Properties and values on the *business layer* describe the owner's requirements with respect to the services. The result is a value proposition toward potential customers. For example, the price property refers to a price model, and the price value to a specific price model, such as Proportional Price [20].

On the *system layer* designers create a complete service model, which is technology-independent and formal. On this formal basis, builders can implement a platform-dependent service description. For example, the price property refers to the obligation a potential service consumer must pay in order to consume the service and price value specifies the amount of money. Suitable model notations for this perspective are the UML Profile and Metamodel for Services (UPMS) [17], the UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (UPMQoS) [16], as well as the Service Component Architecture (SCA) [18].

Properties and values on the *technology layer* are adapted to a concrete technology. Appropriate technologies for web services include the Web Service Modeling Ontology (WSMO) [24] and Web Ontology for Services (OWL-S) [13].

On the *component layer* the service description is divided into sections. This parting allows to realize different parts independently. For example functionality properties and values are presented with a WSDL file [23], and quality of service properties and values with the web service level agreement (WSLA) [10].

The *operation layer* represents the implemented service description itself.

6 Conclusion and Future Work

First, service properties were presented to describe services' external view. Second, an adaptation for the Zachman Framework was motivated and introduced. Lastly, notations were selected for each perspective. Service providers are now in the position to categorize different service description notations with respect to the various perspectives which are involved during service engineering. Nonetheless, nothing is said about method. A perspective's service description can be either modeled *before* the other descriptions as a requirement, or *after* the other descriptions. The service properties offer attributes to describe services in a common way and hence narrow the gap between business model's value propositions and service description implementations. It serves as a domain-specific taxonomy for describing services for service ecosystems.

Future work includes the evaluation of the service properties. Additionally, requirements for service descriptions must be derived from service ecosystems to improve and complete the service property set. Furthermore, the service properties will be codified in a formal language, in order to share this knowledge between service providers and service consumers. The Zachman Framework adaptation is already in place with the Inter-enterprise Service Engineering (ISE) methodology [11]. Tool support is under development. Finally, a routine needs to be developed to model and to derive service properties from business model's value propositions [25].

Acknowledgments

References

1. Avizienis, A., Laprie, J.-C., Randell, B.: Dependability and its threats - A taxonomy. In: Jacquart, R. (ed.) IFIP Congress Topical Sessions, pp. 91–120. Kluwer, Dordrecht (2004)
2. Baida, Z., Gordijn, J., Omelayenko, B.: A shared Service Terminology for Online Service Provisioning. In: Janssen, M., Sol, H.G., Wagenaar, R.W. (eds.) ICEC. ACM International Conference Proceeding Series, vol. 60, pp. 1–10. ACM, New York (2004)
3. Barbacci, M., Klein, M.H., Longstaff, T.A., Weinstock, C.B.: Quality attributes. Technical Report ESC-TR-95-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (December 1995)
4. Barros, A.P., Dumas, M.: The Rise of Web Service Ecosystems. IT Professional 8(5), 31–37 (2006)

5. Booms, B., Bitner, M.: Marketing strategies and organization structures for service firms. In: Marketing of Services, American Marketing Association, Chicago, IL, pp. 47–51 (1981)
6. Gordijn, J., Petit, M., Wieringa, R.: Understanding business strategies of networked value constellations using goal- and value modeling. In: RE, pp. 126–135. IEEE Computer Society, Los Alamitos (2006)
7. Hepp, M., Leukel, J., Schmitz, V.: A quantitative analysis of product categorization standards: content, coverage, and maintenance of ecl@ss, UNSPSC, eOTD, and the rosettanet technical dictionary. Knowl. Inf. Syst. 13(1), 77–114 (2007)
8. International Organization for Standardization (ISO). ISO/IEC 9126: Software engineering - Product quality (2001)
9. Janiesch, C., Ruggaber, R., Sure, Y.: Eine Infrastruktur fuer das Internet der Dienste. HMD - Praxis der Wirtschaftsinformatik 45(261), 71–79 (2008)
10. Keller, A., Ludwig, H.: The WSLA framework: Specifying and monitoring service level agreements for web services. J. Network Syst. Manage 11(1) (2003)
11. Kett, H., Voigt, K., Scheithauer, G., Cardoso, J.: Service Engineering for Business Service Ecosystems. In: Proceedings of the XVIII. International RESER Conference, Stuttgart, Germany (September 2008)
12. Kuropka, D., Troeger, P., Staab, S., Weske, M. (eds.): Semantic Service Provisioning. Springer, Heidelberg (2008)
13. Martin, D., Paolucci, M., McIlraith, S.A., Burstein, M., McDermott, D., McGuinness, D.L., Parsia, B., Payne, T.R., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K.P.: Bringing semantics to web services: The OWL-S approach. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 26–42. Springer, Heidelberg (2005)
14. Mörschel, I., Behrens, H., Fähnrich, K.-P., Elze, R.: Standardisation in the Service Sector for Global Markets. In: Advances in Services Innovations, Engineering, pp. 257–277. Springer, Heidelberg (2007)
15. Mörschel, I.C., Höck, H.: Grundstruktur für die Beschreibung von Dienstleistungen in der Ausschreibungsphase. Beuth Verlag GmbH, Ref. Nr. PAS1018:2002-12 (2001)
16. Object Management Group (OMG). Specification: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, v 1.1 (May 2005), http://www.omg.org/technology/documents/formal/QoS_FT.htm
17. Object Management Group (OMG). Specification: UML Profile and Metamodel for Services RFP UPMS Services Metamodel (September 2006), http://www.omg.org/cgi-bin/doc?soa/2006-09-09
18. Organization for the Advancement of Structured Information Standards (OASIS). Specification: Service Component Architecture (SCA), http://www.oasis-opencsa.org/sca
19. Osterwalder, A.: The Business Model Ontology: A Proposition in a Design Science Approach. PhD thesis, Universite de Lausanne Ecole des Hautes Etudes Commerciales (2004)
20. O'Sullivan, J.: Towards a Precise Understanding of Service Properties. PhD thesis, Queensland University of Technology (2006)
21. Peneder, M., Kaniovski, S., Dachs, B.: What follows tertiarisation? structural change and the role of knowledge-based services. The Service Industries Journal 23(2(146)), 47–66 (2003)
22. Pijpers, V., Gordijn, J.: Bridging Business Value Models and Process Models in Aviation Value Webs via Possession Rights. In: HICSS, p. 175. IEEE Computer Society, Los Alamitos (2007)

23. Chinnici, A.R.S.W.R., Moreau, J.-J.: Specification: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation, 6 (2007)
24. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web Service Modeling Ontology. Applied Ontology 1(1), 77–106 (2005)
25. Scheithauer, G.: Process-oriented Requirement Modeling for the Internet of Services. In: Ruggaber, R. (ed.) Proceedings of the 1st Internet of Services Doctoral Symposium 2008 (I-ESA), Berlin, Germany, vol. 374, March 25 (2008)
26. Scheithauer, G., Winkler, M.: A Service Description Framework for Service Ecosystems. Bamberger Beiträge zur Wirtschaftsinformatik 78, Bamberg University (October 2008) ISSN 0937-3349
27. Scheithauer, G., Wirtz, G.: Applying Business Process Management Systems – a Case Study. In: The 2008 International Conference on Software Engineering and Knowledge Engineering (SEKE 2008), Redwood City, California, USA, pp. 12–15 (2008)
28. Scheithauer, G., Wirtz, G., Toklu, C.: Bridging the Semantic Gap between Process Documentation and Process Execution. In: The 2008 International Conference on Software Engineering and Knowledge Engineering (SEKE 2008), Redwood City, California, USA (2008)
29. Software Engineering Standards Committee of the IEEE Computer Society USA. IEEE Guide for Software Requirements Specifications, 830-1998 (1998)
30. Software Engineering Standards Committee of the IEEE Computer Society USA. IEEE Standard for a Software Quality Metrics Methodology, 1061-1998 (1998)
31. Sowa, J.F., Zachman, J.A.: Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal 31(3), 590–616 (1992)
32. Spohrer, J., Maglio, P.P., Bailey, J., Gruhl, D.: Steps toward a science of service systems. IEEE Computer 40(1), 71–77 (2007)
33. Tapscott, D., Ticoll, D., Lowy, A.: Digital capital: harnessing the power of business Webs. Harvard Business School Press (May 2000)
34. W3C Working Group. Web services glossary (February 2004),
    http://www.w3.org/TR/ws-gloss/
35. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3), 276–292 (1987)