Analyse von Service-orientierten Architekturen aus Business- und IT-Sicht

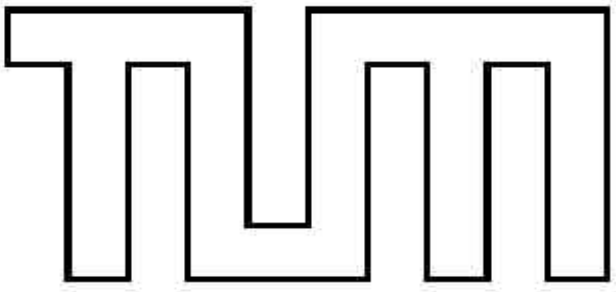
-

Analysis of Service-Oriented Architectures from a business and an IT perspective

Alexander Sterff

Master's Thesis in Wirtschaftsinformatik

Technische Universität München



Fakultät für Informatik
Lehrstuhl XIX
Software Engineering betrieblicher Informationssysteme
Prof. Dr. Florian Matthes

Master's Thesis in Wirtschaftsinformatik

Analyse von Service-orientierten Architekturen aus Business- und IT-Sicht

-

Analysis of Service-Oriented Architectures from a business and an IT perspective

Alexander Sterff

Aufgabensteller:	Prof. Dr. Florian Matthes
Betreuer:	Alexander M. Ernst, M.Sc.
Abgabedatum:	15. Dezember 2006

Ich versichere, dass ich diese Master's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.
Garching, den 15. Dezember 2006

Alexander Sterff

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Structure of this Thesis 4

2 Basic SOA **6**

2.1 Underlying concepts 7

2.1.1 Services 10

2.1.2 Service description 11

2.1.3 Service communication 13

2.1.4 Service repository 13

2.1.5 Additional elements 15

2.2 SOA characteristics 15

2.2.1 Loose coupling 16

2.2.2 Autonomy of services 17

2.2.3 Abstraction 18

2.2.4 Reusability and composability 18

2.2.5 Discoverability 19

2.3 Expected benefits 20

2.3.1 Agility and Flexibility 20

2.3.2 Technology independence 21
2.3.3 Cost savings 22
2.3.4 Reuse 23
2.3.5 Reduced risk 23
2.3.6 Business orientation 24
2.4 Topics of an SOA 24
2.4.1 Granularity of services 25
2.4.2 Orchestration 25
2.4.3 Choreography 26
2.4.4 Service layer 28
2.4.5 Service bus 29
2.4.6 Governance 30
2.4.7 Business Process Management 31
2.4.8 Web services 31
2.5 Comparison with other concepts 32
2.5.1 Client-Server architecture 33
2.5.2 Component based development 33
2.5.3 Object orientation 34
2.5.4 Enterprise Application Integration 35

3 Analysis **36**
3.1 Approach of analysis 37
3.1.1 Stepwise procedure 37
3.1.2 Graph generation 40
3.1.3 Statistics 40
3.2 General results 41
3.2.1 Disposability of articles 42

3.2.2 Distribution in time 44
3.2.3 Authors 46
3.2.4 Reference origin 48
3.2.5 Frequency of referencing 50
3.2.6 Referenced specifications 52
3.2.7 Visualization of references 52
3.3 Topics of the articles 57
3.3.1 Business related keywords 57
3.3.2 IT related keywords 61

4 Enterprise SOA **66**
4.1 Business and IT: SOA as a mediator 67
4.2 SOA and Business 70
4.2.1 Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals . . 72
4.2.2 A new service-oriented architecture Maturity Model . 77
4.2.3 On business value and technical challenges of adopting Web services 81
4.3 SOA and IT 86
4.3.1 Service-Oriented Computing: Key Concepts and Principles 87
4.3.2 Service-Oriented Computing: Concepts, Characteristics and Directions 90
4.3.3 Using Architectural Patterns and Blueprints for SOA 95

5 Conclusion **99**

A Results of analysis **140**

B Complete graph **165**

Chapter 1

Introduction

The impact of Information Technology (IT) within an enterprise has undergone many changes over the years. In its early days in the 1960s IT was merely used to support some specific processes within an enterprise. Until the end of the 20th century IT was commonly used for automation of processes and helped enterprises to increase their efficiency [He05c]. Since then, IT more and more plays an important role within companies. It is an enabler for new technology based business models and has become a crucial factor for the success and the survival of business. IT has to be aligned with the business strategy of an enterprise on one side but also acts as an enabler for new strategies on the other side [Kr05b]. This leads to a very basic distinction of the attitude towards the IT in a company: While companies with a strong enabling function of the IT, focus on new technologies and put a lot of resources into their IT departments, others using the IT to support their business and increase their efficiency, have a more businesslike view on the IT and consider it as commodity. This dilemma becomes obvious in the article "IT doesn't matter" [Ca03]. Nicholas J. Carr claims, that IT will only be valued by the costs and no competitive advantage can be achieved by the use of IT. Due to this circumstances and the challenges of the competition, it is becoming more and more evident, that enterprises not only have to focus on the technical opportunities of new technology, but also have to value the costs and benefits of the use of IT.
One of the main challenges for modern IT are the fast changing business processes that demand a high level of flexibility. This makes IT a complex discipline, that has to deal with a lot of constraints which have to be fulfilled. Many requirements from various stakeholders have to be consid-

ered when planning and running IT systems. One of the main problems a lot companies face is the lack of alignment to the business. Often the IT departments are detached from the core business processes and therefore fail their responsibilities in supporting the business needs. One promising solution to these problems are the so called Service-Oriented Architectures (SOA). It's supporters claim a better alignment of IT and Business along with many other benefits.

1.1 Motivation

The Chair for Software Engineering for Business Information Systems at the Technische Universität München held by Prof. Dr. Florian Matthes is researching in Softwarecartography a discipline in Enterprise Architecture Management (EAM). EAM is an approach that tries to align business and IT. Softwarecartography is a discipline that describes, evaluates and designs application landscapes to provide an intuitive and perspicuous view to all relevant aspects. These aspects are composed of economical, technical, operational and strategic aspects. Based on this concept, several views on the application landscape are supplied, that allow a transparent and holistic overview over the existing application landscape and enable long-term planning, control and evaluation of the application landscape. The aim of this research project is to collect empirical data from big German companies about the structure and the economic relevance of enterprise information systems. This information should help to verify what concepts and notations are well suited to depict the economic relevance of the enterprise information systems of those enterprises in a way, that they can be easily understood by the respective stakeholders of the application landscape.
Today many companies are confronted with an application landscape that is characterized by a high level of diversity and heterogeneity. Thus making workflows complex and hindering enterprise wide integrated business processes. Enterprise Application Integration (EAI) is one approach to address these problems. But as numerous projects in the past have shown, this just shifts the complexity of the heterogeneous application landscape into the EAI layer, not really solving the fundamental problems [KBS05] (p. 21). The reason for this complexity lays in the different platforms, databases and systems that have evolved over the time. All this impedes efficiency and effectiveness within an enterprise and makes it hard to adopt to changes and deliver flexible IT solutions.

The basic concepts and techniques behind SOA are not a new to application development, in fact they have been used in various contexts in the past. Only now with heterogeneous application landscapes that can be found in business computing, the demand for such concepts has leveraged their use. In general a SOA focuses on services that can be reused and flexibly combined. Thus many people expect SOA to remedy their problems with their enterprise IT landscape. But there are also some problems to the introduction of an SOA that have to be taken into account. For example some projects failed due to the lack of sufficient governance structures or the wrong granularity of the developed services. Along with the potential benefits of an SOA, also the problems and risks will be addressed in the following chapters.
SOA is introduced in many companies to help them dealing with their problems and renovate their IT landscape. But with the first projects getting out of hand and first problems arising it gets more and more obvious, that certain constraints have to be taken into account. Nevertheless many solution provider try to push SOA into the market as part of their software solution portfolio (e.g. IBM WebSphere, SAP Netweaver, etc.). Resulting in high expectations, this lead to a big hype surrounding the SOA development. According to Gartner Research, SOA has already passed the climax in its Hype Cycle for Emerging Technologies. As shown in figure 1.1 SOA has already reached the "Through of Disillusionment".

This classification is not without controversy as some circumstances lead to different classifications. For example a high number of articles in current technical magazines deal with SOA, which makes it hard to believe that this topic has already lost interest. Furthermore, the number of companies that plan to investment into SOA in 2006 regarding to [CIO06] is very high. This could suggest that SOA is somewhere between the "Peak of Inflated Expectations" and the "Through of Disillusionment". But it seems to be a legitimate proposition to say that SOA has passed the peak of the hype. Therefore, a rational reflection of this technology is more then ever necessary to support a proper usage of SOA, especially a neutral assessment of its potentials and its risks. This thesis gives an overview over the existing literature about SOA in general and about the business and IT related impacts of SOA in particular. Although a lot of literature casually covers these issues there are only a few focusing on this topics. Among the huge variety of origin, level of detail, intended audience and motives of the authors it is very hard to find relevant articles. In addition, there seems to be no common understanding of basic terms and concepts, e.g. some literature

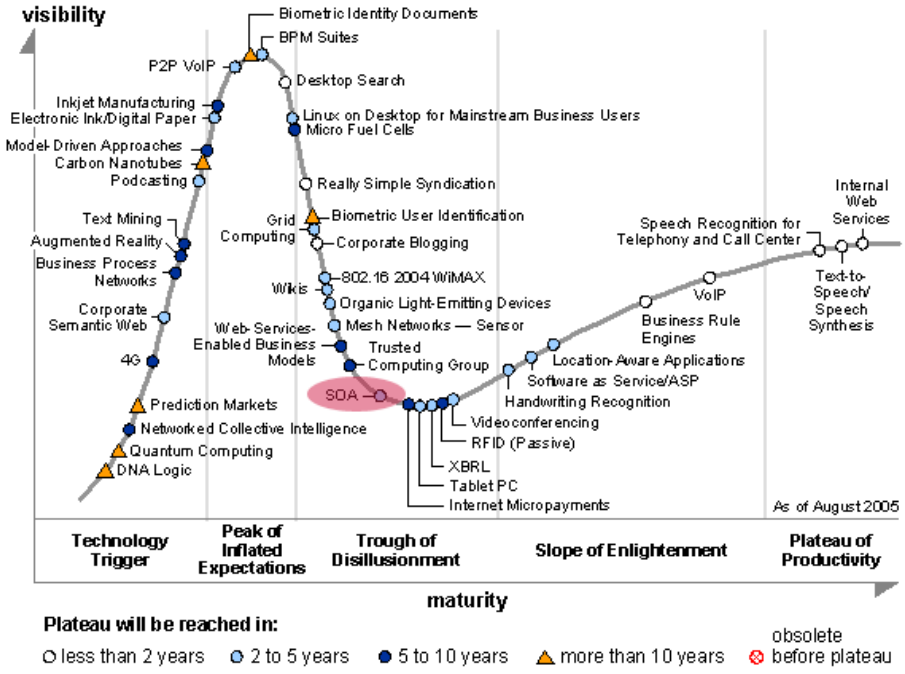


Figure 1.1: Gartner Hype Cycle for Emerging Technologies

refers to service oriented architecture, others to Web Service architecture, sometimes meaning the same.

1.2 Structure of this Thesis

This thesis provides an introduction into the basic ideas of an SOA and its underlying concepts. Chapter 2 will describe the foundation of SOA that is generally accepted and found common throughout the articles, based on two representative resources. Thereby topics that are argumentative will be pointed out, as well as topics that are field of ongoing research. In addition, the key issues that are relevant relating to an SOA will be elaborated. Finally, SOA will be compared to other important technologies and concepts that can be found in an Enterprise Architecture (EA) today, to clarify their similarities and their distinctions with SOA. This will help to outline the specifics of SOA and shows its application areas.

These key issues are building a foundation for the analysis that will be described in chapter 3. One goal of this thesis was to find relevant articles covering the topic SOA in order to provide basic definitions around this subject. In addition to the basics of an SOA, also the benefits and risks of an SOA from a business and an IT perspective as well as the alignment of business and IT within an SOA should be described. Therefore, the most interesting and credible articles on these topics should be found to attain a very comprehending and holistic overview over the topics. At first, the approach of the analysis will be described in detail. Some general results about the material used for the analysis will be presented, followed by a detailed categorization of the articles by the topics established earlier. Afterwards implications from this findings will be described.
The results of the categorization of the articles and the general findings were used to depict the most relevant sources, dealing with SOA, from an academic perspective. In chapter 4 the content of these analyzed papers will be described and summarized. Thereby, SOA will be analyzed from a business and an IT perspective in order to cover the most interesting topics. For both perspectives, IT and business, potentials and risks as well as benefits and problems will be drafted and compared to the basic SOA as it is described in chapter 2.
At last chapter 5 summarizes the concept of an SOA as it is widely used today and described in the analyzed articles. Furthermore prospects will be given for future development on SOA, based on the open issues and the propositions within the surveyed articles.

Chapter 2

Basic SOA

This chapter gives an overview over the basic understanding of an SOA and describes the components that are usually mentioned to be part of an SOA. As there is no generally accepted definition of an SOA and its relevant parts, instead the concepts that appear throughout the literature in the context of SOA are described. These concepts are important for understanding SOA and many of them are important parts for building an SOA. This thesis does not try to find a definition for SOA, instead important topics are described, to give the reader an overview over the basic concepts. Many of the ideas that are described in this chapter are adopted from the books *Service-Oriented Architecture* by Thomas Erl [Er05a] and *Enterprise SOA* by Dirk Krafzig et al. [KBS05], which give a good overview over SOA. Further sources were used to extend these ideas and to show different approaches.

At first, all the underlying concepts of an SOA are described in section 2.1. Thereby only the very basic ideas are covered and abstracted from concrete implementation issues. The ideas that are common throughout these two books will be opposed to the distinctions.

At second, core characteristics and the expected benefits of an SOA are illustrated in section 2.2 and 2.3. These benefits are often the motivation to introduce and maintain an SOA, and are important to understand many of the analyzed articles. These expected benefits are also crucial for the later discussion of the business value and the application of an SOA within an enterprise in chapter 4.

In addition to the concepts that are part of an SOA also a comparison is made with other concepts and methods in Software Engineering. Some of

them address similar problems or are forerunners of SOA. This helps to show the differences and commonalities between an SOA and other concepts. This offers the possibility to give a better impression of the key issues of an SOA. Finally some important issues of an SOA are presented. Those are topics that extend the basic understanding of an SOA and refer to issues that are found interesting throughout the sources. These are either open challenges, special fields and applications of SOAs, as well as technical and architectural issues.

2.1 Underlying concepts

As there is no generally accepted definition of the term SOA, the understanding of the term SOA as it is described in the books by Erl and Krafzig et al. is presented in the following. Since Erl focuses on the technological aspects of an SOA and Krafzig et al. focus on the adoption of an SOA within an enterprise, these two understandings represent the two main fields of interest that are covered in chapter 4. Their understanding of SOA should illustrate the basic concepts and also demonstrate how different the definitions and the view on this topic can be, although the general ideas seem to be similar.
According to Erl, SOA is a software architecture that is built using service-oriented design principles, whereas service-orientation is a concept in Software Engineering (SE) that represents a "distinct approach for separating concerns" [Er05a] (p. 32). This means, that the functionality of a system is decomposed into smaller logical units called *services*. These services are independent from each other, but have the ability to interact with each other via a certain communication mechanism. Therefore, Erl defines the components of an SOA as: *services, descriptions, and messages*. Erl refers to this principles as the *fundamental SOA*. Services are communicating with each other via messages and to allow interaction between services, they need to be clearly specified by a description. Two services communicating with each other are often referred to as service requestor and service provider, where the requestor is the service that calls another service, which is called service provider.
In many other sources a similar definition of a basic SOA can be found. Sometimes it is also called *Web service framework*, but in general it is depicted like in figure 2.1. Services that offer their service to the public, take the role of *service providers* that promote the description of their service to

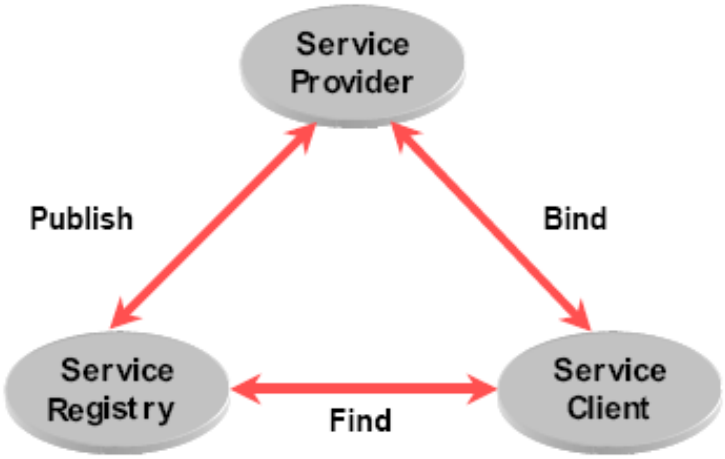


Figure 2.1: Basic Service-Oriented Architecture [Pa03]

the *service registry*. *Service requestors*, demanding the delivery of a service can find the respecting service provider using this service registry. These two services are bound to each other, to exchange data and use each others functionality. This conforms to the definition that is presented by Erl, extended by a service registry. In this service registry, services can be registered with their service description (see section 2.1.2), so that they can be found by the service requestor. Throughout his book, Erl expands his very basic definition of an SOA, using the *Web service framework*, a concrete SOA framework based on Web services. For a more detailed description on Web services, please refer to section 2.4.8. The *Web service framework*is a technology framework, that is based on standards, which are mapped onto the primitive SOA model as follows:

- Services are realized as Web services
- Messages are described by the SOAP protocol
- Descriptions are specified by WSDL
- In addition to the primitive model, a service registry using UDDI is introduced

This basically corresponds to the correlation shown in figure 2.1. As this description is used in many articles, it represents a common understanding of an SOA. However it has to be considered that this definition of SOA is only dealing with the technological aspects of an SOA. It is very closely related to solutions based on Web services and their requirements, but the concepts can be abstracted to build a general SOA foundation.
Krafzig et al. on the other hand define SOA from an enterprise point of view. Thus their definition of SOA is not as abstract as the way Erl describes SOA, instead he describes concrete parts of an SOA that solve enterprise related problems. Krafzig et al. even give an explicit definition of SOA and the respective components: "A Service-Oriented Architecture (SOA) is a software architecture that is based on the key concepts of an application frontend, service, service repository, and a service bus. A service consists of a contract, one or more interfaces, and an implementation" [KBS05] (p. 57).
These elements of an SOA and their dependencies as defined above are visualized in figure 2.2. Each entity in this diagram is represented by a rectangle. Each line that connects these entities and leads from a higher level to an entity depicted below, describes a consists-of relationship. For example, SOA consists of application frontends, services, service repositories and a service bus. .
The application frontend is the owner of the business processes, and is connected to the services via the service bus. Services are defined via an interface that allows access to the service, a contract that specifies the functionality and the implementation of the service. The implementation of a service provides relevant business logic and data. Services provide business functionality that can be used by application frontends or other services. A service repository is used to store the service contracts of the services and thereby enable the usage of the services.
Obviously there are certain commonalities among the two SOA definitions. Both contain services as a basic entity within their concepts, and services descriptions, that describe the services that are provided. Also both contain a component for communication between the services, but where Erl deals with messages as an abstract concept, Krafzig et al. use a service bus. In Erl's definition of fundamental SOA, there is is no such element as a service registry. By contrast a service registry is part of the SOA definition by Krafzig et al., but as this element is included into the definition of the Web service based SOA that Erl describes later in his book, this component can be assumed to be an important part of any SOA. The definition from Krafzig et al. contains some additional elements that cannot be found neither in the primitive SOA nor in the extended Web service based SOA.

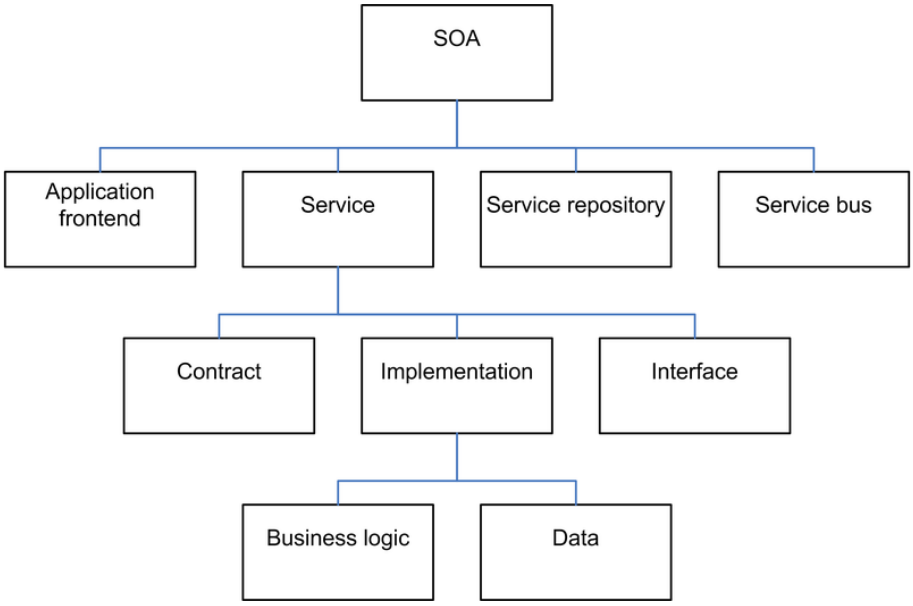


Figure 2.2: Artifacts of an SOA, [KBS05]

The following sections take a detailed look at the elements of an SOA and how they are seen by the two authors. These sections were chosen in order to group the elements according to their common functionalities.

2.1.1 Services

The service concept can be found with a very similar definition throughout the literature and is the element with the most broadly accepted understanding. According to Erl a service "encapsulates logic within a distinct context". This context can either be "specific to a business task, a business entity, or some other logical grouping" [Er05a] (p. 33). The definition by Krafzig et al. sounds very similar: "A service is a software component of distinctive functional meaning that typically encapsulates a high-level business concept" [KBS05] (p. 59).
Both definitions describe a service as a software component that encapsulates a specific part of logic. This logic usually represents some kind of business related logic. The strong relation to business functionality is what distinguishes SOA from any other concept that propagates separation of

concerns. For example in an object-oriented environment, objects also encapsulate a specific part of logic, but in contrast to service-orientation this logic is not directly related to business functionalities. Section 2.5 takes a closer look at the comparison of SOA with object-orientation.
The reason why the term service shares a very similar definition among many articles may be found in the real world analogy, the service industries. Services are offered can be used in a similar way as within an SOA. The services are mostly generic and not specific to customers which makes the service delivery more efficient.
The definitions do not say anything about the granularity, the level of detail a service covers. This can either be a single process step, a subprocess or an entire process. Depending on the purpose, the stakeholder of the system, the intended reuse and many other influences, the granularity of a service can vary. To define services with a reasonable granularity is a very big challenge when building an SOA. Section 2.4 takes a closer look at granularity and the involved difficulties.
So far the understanding of services was very similar in both sources. But Krafzig et al. define some additional components, a service comprises of. These are a contract, an implementation and the interface. Since the interface and the contract are related to the service description, they will be covered in section 2.1.2. The implementation is further divided into business logic and data. As already described, the business logic that is implemented by a service, is one of the central ideas of an SOA. On the other hand, a service that provides access to a persistent storage of data is called data centric service. This distinction is not in contrast to Erl's fundamental SOA. It seems that Erl just implicitly assumes that the storage of data is part of many services. But the data centric services exemplify one specialty of services. In classical n-tier applications access to persistent storage is provided by one tier for the entire application, in a service oriented environment, each service is responsible for its own entities.

2.1.2 Service description

Since a service exposes its functionality to be used by other services or programs, each service needs to provide a description. To interact, services must be aware of each other, which is achieved by such a *service description*. Erl provides a very basic characterization of a service description: "A service description in its most basic form establishes the name and location of the service, as well as its data exchange requirements" [Er05a] (p.35).

This kind of relationship between services is often referenced to as loose coupling. Meaning that one service remains independent of another service. It is just depending on the information that is provided by such a service, not on the concrete service implementation. The benefits of loose coupling will be covered in chapter 2.2, when the specifics of SOA and the expected benefits are discussed.
In the definition of SOA by Krafzig et al., there is nothing named service description. Instead they define a service contract and a service interface, both as part of a service. Basically the functionality that Erl sees in a service description can be found within the contract. According to Krafzig et al., "The service contract provides an informal specification of the purpose, functionality, constraints, and usage of the service" [KBS05] (p.59). A formal interface definition is considered optional. This formal definition of the interface corresponds to the description demanded by Erl, which provides information that is used to find and use the service. But Krafzig's et al. definition goes beyond the formal aspects and they explicitly claim that a contract "provides more information than a formal specification" [KBS05] (p.59). This can be a detailed description of the semantic and the functionality of the service. The interface that Krafzig et al. mention, can be considered part of the service definition, but as it is also very implementation specific, it could also be assigned to the service. The role of the interface is to provide a concrete implementation of the interface, that is accessible to the service consumer. This are often service stubs, which are a local representation for the interface that is located elsewhere. As this stubs describe the implementation of the interface, they are considered to be part of the service description, within this thesis.
Since the definition of services can be considered crucial when working with services, the service definition was also incorporated into the *Web service framework*. The Web Services Description Language (WSDL) [1] has become one of the key concepts of Web services. WSDL is an XML based description of Web services and provides a standardized description of web services, that is platform- and protocol-independent. This might also be an important factor for the success of Web services in the recent past.

[1] http://www.w3.org/TR/wsdl

2.1.3 Service communication

The motivation for defining a service description was to allow services to communicate with each other. But the communication mechanism for services still needs to be specified. In the context of Erl's primitive SOA he claims, that for the communication of services in a loosely coupled manner, a special communication framework like *messaging* is required. This implies that these messages have to be independentand autonomous, like a service. Later in his book, in the context of the *Web service framework*he describes, how these messages are implemented using the SOAP[2] messaging framework. SOAP is a standard message format, that provides a common format and transport protocol for all services within an SOA. These messages encapsulate intelligence, are document-style and highly extensible, to meet the requirements of SOAs for independent units of communication.
Krafzig et al. define a service bus for the communication within an SOA, in comparison to Erl' more generic approach, using messages. Nevertheless, messages are also used for communication within the service bus. The introduction of a service bus is more focused on the needs of enterprises with a heterogeneous application landscape, which is one of the key issues that are addressed by Krafzig et al. A service bus is used to connect the participants of an SOA as described in section 2.1. This can be either from the application frontend to invoke a service or services communicating with each other. This concept of a service bus is similar to a software bus, as it is used in the CORBA environment[3]. The idea of a software bus or in particular a service bus, is to enable objects or services to locate each other and call supplied operations. Thereby it is important, that a service bus offers various communication methods like synchronous, asynchronous or file based communication, to offer a broad variety for different vendors and applications. In addition to the communication a service bus should also offer "Technical services such as logging, auditing, security, message transformation, or transactions" [KBS05] (p.65).

2.1.4 Service repository

Specifying services with a service description, to allow them to communicate with each other, is a fundamental principle of an SOA. But with an increasing demand for flexibility and dynamic usage of services, a central

[2]http://www.w3.org/TR/soap
[3]http://www.corba.org

repository with information about the offered services is needed. As such a repository is not necessary for an SOA, Erl does not use such a concept when describing his fundamental SOA. Later, in the context of his *Web service framework*he introduces this concept. This central repository is called service registry in the context of Web services and is usually implemented using UDDI (Universal Description, Discovery and Integration)[4]. The main goal of a service registry is to offer the possibility to provide access to the service descriptions and to discover services. This includes latest versions of services that are already known by there service descriptions as well as the discovery of new services by certain criteria. To access these service descriptions programmatically or by hand, each record contains information about the *business entity* and one or more *business service* areas. These business entities contain information about the organization or service provider. Business services provide information about the service offered, which is basically an additional description of the service. Erl subdivides the registries into public and private registries, where private registries are only accessible within an organization. In Krafzig's definition of an SOA, a service repository "provides facilities to discover services and acquire all information to use the services" [KBS05] (p. 60). This comprises information that extends the service contract like physical location, information about the service provider or technical constraints. Like Earl, they distinguish between public repositories and private repositories within an enterprise. They focus on repositories within an enterprise since the main focus of their book lies on enterprise SOA and public available repositories face some additional issues. Furthermore, they also state (like Erl), that an SOA can be built without a service repository, or without a technically implemented repository. This could for example be printed service contracts or a proprietary database, but again, this solutions would not be usable in many enterprise scenarios, especially in dynamically and large projects. Krafzig et al. list some information that should be contained in enterprise-wide service repositories. This includes service operation, and arguments signatures as well as information about the intended performance of the service, which somehow overlaps with their definition of a service contract. Furthermore, information about the service owner, access rights and transactional properties of the service should be included in the service repository. Krafzig et al. also distinguish between development time binding and runtime binding of services. Services that are known and bound at development time must be known in advance and provide only little flexibility, whereas services that are bound during runtime offer more flexibility and make more use of a service repository.

[4]http://www.uddi.org/

2.1.5 Additional elements

The elements that have been described in previous sections can be considered important parts of any SOA, as they are mentioned in most available articles. Nevertheless in some articles additional elements of an SOA can be found that allow for special topics and applications. Considering all this specialties would make the basic understanding of an SOA much to complex and does not necessarily lead to a better comprehension of the topic.
As an example of such additional elements the *application frontend* as introduced by Krafzig et al. [KBS05] (p. 58) can be seen. An application frontend can be compared to the presentation layer in a multi-layer application. It interacts with the end users, initiates business processes and receives the results. Krafzig et al. call them "active players of an SOA" , because they "initiate and control all activity" [KBS05] (p. 58). Such an application frontend can be a graphical user interface, like a Web application or a rich client application. Batch programs and processes that react to events are further examples. Those application frontends delegate their responsibilities to one ore more services, which execute the business process.
This example of an additional element aims at pointing out a specific problem within an SOA, namely where the control over the business processes happens. But as described earlier, this component is not fundamental for characterizing an SOA. This might be the reason, why it appears in Krafzig's definition of an SOA, to show the integration into the enterprise, but is not part of Erl's fundamental SOA.

2.2 SOA characteristics

After the basic concepts of an SOA have been described, this section takes a closer look at the special characteristics of an SOA and the benefits that are commonly expected of an SOA. These benefits often result from one or more of the described characteristics. Some of the characteristics that are presented can also be found in other concepts of software engineering (SE), but the combination of the characteristics is very specific to SOA. In section 2.5 a detailed comparison with those concepts and their characteristics will be made.

2.2.1 Loose coupling

Basically, the most fundamental characteristic of an SOA is *loose coupling*. It can be found across the literature in many articles and represents one of the basic ideas behind SOA. Coupling "refers to the degree to which software components depend upon each other" [KBS05] (p. 47), and describes the way in which the different services are connected. Instead of tightly coupling the services together, the services are using each others functionality while still remaining independent. This can be achieved by the use of service contracts and service repositories as described above. These concepts offer the possibility to dynamically call the functionality of other services using their interface description and the repository to find appropriate services. According to Krafzig et al. [KBS05] (p. 47), several levels of coupling exist. The *physical coupling* describes how remote components are connected. If they are tightly coupled, a direct physical link between the components is required, e.g. a client requires a server to be alive. Loose coupling at this level can for instance be achieved by the use of message queues that act as an intermediary. The *communication style* also has an influence on the degree of coupling. In general asynchronous communication is more loosely coupled than synchronous communication where one component has to wait for another components response. The next level of coupling is based on the underlying *type system*. Stronger type systems result in tighter coupling of the components, as the components have to adopt to the respective type systems of other components. This makes changes to the system very hard and the system becomes less flexible. The level of *interaction patterns* describes how distributed components interact. This can be either navigating through object oriented components or message based systems. While the navigation through object trees leads to a high degree of coupling since a lot of knowledge about the provided objects is needed, message oriented systems provide a simpler interaction model. Therefore, the message based approach provides a better solution for loose coupling as the data that is needed is provided in a single message. The *control of process logic* can be managed centrally or business processes can be highly distributed. If the business processes are distributed throughout the system, they are more independent and therefore more loosely coupled. The drawback of such a system is the fact, that no globally defined consistent process state can be guaranteed. The last level of coupling considers the way in which *services are discovered and bound*. If services are bound statically, which is usually done at compile time, the system is very tightly coupled, whereas the use

Level	Tight Coupling	Loose Coupling
Physical coupling	Direct physical link required	Physical intermediary
Communication style	Synchronous	Asynchronous
Type system	Strong type system (e.g., interface semantics)	Weak type system (e.g., payload semantics)
Interaction pattern	OO-style navigation of comlex object trees	Data-centric, self-contained messages
Control of process logic	Central control of process logic	Distributed logic components
Service discovery and binding	Statically bound services	Dynamically bound services

Figure 2.3: Levels of loose and tight coupling [KBS05] (p. 47)

of service repositories to dynamically bind the services leads to a loosely coupled system. Figure 2.3 gives an overview over the levels of coupling, as described by Krafzig et al.

Loose coupling can make systems very dynamic and flexible. But loose coupling also makes systems more complex and harder to maintain. The concept of loose coupling requires a broad understanding of the system, additional skills from the developers and extra effort in development.

2.2.2 Autonomy of services

Another fundamental characteristic of an SOA is the *autonomy of services*. "Autonomy requires that the range of logic exposed by a service exist within an explicit boundary" [Er05a] (p. 303). This means in other words, that a service can govern itself without having to deal with many external dependencies to other services. Furthermore, this ensures more independency of the service implementation and avoids side effects. The logic that a service represents does not necessarily need to be exclusively owned by one service, but during the execution the service has full control over its represented logic. Erl distinguishes two types of autonomy: *service-level autonomy* and *pure autonomy*. Service-level autonomy on the one hand describes the case where a service has distinct boundaries, but shares resources with other components or services. Pure autonomy on the other hand means, that the logic is "under complete control and ownership of the service" [Er05a] (p. 305).

2.2.3 Abstraction

Abstraction is an important characteristic of software systems. It allows to conquer complex systems, as it hides implementation details and logic. Components of systems can be combined without having to know the details of the components, as they are only used as black boxes that are accessed through their interfaces. The implementation of such a black box can be changed independent from the rest of the system as long as the interface remains unchanged. Thereby, abstraction can increase the robustness and the flexibility of a system.
An SOA is characterized by abstraction because the services act as black boxes that hide their internal logic and the interfaces are described and made public using the service description. The amount of logic that is contained within a service is not further specified and leaves room for different design strategies. Actually, finding the right granularity of services is a difficult and important topic in SOA development and will be further examined in section 2.4. This influences the level of abstraction that a service and an SOA provide.

2.2.4 Reusability and composability

An SOA promotes *resusability*, since the logic is divided into services, that represent components within the system. This logic can then be used by other services or applications with only little effort. To achieve reusability, design standards have to be applied to provide a generic service, that can be used for a wide variety of applications. Services do not provide reusability on their own. Services that are not well designed or solve a very specific problem are unlikely to be reused. But in general, since services encapsulate logic that is often related to business functionality, this logic can easily be called by any other service.
Again, abstraction plays an important role in defining a flexible interface to the service to make it potentially reusable. If the interface is not specific to certain activities, the chance for reuse increases. Using intelligent messages for the communication among the services further increases reusability, as some of the logic can be incorporated into the message. Erl claims, that especially with the use of SOAP messages this goal can be achieved, as messages can be equipped with processing instructions. These instructions can contain application specific information, whereas the service remains generic.

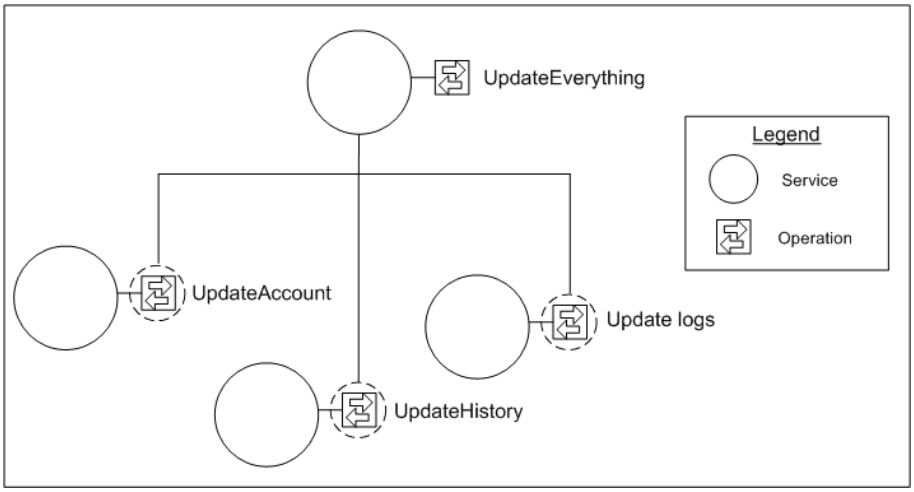


Figure 2.4: UpdateEverything operation encapsulates service composition [Er05a] (p. 301)

As already described, the granularity of logic a service represents is not limited and can vary. Given the principle of *composability*, services can be composed of other services and therefore further promote reuse. Especially when services with a fine level of granularity are built, composition makes it possible to offer a service that provides an interface that is very coarse grained. This allows building a hierarchy of services with different levels of granularity, providing the possibility of reusing low level services. Figure 2.4 shows how a service can be composed of several services. An important concept within an SOA that closely relates to composability is the *orchestration* of services, which will be covered in more detail in section 2.4.

2.2.5 Discoverability

Discoverability characterizes the ability of an SOA to provide a mechanism for the discovery of services. This is usually implemented by a service repository as described in section 2.1.4. Such a mechanism for the discovery of services enables an SOA to avoid redundant services and redundant logic. Discoverability is very important to leverage the true potentials of an SOA, as it also encourages reuse of services. If the possibility to find adequate services that offer necessary functionality is not given or not convenient to

use, the stakeholders of an SOA will rather build redundant services than use existing ones.
In addition to discoverablity on the architectural level, which describes the facilities that have to be provided by the SOA architecture, there is also discoverability on the service level. On this level, discoverability "refers to the design of an individual service so that it can be discoverable as possible" [Er05a] (p. 309). Meaning, that a service is designed in such a way, that they provide a declarative service description that can be used to find and value the service and its provided operations. Service description and service contracts that have been depicted in section 2.1.2 contain elements that are able to achieve discoverability within an SOA, if they are used properly.

2.3 Expected benefits

The adoption of an SOA is driven by certain expectations regarding the benefits an SOA provides. For an enterprise, two main goals in managing its IT-systems are important, because they generate a competitive advantage on the market. On the one hand the reduction of costs of IT and on the other hand an improvement in their business processes. Each of the benefits that an SOA can imply is aimed at one of these goals. The following section shows some benefits that an SOA can possibly create, but the degree to which these benefits occur strongly depends on the concrete realization of the SOA. Many benefits are listed in the articles, but to give an overview, the following section just covers a view important aspects.

2.3.1 Agility and Flexibility

Increasing agility and flexibility of the business processes within the enterprise is an important goal for many companies. They face the need to quickly react to changes in market and business to adopt their business processes to the ever-changing environment. The driving forces behind this constant change can be customers that change their consumer behavior, competitors offering new products or mergers and acquisitions, that require the integration of the IT systems of two companies. One of the biggest obstacles for agility is complexity. Thus the reduction of complexity results in more agile IT systems, which leads to more flexibility within an enterprise.
SOAs are designed to support ongoing change in IT systems and constituting flexible and agile systems. Loose coupling, composability and reuse,

are characteristics that help decreasing complexity and increasing flexibility. As the services are loosely coupled within an SOA, they can be dynamically exchanged, reducing dependency among each other. If the logic within one service changes, only the affected service has to be changed without any further influence on other services. This reduces the complexity in development, testing and maintenance of the system. Composability reduces the complexity of systems. Large and complex system can be decomposed into smaller units of logic that can be managed more easily. With decomposition, also an abstraction from technology can be achieved, as for the usage of composed services no knowledge of the underlying technology is required. Composing services to build new business processes also contributes to an increased flexibility. This results in less development efforts and faster development, as existing services can be reused.

2.3.2 Technology independence

Many IT projects are focused on technology issues instead of the required business functionality. SOA abstracts from the underlying technology using service contracts and service composition. The technology is hidden from the consumer as it is encapsulated within services. Especially in very heterogeneous IT landscapes this can lead to a better integration of different applications into a business process. With different platforms involved and a tight coupling of business functionality to technology it can be very difficult to integrate these distinct systems. SOA provides a solution to this problem, as these heterogeneous systems can communicate with each other using a common protocol and standardized interfaces, which are decoupled from the technology. In comparison to other approaches (see section 2.5 for details) SOA tries not to eliminate heterogeneity but enables different technologies to integrate. SOA also decouples technology lifecycles from business lifecycles [KBS05] (p. 246). This makes an enterprise less vulnerable to technology changes since the business functionality is decoupled from the technology. This makes an exchange of technology less painful and limits the impact on business functionality to a minimum.
Nevertheless, an SOA also requires a baseline of technological infrastructure to do its work. this includes a service repository and as specified by Krafzig et al. also a service bus. But this technology is encapsulated wheres the overall technology dependence is reduced.

2.3.3 Cost savings

Many of the benefits covered, also result in cost savings, that can either be "direct (saving IT costs) and indirect contributions (saving business costs)" [KBS05] (p. 243). Krafzig et al. also mention some examples where costs can be saved at business and IT level due to SOA. At a business level, choosing the cheapest supplier, supported by a flexible, SOA based IT, can save production costs. SOA can enable such an advancement, as it provides an integration platform that can be used by the suppliers and also enables flexible adaption of business processes. Another example for saving at the business level is an improved financial reporting. Because SOA allows, that "different parts of the architecture share live-data" [KBS05] (p. 243), important data can be integrated more frequently, which results in more up-to-date business reports. These again can lead to faster and more well-founded decisions, which save costs or increase returns. These examples for saving costs at the business level might seem farfetched, but especially in low-margin markets they can make an important difference and can lead to an edge over business competition.
Cost savings at IT level can be achieved in various ways. IT cost savings are usually the first that come to mind when thinking about the benefits of an SOA, as they immediately relate to the development costs. Furthermore, a more efficient implementation and deployment when using an SOA, as well as a simpler design of business processes can reduce project costs. However these reduced cost have to be weighed up against higher initial costs for introducing an SOA. Another savings potential lies in the reduction of maintenance costs. Since side effects within the system are reduced and the application landscape is decomposed into manageable subsystems, the efforts for maintenance can be decreased. An up-to-date documentation that is necessary due to the service contracts also ensures fewer problems in maintenance. Employing an SOA also provides a future proof solution. Abstraction avoids a tight coupling with underlying technology, which makes it more technology independent and even allows the exchange of technology. The ability of an SOA to integrate existing systems protects previous investments and does not render existing applications useless. Finally, once the functional infrastructure is established, it can be reused with many future scenarios. These are no immediate cost savings in IT development, but in cases where new approaches are necessary they can save costs compared to other technologies.

2.3.4 Reuse

Reusing existing assets is an achievable goal in IT development. SOA offers a possibility for reusing existing business logic and program code at various levels, in conjunction with the ability to compose services. Depending on the required functionality, services with very basic functionality as well as services that describe entire business processes or subprocesses can be reused. Every service can be used as a building block for the creation of new new services, that implement new business processes. With the usage of service repositories for the discovery of services, the use of a service is not limited to projects or business units, but can be applied throughout the enterprise. This often has been a problem with existing architectures, because the information about existing code, that could be used was not available across the enterprise. Reusing services also "significantly reduces redundancies and inconsistencies in business data" [KBS05] (p. 244). Services would not only share the same code, but as there is only one actual instance of an service that is supervised by a responsible department, the data is more likely to be consistent and the problem of data integration will be reduced. Reusing services also increases the robustness and reliability of a system. Services are described by a service contract, which reduces failure when integrating services, due to a mandatory description of the interfaces. At the same time, the usage of existing services reduces testing and debugging times, since services that are already used in production are usually very robust and testing can basically be limited to integration tests.

2.3.5 Reduced risk

According to Krafzig et al., the reduction of risk "could even be the most important benefit of an SOA" [KBS05] (p. 249). They describe certain domains where SOA can reduce critical project risks. An SOA requires a detailed description of its services within the service contract or service description (see section 2.1.2). This helps ensuring that a projects requirements and objectives are correctly defined, especially when specifying business-level services. Thereby the gap between business and technology is closed and the business side becomes more involved in the development process. Thereby, the risk to develop a product that fails the requirements is reduced.
Loose coupling and composability also enable an SOA to follow a divide-and-conquer strategy. This reduces project risks, as the decomposed services are easier to manage and therefore problems can be identified early and resource

planning can be improved. Furthermore, the dependencies within a project are reduced due to the logical decomposition, which also eases project management and reduces integration problems of sub-projects. Shifting to a new technology involves a lot of risks for an enterprise, as not only the investment into the new technology is involved, but also legacy systems and their integration into the new technology have to be taken into account. An SOA can reduce such introduction risks. Companies often fear to invest into the wrong technology which becomes rendered useless after a short period. SOA is based on architectural concepts rather than on concrete technology, which helps reducing this risk. Trough its flexibility and loose coupling, SOA provides the possibility to integrate legacy systems. Therefore, existing and mature technology can be used, which further reduces the risk of migration.

2.3.6 Business orientation

In many companies, the contribution of IT to the business success is realized through monolithic applications with static and fixed functionality. The demand for more flexible and agile IT systems is linked with a demand for a better business orientation of IT. Often the gap between the demands from the business departments and the potentials provided is dissatisfying.
Instead of developing monolithic applications with limited use, SOA allows the development of services that represent encapsulated business logic. Together with reuse and composition, this not only offers the possibility for flexible design of business processes, but also allows a stepwise development of applications. This leads to a business infrastructure that lowers the mismatch between technology constraints and business demands. Applications can be developed more flexible and their functionality can be changed and enhanced more easily. With the use of an SOA and business oriented services, the business can also be involved in the development process by concrete specifying their demanded services. This results in a better business understanding by the IT and bridges the gap between the two different views on the business.

2.4 Topics of an SOA

Achieving the above covered benefits of an SOA is not a trivial task. Many stakeholders are involved in the adoption of an SOA and many pitfalls have to be avoided. This section takes a closer look at special topics that have to

be considered when deploying an SOA, to leverage its full potentials. At the same time, these topics are mentioned in many articles in the analysis that is described in chapter 3. The topics in this section are described with regard to potential problems and basic methodologies for facilitating a successful SOA.

2.4.1 Granularity of services

When describing the benefits of an SOA, the potential for describing services with various levels of granularity was mentioned. But this advantage can also become a problem when choosing the wrong granularity. Finding the right granularity for involved services is a challenge. Fine grained services are limited to a small part of business logic and only exchange small amounts of data. Whereas, coarse grained services are likely to exchange large amounts of data, but encapsulate a larger amount of business functionality within one service request. When splitting functionality into many fine grained services, instead of one coarse grained service, the number of service calls increases, which might cause a communication overhead. The basic idea of services within an SOA is, that "every service is an entity of distinctive functional meaning that typically encapsulates a high-level business entity" [KBS05] (p. 60). This would in general lead to coarse grained services that closely align to business functions. The drawback in designing coarse grained service interfaces, lies in the possible loss of reuse potential. Service requestors that only need parts of the functionality might not be interested in using the offered service functions. In summary it can be ascertained that "interoperability is promoted in coarse-grained services, and reusability is more fostered in finer-grained services" [Er05a] (p. 557).
Finding the appropriate level of granularity within an SOA depends on the concrete circumstances of the project and the enterprise. Basically it is desirable to generate coarse grained services, that align with business functionality and provide a higher level of abstraction. This reduces dependencies and the number of communications.

2.4.2 Orchestration

Orchestration describes an extended concept of an SOA that is based on the principle of composability. Complex activities are controlled by orchestration services, which encapsulate a set of workflow logic. The orchestration

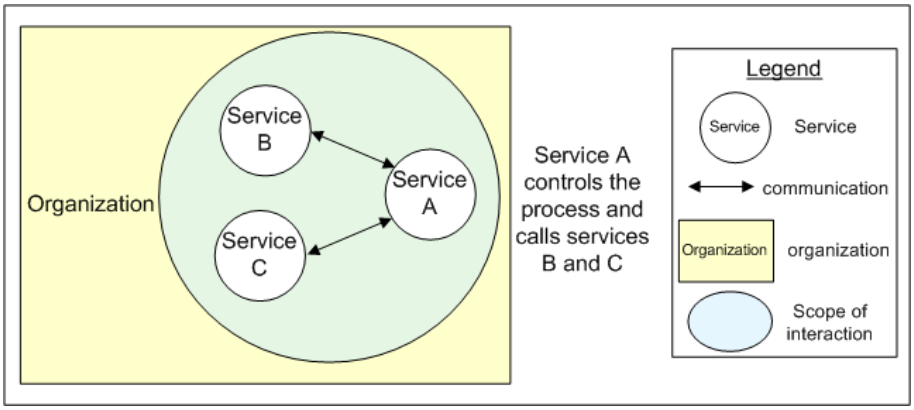


Figure 2.5: Orchestration in dependence on [Er05a] (p. 201)

service controls the workflow and therefore interacts with other services, as illustrated in figure 2.5. This enables different applications to collaborate and to connect different processes. Moreover, this leads to an abstraction of workflow logic and standardization of processes. The encapsulation of the logic within orchestration services also reduces the complexity of the system and leads to a better maintainability of the workflow logic. Furthermore, "through the use of orchestrations, service-oriented solution environments become inherently extensible and adaptive" [Er05a] (p. 205). As orchestration services provide potential integration endpoints for other processes and applications, orchestration can be considered as an integration enabler and "for many environments, orchestrations become the heart of SOA" [Er05a] (p. 206). To standardize orchestration, the specification of Web services Business Process Execution Language (WS-BPEL)[5] was introduced by the OASIS[6] consortium. It is aimed at the use of Web services, but the basic concepts also apply to SOA in general.

2.4.3 Choreography

Collaboration between companies based on SOA is a complex task and requires rules for collaboration. Service choreography addresses this issue to allow services from different organizations to work together. The basic idea

[5] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
[6] http://www.oasis-open.org

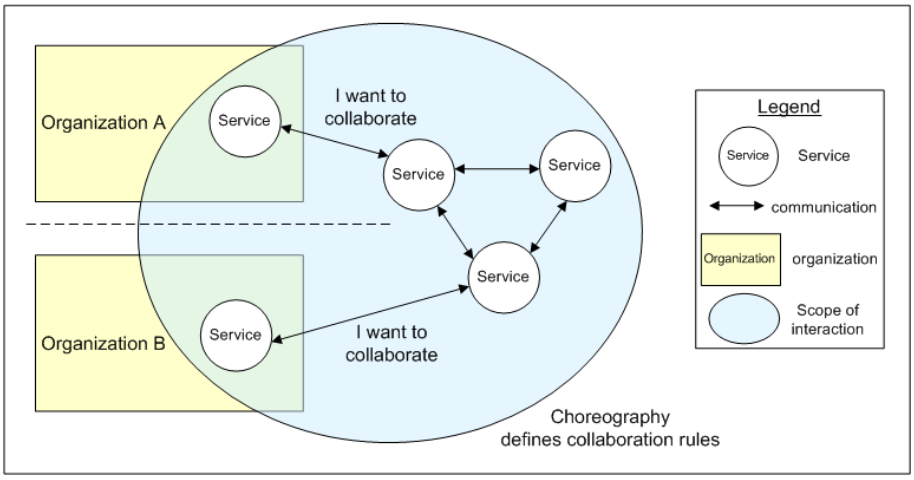


Figure 2.6: Choreography in dependence on [Er05a] (p. 208)

is to arrange organized message exchanges of services, whose collaboration logic is not necessarily controlled by a single organization. Instead, it "acts as a community interchange pattern" [Er05a] (p. 211). Services from different organizations expose their functionality and allow each other to collaborate using choreography to execute complex tasks. Figure 2.6 shows, how services from different organizations can interact, enabled by choreography.

While choreography enhances composability, reusability, and extensibility, it is strongly depending on discoverability to allow other participants to find and use their provided services. Altogether, choreography helps making enterprise IT more agile. To provide a common basis for choreography, some specifications emerged, trying to lay the groundwork for SOA based business collaboration. A very famous specification is provided by the W3C and is called Web Services Choreography Description Language (WS-CDL)[7]. Like WS-BPEL it also uses Web services as underlying technology, but nevertheless, the basic concepts also apply to SOA in general.
It is important to understand how orchestration and choreography relate and what the differences are. Orchestration on the one hand describes business workflows within one organization that controls and owns the business workflow that describes the orchestration. Choreography on the other hand describes collaboration between different organizations that are not owned

[7] http://www.w3.org/TR/ws-cdl-10/

by a single one of these organizations. Orchestration can be seen as "a business-specific application of a choreography" [Er05a] (p. 212). Orchestration and choreography overlap in the sense, that orchestration can also incorporate participants from other organizations. Still the logic is centralized within one organization.

2.4.4 Service layer

Traditionally in an enterprise, the enterprise logic can be divided into business logic and application logic. The business logic represents the business requirements that are described as business processes. They show a series of steps with their constraints and dependencies that are involved in achieving a business goal. Application logic on the other hand is described by an implementation within an IT system. Application logic implements business process workflows within a distinct boundary of the technological solution, which is limited by IT infrastructure, security constraints and technological prospects. Services provide abstraction and composability and are therefore hard to fit into this concept. In fact, when looking at SOAs, services are assigned to a new layer - called service interface layer - between business logic and application logic, as illustrated by figure 2.7. This allows to provide a high degree of abstraction, as the service interface layer can encapsulate logic from both, the business layer and the application layer. The service interface layer acts as an integration and connectivity layer between the flexible business processes and the confined applications. Although, the service interface layer is illustrated as a single layer, this does not mean, that all the service implementation resides within this layer. In fact, the implementation is likely to be encapsulated within the application layer, but the service interface layer describes the service interfaces and thus provides a layer of further abstraction.
Krafzig et al. and Erl, both also describe a concept for layering services within the above described service interface layer. Erl divides them into orchestration-, business-, and application service layer [Er05a] (p. 337). Krafzig et al. on the other hand define enterprise-, process, intermediary-, and basic layer. Basically the idea behind their ideas seems to be similar. They both describe how the granularity of services can be used to compose a hierarchy of layers, providing more coarse grained interfaces at the upper layers.

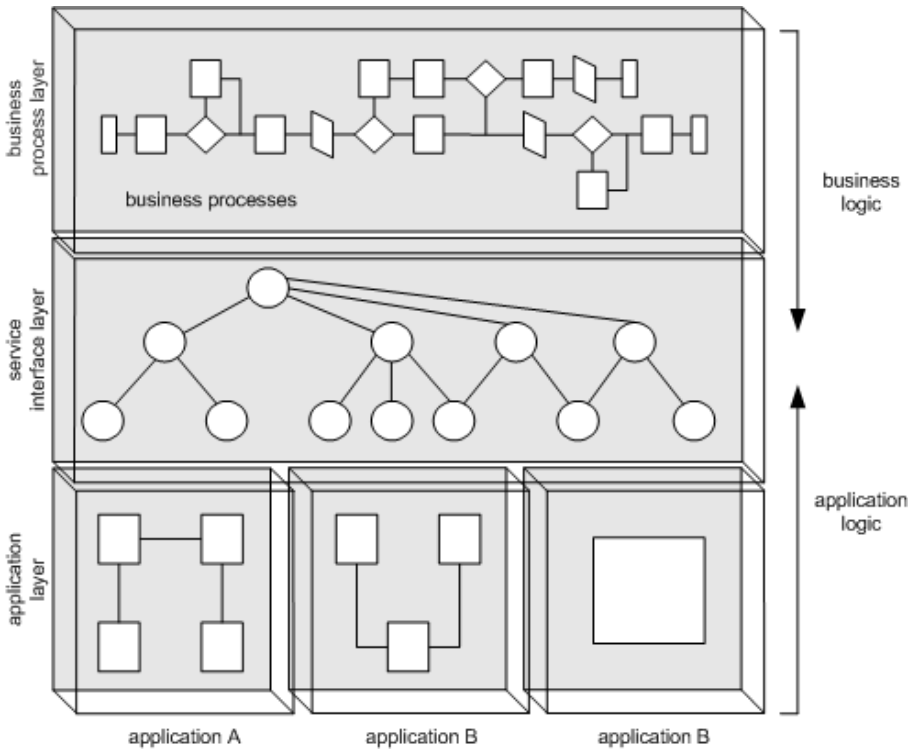


Figure 2.7: The service interface layer, [Er05a] (p.282)

2.4.5 Service bus

Section 2.1.3 already covered the topic of communication within an SOA, including a short overview over the service bus as it is defined by Krafzig et al. The problem of the basic description of SOA, that just defines messages for communication, becomes obvious when looking at enterprises with a heterogeneous IT landscape. It seems to be almost impossible to introduce an enterprise wide standard for a single communication technology. Therefore, in extension to the definition of messages for communication, more sophisticated concepts are needed, to face many of the problems within an enterprise. According to Krafzig et al. a service bus should "enable basic interaction with different service components across the different platforms" [KBS05] (p. 159). Thus they claim, that a "Meta bus" [KBS05] (p. 162) should be created. This Meta bus should support different communication types and products from different vendors. Therefore, the SOA communica-

tion infrastructure has to conform to standards to support interoperability. The communication within a service bus is handled using service stubs and dispatcher. The service stub is a software component that resides on the communication client. It provides easy to use, access methods to the remote services. Furthermore it also encapsulates functionality for handling security and network issues. On the side of the service provider, acting as a counterpart, the service dispatcher is responsible for handling incoming service requests over the network. It analyzes the incoming requests and invokes the service operations that are called by the service requestor.
One concept that emerged from the service bus approach, is the Enterprise Service Bus (ESB) [Ch04a]. An ESB is based on a coarse-grained XML communication protocol in combination with a message-oriented middleware.

2.4.6 Governance

Many different stakeholders are involved in the implementation and management of an SOA. Each of them is driven by the requirements of his field of responsibility. In addition to an initial introduction of an SOA, changing environments and new needs, have to be properly managed and supervised. Especially with the use of an SOA that promotes heterogeneity and distributed responsibilities, it is a challenging task to maintain the usability and to make use of the expected benefits.
In general IT-governance describes, how decision rights and accountabilities have to be specified to achieve the desired outcome of IT usage [WR04]. This approach can also be applied to SOA. For an SOA to work properly, also decision rights and accountabilities have to be assigned to achieve the desired outcome of an SOA. Policies have to be established, that ensure that business logic is implemented using services and that services are actually reused. Furthermore, it has to be assured, that services are designed in a reusable manner, not tailored to the current and specific requirements. Guaranteeing the compliance to these policies is a very challenging task, as for many stakeholders, following the SOA requirements means extra effort, without returning an immediate benefit. Sometimes the established processes and organizational structures within an enterprise might also hinder the proper usage of an SOA. As an example, instead of reusing existing services, someone might implement new ones from scratch, because circumstances can make this the more convenient approach. Lacking a proper governance for an SOA, can put all efforts for introduction at risk, as many of the described benefits require the support of the employees.

2.4.7 Business Process Management

Business Process Management (BPM) is a management topic, focusing on the strategic and operational aspects of process orientation. Unlike Business Process Reengineering (BPR), which focuses on optimizing business processes by reinventing existing business processes, BPM focuses on the incremental change and evolutionary optimization of business processes. Whereas BPR is a one-time activity, BPM is a continuous improvement process.
The reason why BPM is mentioned in the context of an SOA, is that for the support of BPM, an IT system is required which provides the "technical platform for realizing BPM management" [KBS05] (p. 105). Such a system is called a Business Process Management system (BPMS). Since such a system is highly complex and demands the underlying logic to align with the overall business processes, SOA seems to be a promising approach to support such a system. As covered earlier in this thesis, orchestration and composition of services can be used to build services that reflect entire business processes or parts of a business process. Compared to other concepts like object-orientation, this leads to a system that is more likely to fulfill the demands for flexibility, loose coupling and business alignment, which are required to enable BPM. According to Krafzig et al. "SOA represents a good foundation for adopting a process-oriented approach" [KBS05] (p. 115). Such a SOA that adopts the process-oriented approach is called "process-enabled SOA" [KBS05] (p. 115) and is the target of Krafzig's roadmap to SOA implementation.

2.4.8 Web services

When looking at articles about SOA, Web services are mentioned in almost any of them. Although, it is possible to build an SOA without using Web services, e.g. by using CORBA, obviously Web services play an important role for the deployment of an SOA. For example, M. Huhns and M. Singh claim that "Web services [...] form the basis for an SOA" [HS05](p. 78). In fact Web services are "just one possible implementation technology" [St06] (p. 54) for implementing services in an SOA. On the other hand, it is claimed that Web service are the key technology that founded the enormous interest in SOA, since they "form the basis for an SOA" [HS05] (p. 78).
Web services are based on XML standards for data exchange. The World Wide Web Consortium (W3C) defines Web services as follows: "A Web ser-

vice is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by internet protocols" [W3C06c]. Several standards exist in the area of Web services, ranging from basic Web service specifications to elaborate definitions for Web service transactions[8] and Web service choreography[9]. The basic Web services can be described using figure 2.1, as an interaction between service requestor, service provider and service repository. The basic actions that are performed by service requestor, service provider and the repository are defined by the Web service description of the W3C[10]. For the communication between the different participants, the *SOAP* protocol[11], which is based on XML is used. Furthermore, for the description of service interfaces, the *Web Service Description Language (WSDL)*[12] is used. It contains relevant information about the service provided, including interface description as well as necessary information on how to access the service. Finally, the service repository is implemented using *Universal Description, Discovery and Integration (UDDI)*[13]. This is a directory index, containing service descriptions along with the organization providing the service.
Web services support the basic characteristics of an SOA, described earlier, but according to [HS05] (p. 75), "they are unnecessary limited". Anyhow, as some papers dealing with Web services, their findings can still be abstracted for SOA in general, if the special characteristics of Web service are taken into account.

2.5 Comparison with other concepts

Looking at the characteristics and potential benefits of an SOA, the question emerges, what differentiates an SOA from existing concepts and what the similarities are. This section compares SOA with concepts that relate or contrast to the service-oriented concept. This also helps getting a better understanding of SOA, as it allows to classify SOA in reference to existing

[8] http://www-128.ibm.com/developerworks/library/specification/ws-tx
[9] http://www.w3.org/TR/ws-chor-model
[10] http://www.w3.org/2002/ws
[11] http://www.w3.org/TR/soap
[12] http://www.w3.org/TR/wsdl
[13] http://www.uddi.org

knowledge and experience. Some of the covered concepts aim at the solution of similar problems, while others have different goals, but share similar strategies.

2.5.1 Client-Server architecture

Client-server architecture distinguishes between a client and a server, each playing a specific role. Any instance of the client software can send requests to a central server. The main idea behind this concept is to split responsibilities between these two entities, where the server is merely responsible for data management and storage and the client for the presentation and business logic. This is also referred to as the two-tier architecture. Modern architectures often rely on the n-tier principle to further divide the system into several different modules.
Within a client-server environment, a majority of application logic is placed into the client software, resulting in monolithic fat clients. In an n-tier environment this application logic can reside in a separate application server, resulting in a more complex system. In contrast, within an SOA the distribution of this application logic can vary, since any service can be a service requestor, that invokes this logic. In a client-server environment, usually a large amount of processing is done by the client. Furthermore, "communication is predictably synchronous" and "connections or often persistent" [Er05a] (p. 91). SOA on the other hand is highly distributed, with each service working in explicit functional boundaries. The communication within an SOA can be either synchronous or asynchronous, providing an increased level of flexibility. In addition, SOA puts a large amount of intelligence into the messages allowing stateless and autonomous services.
At first glance, the basic SOA model looks like traditional client-server interaction with requests and responses, but one big difference is the fact, that a service can be both, client and server at the same time, as described in section 2. In addition, the client-server model is tightly-coupled, whereas the SOA model supports loose-coupling between the services.

2.5.2 Component based development

Component Based Development (CBD) focuses on the decomposition of systems into smaller units of logic which can be viewed as black box and are exposed by an interface. Obviously some of these characteristics have also

been described for the basic requirements of an service. In fact many of these concepts form the basis for service-orientation. R. Perrey et al. describe the relationship between service-orientation and CBD as follows: "One damaging assumption for service is that it is another term for component. Many of the good practices that were expounded in component-based development and integration methodologies are relevant but services are freed from the ties to platform or language specific technical frameworks" [PL03]. One big difference between these two concepts concerns the approach to flexibility and reusability. Although CBD also addresses these issues, an SOA allows dynamic invocation of services, based on policies, whereas CBD views components merely as distributed objects. Furthermore, services focus on continuous maintenance and improvement and can be offered to an growing number of consumers. Composability is another important aspect, that is available in service-orientation and is not supported by CBD. In summary it can be ascertained, that SOA focuses on the system behavior, and not on the system structure, like CBD.

2.5.3 Object orientation

Object-orientation (OO) is a development strategy, which is based on the concept that systems should be built from a collection of reusable components. Data and functions are integrated into components called objects [BD00]. According to T. Erl, "object-oriented programming is commonly used to build the application logic encapsulated within Web services" [Er05a] (p. 107). Where service-orientation provides loose coupling between the services, OO is more tightly coupled due to static class dependencies. Furthermore, the services within an SOA should provide coarse-grained interfaces to foster reuse, OO relies on fine-grained interfaces. In general, the scope of services can vary significantly, whereas objects "tend to be smaller and more specific in scope" [Er05a] (p. 107). As claimed earlier, services are designed to function in a stateless manner. In object orientation, data and logic are bound together in objects, resulting in stateful units.
Both concepts support composition to build higher-level entities. Within an SOA, this leads to an increased level of abstraction, while preserving the lose coupling between the services.
Unfortunately, in object-orientation, the composition is realized using inheritance, which increases the coupling between objects. Furthermore, reuse is claimed by both concepts, but in a service-oriented environment reuse plays an important and fundamental role for the creation of systems. In contrast,

object-orientation claims to provide reuse, but it is on the one hand not necessary and on the other hand enabled at a low-level, which makes location and usage of those components a difficult task.
As shown, these two concepts have fundamental differences, yet they complement each other. As shown later in this thesis, object oriented paradigms can be used to implement services, especially with the application of additional concepts like design patterns.

2.5.4 Enterprise Application Integration

Enterprise Application Integration (EAI) "entails integrating applications and enterprise data sources so that they can easily share business processes and data" [Qu05] (p. 341). Since many organizations have a very heterogeneous IT landscape, EAI tries to enable inter-operation between various applications allowing them to effectively and conveniently use each other's functionality. EAI is mostly driven by the IT, aimed at low-level connectivity between applications. Many of the EAI solutions are based on proprietary solutions of software vendors and require a lot of hand work for the integration of the different systems. These systems therefore lack an "ease-of-integration for users" [Ti04] (p. 34) and instead requires an extensive involvement of developers. Nevertheless, both concepts are well suited for the integration of legacy systems, based on the concept of encapsulation.
One big advantage of an SOA compared to EAI, is that it allows integration across organizations, using orchestration and choreography. Where EAI tries to integrate components to avoid heterogeneity, SAO enables heterogeneity. SOA allows services to be developed using different platforms. Instead, the interface description and the communication are standardized to enable easy integration of the individual components. Nevertheless, SOA can be used as a driver for EAI and vice versa [KBS05] (p. 211). A proper deployment presumed, an SOA can "address many of technical and core concerns related to EAI" [KBS05] (p. 212).

Chapter 3

Analysis

Based on the fact that SOA is a very hyped technology on the market, a lot of literature exists, which makes it very difficult to find a common basis and a good starting point for finding relevant information. The existing literature is characterized by a lot of "best practices" and commercial solutions, making it even harder to value its true benefits. As an example one can find a lot of references to the redbooks provided by IBM. Academic articles dealing with the business value of SOA are rare and even the technology related work is often very technical and seldom covering architectural basics. The goal of this thesis is to provide the reader with an overview over important articles and articles that examine the basic criteria for Service-Oriented Architectures from an IT perspective on the one hand and those that deal with the business perspective on the other hand. Furthermore the most important and common points as well as conflicting opinions throughout the articles will be excerpted.
In many fields of research that are already established usually there exist some basic articles that describe the basic understanding of the subject and will be referenced in many of the succeeding articles. SOA, as a concept, is quite a young field in Software Engineering there seems to be a demand for such essential work to refer to in future work. Out of this demand the concept of the research, described in the following, was derived.

Figure 3.1: Stepwise procedure of analysis

3.1 Approach of analysis

The following procedure was chosen, in order to find relevant articles with academic background and classify them according to their relevance for further analysis. Thereby, the main goals of the search were to find the most relevant articles defining and describing the basic concepts of an SOA. The goals of the analysis were to find out more about the historical evolution of SOA and to structure the huge amount of articles according to their covered topics. Especially the classification of the articles, based on their content will be very important for the later description of SOA. The procedure as depicted in figure 3.1 will be described in detail in this chapter.

3.1.1 Stepwise procedure

At first numerous sources were searched for papers covering SOA, service-orientation and services. These sources include: *The ACM Digital Library* [1], *IEEE Xplore* [2], the *Electronic Library for magazines of the Technische Universität München* [3] and to some extent *Google search* [4]. The reason for this broad search was on the one hand to find academic documents about the subject and on the other hand to include papers that are available to the public. This should provide a broad spectrum of articles, to assure that no relevant articles were left out and eventually give some conclusion whether there is some linkage to academic articles. During this search a lot of articles and websites were screened. To limit the result set to a manageable amount, articles that were found irrelevant for this thesis, based on the title or a quick overview over the content were excluded from further inspection. Some additional articles were added to the papers contemplated in this thesis, because

[1] http://portal.acm.org
[2] http://ieeexplore.ieee.org
[3] http://rzblx1.uni-regensburg.de/ezeit/searchres.phtml?bibid=TUM
[4] http://www.google.com

they were found interesting and valuable, but were not accessible through the above mentioned resources. This includes conference proceedings from conferences and articles supplied by experts, as well as articles that were found following links on websites, or from proceedings with references to other relevant articles. Most of the articles that were found are written in English, but a few interesting articles in German ware also included [5]. This procedure unfortunately lacks traceability on the articles that ware selected, but therefore helps getting a preferably broad overview over the subject. To manage the articles the tool *EndNote* [6] was used. EndNote helps managing bibliographies and is able to export the data to various formats, which was very important for the later statistical analysis.
During the next step, the bibliographies of the articles found in the first step (primary sources) were checked for referenced articles (secondary sources). For each primary source all references were checked if they were dealing with SOA, otherwise they were excluded from further inspection. Many of the articles that are referenced by the primary sources are dealing with special topics and domains were SOA is applied as a solution, and therefore have no relevance for the later research. Some other references are covering technologies that can be used to implement certain building blocks of an SOA. These references are used to show the link of articles to this technologies, but will not be part of the keyword-analysis, which has been performed during one of the next steps.
The set of articles that was found was then further analyzed in detail. Each reference was indexed with several attributes, including the publication year, the names of the authors and many more. In addition to the attributes relating to the articles itself, some information about the dependencies of the articles among each other was also added to the analysis. As each article from the set of primary sources was searched for all the references to other papers, those references were stored in a separate document. The idea behind this was to create a graph representing this information in order to visualize the dependencies and give an overview over the articles and their correlations.
Finally the sources that were available in digital form, either as a *PDF* [7] file or a *HTML* [8] page were searched for keywords, that should help describing the content and the main focus of the articles. Articles that could not

[5] A detailed listing of the papers with regard to their language distribution is given in section 3.2.

[6] http://www.endnote.com

[7] Portable Document Format (PDF)

[8] Hypertext Markup Language; http://www.w3.org/TR/html4/

be found or were only available in print-out, had to be excluded from this search. The keywords used for the search (see appendix A for a detailed list), were selected in such a way, that they describe the relevant issues of an SOA from a business and IT perspective as shown in chapter 2 but also produce a limited number of results for the search on the documents. This was a very challenging task, as many of the terms used for describing an SOA are very basic terms that might occur within any text. For example the term *service* will not help to categorize the articles as well as the term *architecture* as it will be contained in almost any document because it is contained within an SOA. Another point for the selection of the keywords was, to get both, keywords that describe the IT related issues as well as business related topics. As the results from this keyword analysis were used to depict the most relevant articles for describing SOA from a business and IT perspective in chapter 4 there should be sufficient keywords for analyzing both perspectives.

The analysis was done using *Adobe Acrobat Reader* [9] and a text editor with the capability to search multiple files. Acrobat Reader was used to search the articles that were available as PDF files, the text editor was used for searching across the files, that were available as HTML. Each keyword was searched within all the available articles. The number of found results for each document was then entered into a spreadsheet containing the labels of the documents, their titles and a column for each keyword. As a result, for each document the number of occurrences of the keyword can easily be found and be used to get an overview over the topics covered in the articles. The complete list can be found in appendix A.

The main drawback about this evaluation is the fact, that the numbers of search results is absolute. Meaning, that articles that comprise a higher number of pages are more likely to contain more hits. But a relative result, specifying the number of results per page might also lead to a wrong selection of papers. For instance an article comprising many pages could contain a few pages focusing on the searched topic, which would make the number of results per page very low, but the absolute number is higher than in other pages.

The spreadsheet was further enriched with information about the results from the dependency analysis. For every article, the number of outgoing references and the number of incoming references was added. These are also valuable information when selecting the important and interesting articles for a special topic. Some of the articles that were found, while checking the

[9] http://www.adobe.com

references from the primary references, are not available, and are therefore highlighted in the spreadsheet. This allows to distinguish them from references that were searched, but did not contain any results from the keyword search. All this data was then used to pick out the most relevant articles to describe SOA from a business and IT perspective.

3.1.2 Graph generation

For the generation of the aforementioned graph the tool *Graphviz*[10] was used. Graphviz is an open source graph visualization software that is able to render graphs from a text file. It allows the user to choose from various output formats and layouts. Since the proprietary format of the Graphviz definition files is not very useful for further analysis of the dependencies among the articles, an *XML*[11] dialect called *DOTML*[12] was used to represent this data. For each primary source a node was generated as an XML element. When the articles were searched for referenced articles, for each reference a node was generated and an edge from the primary reference to this secondary reference was added. In case, the primary reference cites another primary reference only an edge from to this reference was added. The resulting XML document describes all the dependencies among the references. This XML document was then transformed into the DOT [DOT06] format that can be read by Graphviz and used as an input for creating the graph, visualizing the dependencies between the analyzed articles. For transforming the DOTML file into the DOT format, the Extensible Stylesheet Language Transformation (XSLT) processor Apache Xalan[13] was used.

3.1.3 Statistics

For the statistical analysis of the references, a variety of tools and approaches were used. The information that could be analyzed was available in the following formats. The dependencies of the articles was available - as described in section 3.1.1- as an XML file, containing elements for the nodes and edges in the dependency graph . The details of the articles were also available as an XML file, which was exported from EndNote, and were also incorporated

[10] http://www.graphviz.org

[11] Extensible Markup Language (XML) is a simple and flexible text format for documents that can be read by humans as well as machines; http://www.w3.org/XML

[12] http://www.martin-loetzsch.de/DOTML

[13] http://xalan.apache.org

into the spreadsheet that was generated during the keyword analysis. The XML file contains various attributes, including the year of publishing, the authors and the language of the source.
The information contained in these files had to be combined and condensed to result in meaningful statistics. As the relevant information for the statistics were both available via an XML file, XQuery [14], a query language that is based on XML was used to access the data. XQuery allows to query XML files and select elements and attributes based on certain constraints. The resulting output can again be stored as an XML file which makes it easy to import the results into a spreadsheet for further processing. To actually execute XQuery statements, *Qizx/open* [15] an XQuery engine was used. Qizx/open is open source and is very closely sticking to the XQuery specification defined by the World Wide Web Consortium (W3C) [16].
The queries that were written produced the following statistics:

- The number of articles that were written by each author, ordered by the number of articles
- The number of articles that were published in each year, order chronological
- The number of references that were pointing to and from each article, ordered by the number of references
- The number of articles grouped by the type of article, ordered by the number of articles
- The number of articles grouped by their academic background

Furthermore, XQuery was used for the spreadsheet to gather information from the different sources.

3.2 General results

During the collection and the analysis a lot of information was gathered. This section summarizes these information and gives an overview over the

[14] http://www.w3.org/TR/xquery
[15] http://www.axyana.com/qizxopen
[16] http://www.w3.org

articles involved in the research and shows some general results of the analysis. Whenever possible, the data was put into graphs to allow a better visualization of the attained results. As a large number of articles was included in the analysis, some of the tables and graphs just contain a cutout of the most important results. The complete list of the results can be found in the appendix.
After the second step in the analysis, described in section 3.1.1, when all the secondary sources were included, the number of articles adds up to a **total of 415 references**. Thereof **203 are primary references** that were found in the first iteration of the analysis and **212 are secondary resources** that were referenced by the primary references.

3.2.1 Disposability of articles

During the second step of the research, the secondary resources that were included into the further research, were chosen by their topic and their context within the articles. Although a lot of effort was put into gathering the referenced articles, some of them could not be found or at least are not available in electronic form, so they had to be excluded from the keyword search. A few of the primary resources only were available as print-out and could not be included in this step either. Nevertheless, the references to other articles from the printed articles or books, were still included for finding secondary resources.

Table 3.2 shows all the references used for the research, ordered by their disposability. Each resource was assigned to one of the following categories:

- **Book:** Some of the articles include references to books. As these books could not be used for the keyword analysis or they were not available at the accesible libraries, they were not included into further inspection.
- **HTML:** Articles that only were available in HTML, were downloaded and searched separately in the keyword analysis. The results of this search have been integrated with the results from the PDF search.
- **WebPage:** Some of the articles referenced Web pages that either did not contain a single article that could be downloaded and stored as a HTML file, to include them into the analysis, or that were to generally specified. Some of the references for example point to technical specifications that span several pages, making an analysis also inapplicable.

Disposability	References	#
Book	[Al04], [Ap03], [Ba03b], [Ba05], [BCK03], [Bi05b], [Bu96], [Ce02], [Ch04a], [CKK02], [Er04], [Go03], [HB02], [HS02], [JMP04], [Ka03], [Le03], [Li03], [Ne02], [Ne04], [PB03], [Sc92], [SG96], [SH05], [WC03], [We04], [We05], [ZTP03]	28
Not available / Not found	[BK04], [Bl02], [Bu05], [Cn04], [Di05a], [Gr05], [Hu03], [KZ04], [Le04], [Le05c], [Li04], [Ma04b], [Ma05d], [QDD04], [Sp02], [Tu04], [Za03], [ZY04b]	18
WebPage	[An03], [Ar05], [Bo04a], [Bo04b], [Ch01], [Ch02a], [Ch03b], [Ch04b], [Ha03a], [IBM01], [IBM02a], [IBM02b], [IBM02d], [IBM03], [IBM06], [MG03], [OA04], [SUN02], [W3C02a],[W3C02b], [W3c02c], [W3C04b]	22
PDF	[AAS04], [AB05], [Ac05], [AH06], [AHZ04], [AKL99], [Al97], [APS06], [Ar03], [Ba02], [Ba03a], [Ba06], [Be01],[Be02], [Be03], [BGA01], [BH05], [Bi04], [Bi05a], [BL06], [BLM05], [BM05], [BNZ06], [Bo01], [Br02], [BRV04], [BSD03], [BSM05], [Ca03], [Ca04], [Ca05a], [Ca05b], [CBD05], [CDH05], [CFR03a], [CFR03b], [Ch02b], [Ch03a], [Ch05a], [Ch05b], [CHT04], [Ci06], [CK01], [CK05], [CLM03], [CM04], [Co04b], [Co05], [Cr05], [CT05], [Cu02], [Cu03], [Cu05a], [Cu05b], [CV05], [CYN05], [Di04], [Di05b], [Du05], [EDS06], [EL04], [EMM06], [En04], [Er05b], [EWA06], [FB02], [Fe06], [FF03], [FHH04], [FLB05], [Fo02], [Fo03], [Fo04], [Fo05], [Fo06], [FP02], [Fr04], [FS05a], [FS05b], [FT05a], [FT05b], [Fu05], [FWK02], [Ga02], [Ga06], [GB04], [Ge06], [GMK98], [Go02], [Go04], [Gr03], [Gr04], [Gu05], [Ha05a], [Ha05b], [Ha06], [HB03], [HB04], [HB05], [HCD06], [HCZ05], [He02], [He05a], [He05b], [HG05], [HLC05], [HS05], [Hu05], [HW03], [IBM02c], [IBM04a], [IBM04b], [IBM04d], [IBM05a], [IBM05b], [In05a], [In05b], [In06], [JDT05], [Je05], [Ji05], [Jo04], [Jo05], [JRM05], [JS05], [JW05], [Ka05], [Ka06], [KC05], [Ke04], [Ke05], [Ke06], [KEA05], [KJE05], [KKP05], [KKO03], [KKS06], [KL03], [KLS05a], [KM04], [KMM06], [Ko05], [KPR04], [Kr04a], [Kr04b], [KR05a], [Kr06a], [Kr06b], [Kr95], [Ku05], [Ku06], [KZ05], [LA02], [La02], [La03], [La06a], [La06b], La06c], [Le00], [Le05a], [Le05b], [Li06], [LMC05], [LMS06], [Lo06a], [LRS02], [LT03], [LU05a], [Lu05b], [Ma03], [Ma04a], [Ma05a], [Ma05b], [Ma05c], [Ma0b], [Ma06c], [Ma06d], [MBE03], [Me04], [MKC04], [MLD05], [MM04], [Mo02], [Mo06], [MPP02], [MPT99], [MR03], [MSR05], [MVA05], [Na02], [Na03], [Na05], [Ne05], [NG05], [NM02], [No06], [NP03], [NS03], [OHE03], [OY05b], [OYP05], [Pa02], [Pa03], [Pa05a], [Pa05b], [Pa06], [Pe03a], [Pe03b], [Pe04], [PG03], [Pi04], [PL03], [Pl05a], [Pl05b], [Pl05c], [Po06], [PS02], [PSZ01], [PW04], [PY04], [QDS05], [Ra03], [RB05], [RD05a], [RD05b], [RD05c], [RD05d], [RD05e], [Ri05], [Ro05], [SA05], [Sc02], [Sc03], [Sc05], [Sc06], [SDS04], [SEH02], [SEH05a], [SEH05b], [Sh02], [Sh03], [Sh04], [Si04], [Si05], [Sk05], [Sk06], [SLC05], [Sm04], [SM05], [SM06], [SN03], [Sn06b], [So06a], [So06b], [Sp05], [SS04], [St05], [St06], [St97], [SUN04], [SUN06a], [SUN06b], [SUN06c], [SUN06d], [SVA04], [SVW04], [SW04], [Ta03], [TAC05], [TBB03], [TCP05], [TG03], [Th05], [Ti04], [Ti06a], [Ti06b], [TJ05], [TJH05], [TP02], [Tr04], [Ts04], [Ts05], [TYL05], [Vi02], [Vi06], [VM06], [Vo03], [Wa04], [Wa05], [WDL04], [WEI06], [Wi06], [WK05a], [WK05b], [WT04], [WYH05], [Ya03], [YIM05a], [YIM05b], [YK97], [Yu04], [Za87], [Za96], [ZCC04], [Ze03a], [Ze03b], [Ze04], [Zi04], [Zi05], [Zy02], [ZY04a]	314
HTML	[Ar04], [BR03], [Bu02], [CC06], [Ch05c], [CKB04], [Co04a], [Da04a], [Da04b], [Fe03], [Gi06],[Ha03b], [He03], [He06], [IBM00], [IBM04c], [IBM04e], [KLS05b], [KP03], [Li05], [Ma06a], [Mi05a], [Mi05b], [Pa05c], [Ro04], [ZKG04]	26
Printout	[Sn06a], [KBS05], [Gr06], [Er05a], [Br03]	5
Not searchable	[SB96], [OY05a]	2
		415

Figure 3.2: Disposability of articles

Therefore, the URL was included into the list of references, but not into the keyword analysis.

- **PDF:** More than three-quarter of all articles are available as PDF files and are included in the attached CD. When searching the secondary references, this was the preferred document format, as it allows searching across all documents, using the Adobe Acrobat Reader [17].

- **Not available / Not found:** Not all of the articles referenced in the primary sources could be found, despite an elaborate search. The reason for this is manifold. Some articles referenced, are only available for a fee, because they are written by a consulting company. Others are not available to the public. This includes thesis and doctoral dissertations that are only available at the respecting universities. And a few articles could simply not be found with the provided information of reference.

- **Not searchable:** Although, the search engine of Adobe Acrobat works well, some articles could not be searched, because they are provided as a scanned image with bad quality.

Figure 3.3 gives a visual overview over the disposability of the articles.

3.2.2 Distribution in time

For each article involved in the research, the year of publication was annotated. In addition to following the standards for referencing publications, which requires a publication year to be included in the label, this also provides valuable information about the articles involved. Figure 3.4 shows how the articles included into the research, disperse over the years.

Obviously most of the articles included into the research were written after 2002. Only five percent of all papers were written before 2002 and many of these articles cover basic topics of software architecture rather than SOA specific issues. This underlines the assumption, that the topic SOA is comparatively new. Still, any of the underlying concepts and ideas are already established and are known for years, which is shown by the fact that the secondary references go back to 1987. It might be the case, that some early articles covering SOA were simply not found by the chosen approach,

[17] http://www.adobe.com

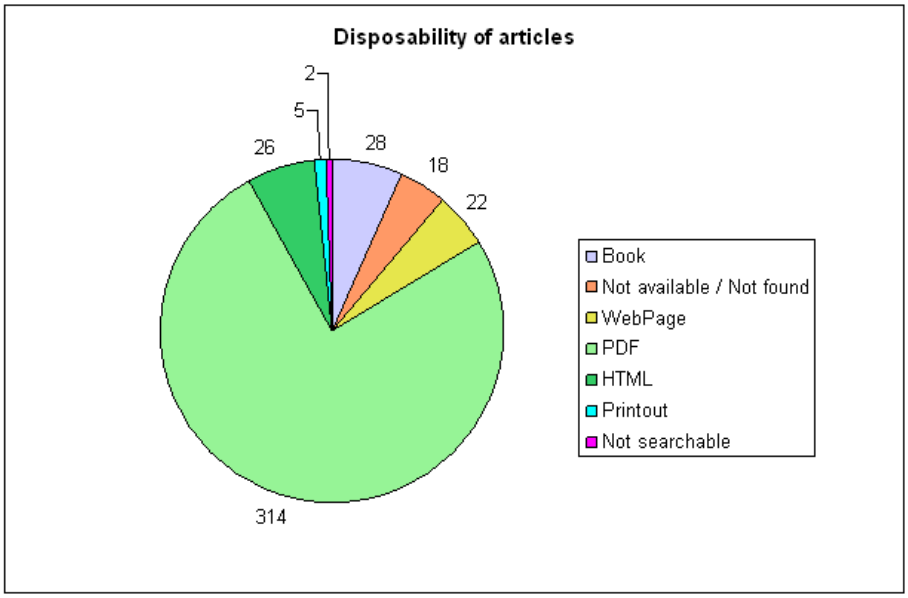


Figure 3.3: Visualization of disposability

but if there was any fundamental paper from this time, covering SOA, it should have been referenced in some of the articles.
The most articles contained in the research were published in the year 2005. In 2004 and 2006 only half the amount was released. One would expect the number of articles published to increase further in 2006, compared to 2005, as the topic is currently said to be of much interest. The reason for this result in the statistics is twofold. Firstly, the search for articles was performed until the middle of the year 2006. Therefore, articles from 2005 were still very up-to-date. Secondly, when searching articles that were written in 2006, they are very unlikely to contain references to articles that were also written in 2006, because of the short timespan that would be necessary between writing and publishing. In fact only two of the secondary references refer to articles from 2006. When looking at the constantly increasing numbers of articles since 2002, the number should actually be expected higher than 2005, at the end of 2006. This could be proven by another analysis in the future.

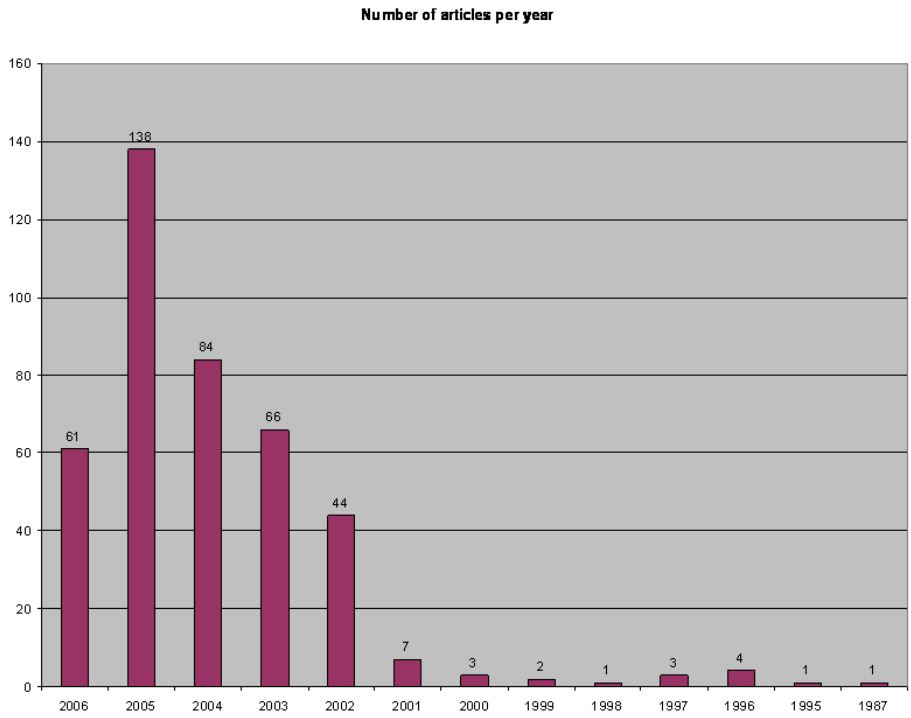


Figure 3.4: Distribution in time

3.2.3 Authors

When working with such a large number of references it is also interesting to look at the authors of the articles, to find out who can be considered an domain expert in this field. Therefore, a query on the list of references was made, giving the authors and their articles along with the number of articles written. The number of articles written refers to any participation of an author in an article. Many articles have been developed by many authors in cooperation. Figure 3.5 shows the results of this query. To keep the results concise, only the authors, that wrote at least three articles were included in the results.

When looking at the results, one can see several entries containing names of organizations like IBM, SUN and W3C. This entries denote articles or Web pages that could not be assigned to single authors. To provide a reasonable abbreviation and complete entries in the list of references, the name

Author	#	References
IBM	15	[IBM00], [IBM01], [IBM02d], [IBM02c], [IBM02a], [IBM02b], [IBM03], [IBM04e], [IBM04d], [IBM04c], [IBM04a], [IBM04b], [IBM05a], [IBM05b], [IBM06]
F. Curbera	8	[An03], [Ar05], [Ch01], [Cu02], [Cu05b], [Cu03], [MKC04], [We05]
F. Leymann	8	[He05b], [KL03], [Le03], [Le04], [Le05c], [Le05a], [LRS02], [We05]
S. Weerawarana	6	[Ch04b], [Ch01], [Cu02], [Cu03], [FWK02], [We05]
J. Yang	6	[Ji05], [OY05a], [OY05b], [OYP05], [PY04], [Ya03]
SUN	6	[SUN02], [SUN04], [SUN06a], [SUN06b], [SUN06c], [SUN06d]
P. Malinverno	5	[BM05], [BSM05], [Ma06b], [Ma06c], [Th05]
B. Benatallah	5	[BSD03], [HB03], [Ze03a], [Ze03b], [Ze04]
M. Dumas	5	[BSD03], [Di04], [QDD04], [Ze03a], [Ze04]
E. Newcomer	5	[Bo04b], [Bo04a], [Ch02a], [Ne02], [Ne04]
I. H. Krüger	5	[BK04], [Kr04b], [Kr04a], [KM04], [KMM06]
S. Dustdar	5	[JDT05], [RD05c], [RD05e], [RD05d], [RD05b]
Y. V. Natis	5	[Na03], [Na05], [NS03], [SN03], [Th05]
C. Ferris	4	[Bo04b], [Bo04a], [Ch02a], [FF03]
R. Khalaf	4	[Cu02], [Cu03], [FWK02], [KL03]
Y. Han	4	[FHH04], [TJH05], [WYH05], [Yu04]
P. Krogdahl	4	[Ke06], [KLS05a], [KLS05b], [ZKG04]
D. Smith	4	[LMS05], [LMS06], [Sm04], [Ti04]
D. Edmond	4	[OHE03], [SEH02], [SEH05b], [SEH05a]
M. P. Papazoglou	4	[OYP05], [Pa03], [PG03], [PY04]
O. Zimmermann	4	[Zi05], [ZKG04], [Zi04], [ZTP03]
F. Casati	3	[Al04], [Be03], [Ca03]
A. Arsanjani	3	[Ar03], [En04], [LA02]
N. Gold	3	[Be01], [Be02], [GB04]
P. Layzell	3	[Be01], [Be02], [EL04]
D. Budgen	3	[Be01], [Be02], [TBB03]
P. Brereton	3	[Be01], [Be02], [TBB03]
S. Bose	3	[Bi05b], [Bi05a], [Du05]
H. Haas	3	[Bo04b], [Bo04a], [Ha03a]
M. Champion	3	[Bo04b], [Bo04a], [Ch02a]
D. Orchard	3	[Bo04b], [Bo04a], [Ch02a]
H. Yang	3	[Ch05a], [ZY04a], [ZY04b]
H. Kreger	3	[CK05], [Go02], [KP03]
I. Foster	3	[Fo05], [Fo02], [FT05b]
G. Wang	3	[Fu05], [Wa05], [Wa04]
Y. Huang	3	[Hu03], [Hu05], [HLC05]
Y. Li	3	[Hu05], [HLC05], [WDL04]
M. Keen	3	[Ke04], [Ke06], [Ke05]
R. Mathew	3	[KM04], [KMM06], [Ma04b]
M. Marchese	3	[Ma03], [YIM05a], [YIM05b]
G. Piccinelli	3	[MPT99], [PSZ01], [PS02]
R. W. Schulte	3	[NS03], [SN03], [SVA04]
B. Orriens	3	[OY05], [Or05], [OYP05]
J. O'Sullivan	3	[SEH02], [SEH05b], [SEH05a]
A. H. M. ter Hofstede	3	[SEH02], [SEH05b], [SEH05a]
F. Rosenberg	3	[RD05e], [RD05d], [RD05b]
W. T. Tsai	3	[Ts05], [TCP05], [Ts04]
W3C	3	[W3C02a], [W3C02b], [W3C02c]
L. Zeng	3	[Ze03a], [Ze03b], [Ze04]

Figure 3.5: Authors of articles

of the publishing organization was entered instead of an author.
The resulting numbers of papers an author published, are influenced by the references found during the research process and the primary references that were chosen in the first place. Some authors tend to reference to their own prior work. This makes authors from the primary references more likely to achieve a higher number of articles in this research.
Recapitulating, 145 authors are listed with more than one article regarding SOA. This can be a result of the references to their own prior work, but it also shows, that a lot of experts in this area exist and that many different ideas were gathered throughout the analysis. Furthermore, this also shows a high level of interest and huge demand for SOA related articles. *Francisco Curbera* and *Frank Leymann* are the authors, who participated at the most articles. Both are listed in eight articles as an author. Some of Cubera's work also includes participation at specifications regarding Web services, which is a reason for the high number of references.

3.2.4 Reference origin

To be able to assess the various statements of the different articles, it is necessary, to know about the origin and the background of each paper. Articles that were written by software vendors might tend to overestimate the benefits of SOA, especially when the company offers a product, based on these concepts. This does not mean, that these papers do not provide valuable information, but the statements have to be considered in the context of the authors and publishers. Therefore, figure 3.6 gives an overview over the resources, based on their origin and type of reference. The following categories where chosen, to represent clusters of different origin:

- **Conference proceeding:** Given the proposition to base the research on an academic and sophisticated foundation, this category contains all articles that have been submitted to a conference. Under the assumption, that papers that are presented at a conference, have been carefully screened and reviewed, this adds the respecting articles a high level of credibility. Moreover the fact, that 160 out of 416 articles fall in that category, builds the basis for a reasonable result of the research in this thesis.
- **Journal article:** Many of the included articles were published by scientific journals. These journals include computational journals, academic journals and journals released by organizations or companies. A

Type	References	#
Conference Proceedings	[AB05], [AHZ04], [Ac05], [BGA01], [BH05], [BLM05], [BNZ06], [BRV04], [Ba02], [Ba03a], [Be01], [Be02], [Be03], [Bo01], [CBD05], [CDH05], [CFR03b], [CK01], [CM04], [CT05], [CV05], [CYN05], [Ca04], [Ca05a], [Ch02b], [Ch03a], [Ch05a], [Co05], [Cu05a], [Cu05b], [Du05], [EWA06], [FHH04], [FLB05], [FS05a], [Fo03], [Fo04], [Fu05], [GB04], [Ga06], [Gr05], [Gr06a], [Gu05], [HB03], [HB05], [HCD06], [HCZ05], [HG05], [HLC05], [Ha05b], [He05b], [Hu03], [Hu05], [JDT05], [JRM05], [JW05], [Ji05], [KC05], [KEA05], [KJE05], [KKP05], [KKO03], [KKS06], [KL03], [KLS05a], [KM04], [KMM06], [KPR04], [KZ04], [KZ05], [Ko05], [Kr04a], [Kr04b], [LMC05], [LMS05], [LMS06], [Le04], [Le05a], [Le05c], [Lu05a], [Lu05b], [MKC04], [MLD05], [MPT99], [MSR05], [MVA05], [Ma03], [Ma05a], [Ma05c], [Mo02], [MPP02], [NP03], [Na02], [Ne05], [OHE03], [OY05a], [OY05b], [OYP05], [PL03], [PS02], [PSZ01], [PW04], [PY04], [Pa02], [Pa03], [Pa05b], [Pe04], [Pi04], [Po06], [QDD04], [QDS05], [RB05], [RD05a], [RD05b], [RD05c], [RD05d], [RD05e], [Ri05], [SDS04], [SEH05a], [SLC05], [SS04], [SVW04], [SW04], [Sc02], [Sc06], [Sh02], [Sh03], [Sh04], [Si05], [Sn06a], [Sn06b], [Sp05], [St05], [St97], [TAC05], [TCP05], [TG03], [TJ05], [TJH05], [TYL05], [Tr04], [Ts04], [Ts05], [VM06], [WDL04], [WEI06], [WK05a], [WT04], [WYH05], [Wa04], [Wa05], [YIM05a], [Yu04], [ZCC04], [ZY04a], [ZY04b], [Zi04], [Zi05], [Zy02]	159
Journal Article	[AKL99], [APS06], [Ar03], [BL06], [BSD03], [Ba06], [Bi04], [Bi05a], [CFR03a], [CK05], [CLM03], [CN04], [Ca03], [Ca06], [Ch05b], [Co04a], [Cr05], [Cu02], [Cu03], [Da04a], [Da04b], [Di04], [EL04], [EMM06], [Er05b], [FB02], [FF03], [FS05b], [FT05b], [FWK02], [Fo05], [Fo06], [Go02], [Go04], [Gr03], [HB04], [HS05], [HW03], [Ha05a], [Ha06], [He02], [He05a], [IBM04b], [JS05], [Jo04], [Jo05], [KP03], [KR05a], [Ka05], [Ka06], [Kr06a], [Kr06b], [Kr95], [Ku06], [LA02], [LRS02], [LT03], [La03], [La06b], [La06c], [Le00], [Li04], [Li06], [MBE03], [MM04], [MR03], [Ma05b], [Ma06d], [Me04], [Mo06], [NG05], [NM02], [No06], [PG03], [Pa05a], [Pe03a], [Ra03], [Ro05], [SEH02], [SM06], [Sc05], [Si04], [Sm04], [So06b], [St06], [TBB03], [TP02], [Ta03], [Ti04], [Vi02], [Vi06], [Vo03], [Wi06], [YIM05b], [YK97], [Ya03], [Za87], [Ze03b], [Ze04]	99
Electronic Article	[AAS04], [AH06], [BK04], [BM05], [BR03], [BSM05], [Br02], [Bu05], [CHT04], [Ca05b], [Ci06], [Co04b], [Di05a], [En04], [FP02], [FT05a], [Fe03], [Fe06], [Fr04], [Ga02], [Ge06], [Gr04], [IBM01], [IBM04a], [IBM04d], [IBM04e], [IBM05a], [IBM05b], [In05a], [In06], [It06], [Je05], [Ku05], [Lo06a], [Ma04a], [Ma06b], [Ma06c], [NS03], [Na03], [Na05], [Pa06], [Pe03b], [Pl05a], [Pl05b], [Pl05c], [SA05], [SM05], [SN03], [SUN02], [SUN04], [SUN06a], [SUN06b], [SUN06c], [SUN06d], [SVA04], [Sk05], [Sk06], [So06a], [Sp02], [Th05], [Ti06a], [W3C04a], [WK05b], [Za96], [Ze03a]	65
Electronic Source	[An03], [Ar04], [Ar05], [Bo04a], [Bo04b], [Bu02], [CKB04], [Ch01], [Ch02a], [Ch03b], [Ch04b], [Ch05c], [EDS06], [Fo02], [Gi06], [Ha03a], [Ha03b], [He03], [He06], [IBM00], [IBM02a], [IBM02b], [IBM02c], [IBM02d], [IBM03], [IBM04c], [IBM06], [In05b], [KLS05b], [La06a], [Li05], [MG03], [Ma05d], [Ma06a], [Mi05a], [Mi05b], [OA04], [Pa05c], [Ro04], [SEH05b], [Ti06b], [W3C02a], [W3C02b], [W3C02c], [W3C04b], [ZKG04]	46
Book	[Al04], [Ap03], [BCK03], [Ba03b], [Ba05], [Bi05b], [Bu96], [CKK02], [Ce02], [Ch04a], [Er04], [Er05a], [Go03], [HB02], [HS02], [JMP04], [KBS05], [Ka03], [Ke04], [Ke05], [Ke06], [Le03], [Li03], [Ne02], [Ne04], [PB03], [SG96], [SH05], [Sc92], [WC03], [We04], [We05], [ZTP03]	33
Report	[Bl02], [CC06], [La02], [Le05b], [Sc03], [Tu04], [Za03]	7
Thesis	[Al97], [Br03], [Di05b], [GMK98], [Ma04b], [SB96]	6

Figure 3.6: Types of references

big part of the journal articles comes from various IEEE journals and the IBM Systems Journal[18]. Thse articles usually have passed several reviews and are therefore considered to be of a high quality.

- **Electronic article:** Electronic articles include mostly papers written by software vendors or commercial organizations and research papers by Gartner research. Also articles that could not be found and are evidently not references to a Web page, a conference proceeding, or a book were included in this category. The main intention of these articles is to provide an overview over certain topics and show trends that have been identified in research studies.

- **Electronic source:** This category mostly contains Web pages containing specifications or portals with information about SOA. Another big part of this category are articles that are published in the Web like articles from IBM developerWorks[19].

- **Book:** Some of the primary and secondary references are referring to books. Other than in section 3.2.1 this category contains any reference to books even the ones that were available as print-out. Still most of the books were not available for the research, but their references are important for showing sources of information about SOA.

- **Thesis:** A few articles were found being theses from universities, either Ph. D. theses, master theses or publications from a chair. They offer good academic background and a high level of independence. Unfortunately some of them were not available for the research, as they were only available at the respective university.

- **Report:** Reports are usually written by consulting firms, analyzing special topics of interest and giving recommendations based on the results. Most of them are only available for a fee, so they were not included in the research of this thesis.

3.2.5 Frequency of referencing

One of the goals of this thesis is to find most relevant articles covering the topics of an SOA. One interesting indicator for the importance of an article

[18] http://www.research.ibm.com/journal/sj/
[19] http://www-128.ibm.com/developerworks

Label	# Incoming	# Outgoing	Incoming references
An03	15	0	[BLM05], [Ca03], [FS05b], [Ka05], [KM04], [KPR04], [KZ05], [Le05a], [MKC04], [Po06], [RD05d], [SLC05], [Sp05], [SVW04], [Zi05]
Ch01	14	0	[BLM05], [Cu05b], [Du05], [Jo05], [KM04], [LMC05], [MKC04], [OYP05], [Pa05a], [Ri05], [VM06], [WK05a], [Zi05], [BNZ06]
IBM03	11	0	[Cu05b], [Du05], [EWA06], [Fu05], [Ga06], [MVA05], [QDS05], [SDS04], [SM06], [BNZ06], [KBS05]
Al04	10	0	[Ca03], [CBD05], [CT05], [Gr06a], [Le05a], [MSR06], [MVA05], [QDS05], [RD05b], [YIM05a]
Ch04a	8	0	[Ac05], [Cu05b], [KJE05], [Pa05b], [Po06], [SM06], [St06], [KBS05]
En04	6	0	[Bi05a], [FS05a], [IBM04b], [Ka05], [KKP05], [St06]
Pa03	6	0	[KC05], [Ko05], [RD05d], [TYL05], [WK05a], [YIM05a]
IBM02d	5	0	[Hu05], [JDT05], [KLS05a], [OY05a], [OYP05]
Ke04	5	0	[Ar04], [Bi05a], [IBM04b], [WEI06], [Zi05]
PG03	5	0	[Fu05], [KEA05], [OYP05], [SW04], [TJ05]
ZKG04	5	0	[Ar04], [Ch05a], [KLS05], [KLS06], [WK05a]
Ar04	4	4	[KLS05], [KLS06], [Mi05b], [WYH05]
Cu03	4	0	[BH05], [Ch03a], [KEA05], [TJ05]
QDD04	4	0	[MLD05], [QDS05], [WYH05], [YIM05a]
W3C04a	4	0	[Ba03a], [Ch05a], [Co05], [Gr06a]
KBS05	3	4	[Sn06a], [Sn06b], [St05]
Sc05	3	3	[Ch05b], [CK05], [NG05]
Bo04a	3	0	[JDT05], [Pa05a], [VM06]
Co04a	3	0	[KLS05], [KLS06], [TYL05]
Di04	3	0	[OYP05], [QDS05], [YIM05a]
Er04	3	0	[FLB05], [Sh04], [KBS05]
FB02	3	0	[Co05], [OYP05], [WDL04]
IBM00	3	0	[Ba02], [Lu05a], [Sc05]
LRS02	3	0	[Fu05], [KZ05], [SW04]
Li03	3	0	[CBD05], [JW05], [ZY04a]
Li04	3	0	[Ch05a], [Sn06a], [Sn06b]
SG96	3	0	[KM04], [KMM06], [PL03]
SH05	3	0	[Gr06a], [HS05], [Ts05]
Za87	3	0	[Jo05], [OYP05], [St05]
Ze03b	3	0	[OY05a], [OYP05], [SW04]

Figure 3.7: Number of ingoing and outgoing references

is, how often an article is referenced by other authors. For this purpose, every reference from a primary reference to another article covering SOA, was stored as described in section 3.1.1. This information was then transformed into an ordered list, containing the number of incoming and outgoing references for each article. Figure 3.7 shows the articles with more than three ingoing references, along with the number of outgoing references and the labels of the incoming references. This helps identifying articles that were considered important by other authors. The list of labels of the outgoing references was omitted, since only a few articles contain outgoing references and these references can easily be looked up in the relevant articles.
Looking at the results, the three references with the most ingoing references are specifications in the Web service area. This shows, that many articles

focus on Web services in association with SOA. On the other hand, specifications are often referenced, when the technology is mentioned within an article, as it is the best way for providing comprehensive information about that technology. Another influence on the references that are given in an article is the academic background of an article. Academic articles and conference proceedings often contain a lot of references, whereas reports for example by Gartner seldom reference other articles. When collecting the references, it became obvious, that often references were provided to articles that deal with the domain, an SOA should be deployed. References to basic SOA issues on the other hand were sparse. Another driving force for the references that are made in the articles is the origin. Articles that are written by organizations, mainly tend to reference other articles from that organizations. For example, many articles that are published by IBM, include a high percentage of references to other articles that were published by IBM. Furthermore, the fact that many references are made to papers by IBM, suggest the assumption, that SOA lacks a common foundation. These issues may limit the conclusions that can be drawn from this statistic, but still, the indication of important articles can be valuable for finding a starting point and further readings.

3.2.6 Referenced specifications

When looking through the primary references in step 1 of the analysis, a lot of references to Web service related specifications were found, that deal with specific problems of Web service architectures. Since these specifications closely relate to topics of an SOA, because Web services can be viewed as a characteristic SOA, they can provide interesting approaches. On the other hand they do not deal with SOA itself, and therefore were mostly not included into the secondary resources. Figure 3.8 gives an overview over specifications that were found during the analysis.

3.2.7 Visualization of references

To give a quick overview over the articles and their relations, a graph was built that contains all the references found during the analysis. Primary references are depicted with a yellow ellipse and secondary references are represented by a white ellipse. Each node also contains the label of the reference. For any article that is referenced by a primary reference, an edge

Specifiaction	URL
Business Process Execution Language for Web Services	http://www-128.ibm.com/developerworks/library/specification/ws-bpel
SOAP Messaging Framework	http://www.w3.org/TR/soap
The Universal Description, Discovery, and Integration	http://www.uddi.org
Web Service Choreography Interface (WSCI)	http://www.w3.org/TR/wsci
Web Services Architecture	http://www.w3.org/TR/ws-arch
Web Services Architecture Requirements	http://www.w3.org/TR/wsa-reqs
Web Services Conversation Language (WSCL)	http://www.w3.org/TR/wscl10
Web Services Description Language (WSDL)	http://www.w3.org/TR/wsdl
Web Services Transaction (WS-Transaction)	http://dev2dev.bea.com/pub/a/2004/01/ws-transaction.html

Figure 3.8: Referenced specifications

is drawn, from the primary reference to the referenced article. Figure 3.9 shows the complete graph, containing all the primary and secondary references. Unfortunately, this graph is very widespread and confusing. This is based on the fact, that over 400 resources are depicted in the graph. Since the layout is done by the graphviz tool, without any detailed influence on the layout of the nodes and edges, some of the edges are overlapped by the nodes, which decreases the readability of the graph. Nevertheless, overlapping of the nodes was avoided by certain parameters of the layouter. For a detailed view on the graph, please refer to the appendix.
Figure 3.10 shows typical primary and secondary references. The primary reference [TYL05] contains references to the secondary references [LT03], [Ta03], [CN04], etc. In addition it also references [CHT04], which was already selected as a primary reference. Due to the approach that was chosen to build the graph, each secondary reference must at least be associated with one primary reference, the one that cited this particular secondary reference.

Some of the chosen primary references do not contain references to other articles or at least do not contain references that deal with SOA related topics. When looking at the graph that was generated, those articles can be easily found as they do not contain any edges pointing to other references, like [GMK98] in figure 3.11. Unfortunately, these articles do not contribute to the search for secondary references that was performed in the second step of the analysis, but on the other hand they also contain valuable information for the keyword analysis. Furthermore, these articles are often journal articles, that give a compact overview over the topics covered within

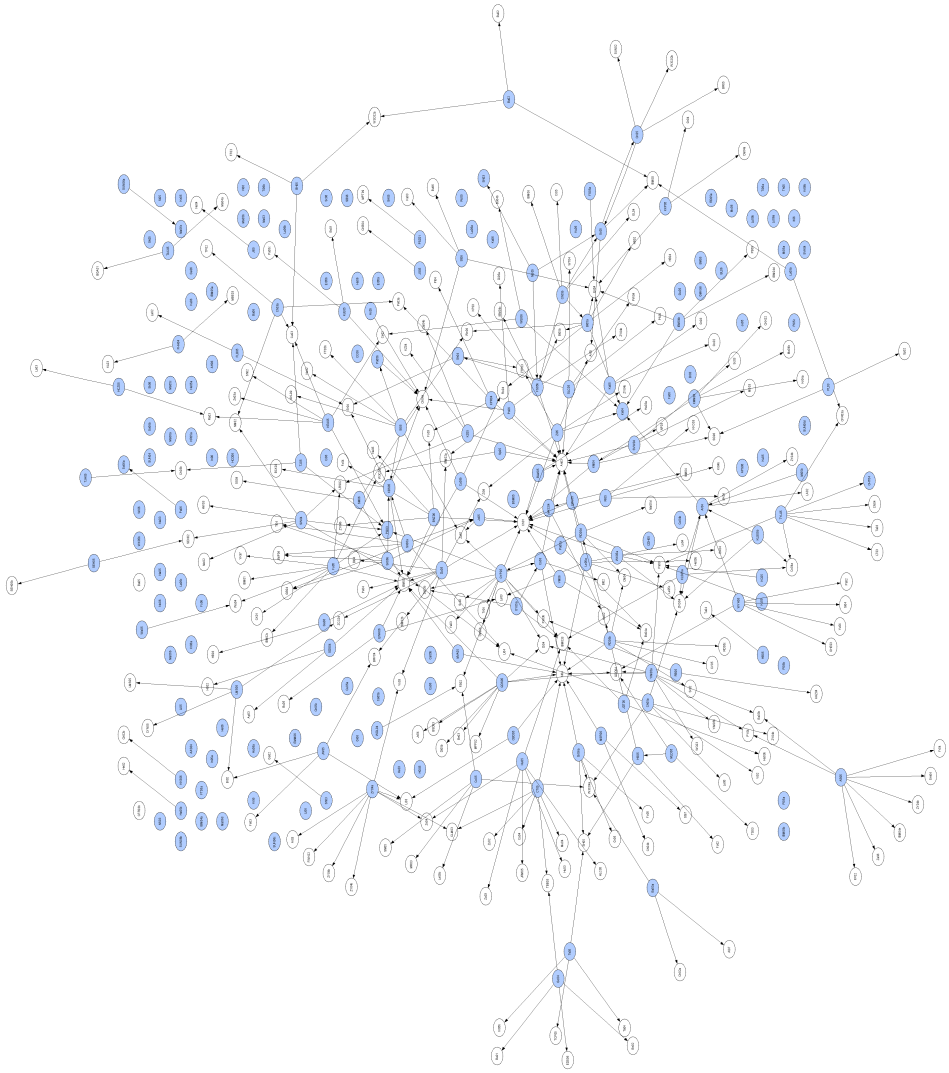


Figure 3.9: Complete Graph

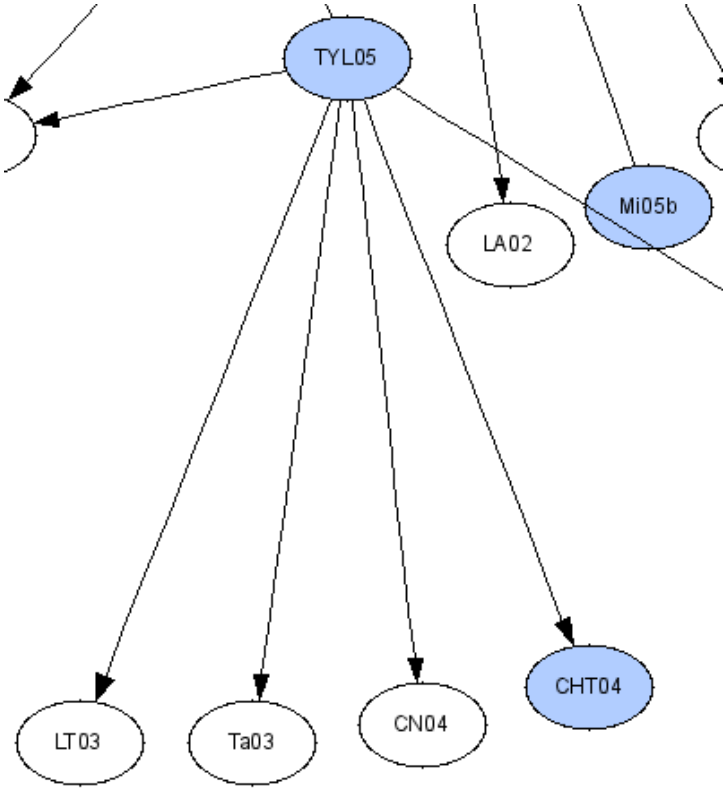


Figure 3.10: Primary and secondary references in the graph

the article. Some of them are also reports from consulting companies that broaden the view on SOA and reflect the problems and risks of applying an SOA in concrete scenarios.
The graph also visualizes articles that are often referenced, through a multitude of edges pointing to the article, as shown in figure 3.12. As already discussed, some of these references are specifications that are often referenced throughout the articles. For example the reference to [An03] in figure 3.12 describes the specification of the Business Process Execution Language for Web Services (BPEL4WS). Another group of articles with one of the most ingoing references are articles that are published by software vendors like IBM. This indicates the problem of missing standards and basic work in the field of SOA.

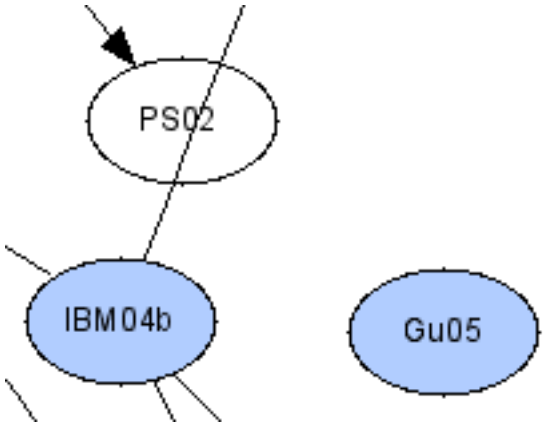


Figure 3.11: Nodes containing no references to other articles

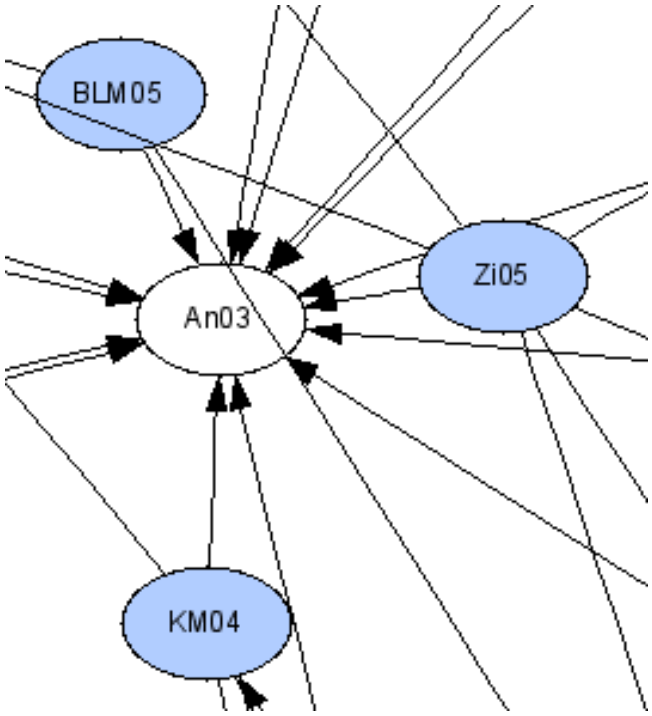


Figure 3.12: An article with many ingoing references

3.3 Topics of the articles

In the last step of the analysis, the available papers were searched for keywords, describing the content of the articles. These keywords where chosen with reference to the topics of an SOA which have been described in section 2.4. For a better overview they were divided into business and IT related keywords. For each keyword that was analyzed, the following sections show a list of the top ten articles, that contained the most hits in the keyword search. In case that there are several articles with the same number of hits than the article rated at position ten, the list was expanded with these articles. In total, 341 articles were included into the keyword analyzis, being either PDF or HTML documents as described in section 3.2.1. When looking at the numbers provided in this section, it has to be taken into account, that these are absolute figures, meaning that they are not independent of the size of the searched articles. To make them relative and more comparable, they could be put in relation to the number of pages of each article. Nevertheless, this would not necessarily improve the significance of these numbers, since one paragraph dealing with the topic in detail would get a bad rating in a large document. Therefore, absolute numbers were considered the best approach and are provided throughout the following results.

3.3.1 Business related keywords

- **Business Value**
  The term *business value* was searched in order to find articles that deal with the benefits of an SOA in business terms. The term was searched case insensitive over all available articles and found **130 times** in **48 documents**. The results shown in figure 3.13 contain mostly journal articles and papers from companies.

- **Governance**
  To find appropriate articles dealing with *governance*, this term was searched case insensitive throughout the articles and found **614 times** in **60 documents**. More than half of the occurrences were found within the first four articles listed in figure 3.14. The high number of documents dealing with governance shows the relevance of this topic, especially when considering, that many of the articles are exclusively dealing with technical issues. Most of the articles that are listed in figure 3.14 are published by companies like Gartner and IBM. Only Two of them can be considered to have an academic background, namely

Label	Title	#
Ti04	On the business value and the technical challenges of adopting Web services	19
Gr03	An Analytic Approach for Quantifying the Value of e-Business Initiatives	13
RD05a	Managing End-to-End Lifecycle of Global Service Policies	7
Ch05b	Impact of service orientation at the business level	6
Sn06b	Integrating legacy Software into a Service oriented Architecture	5
Ca05b	Service-Oriented Enterprise: How To Make Your Business Fast, Flexible and Responsive	4
Fe06	Findings From the 'All Company' Research Meeting: What's the Value of Service-Oriented Architecture?	4
KLS06	Service-oriented agility: Methods for successful Service-Oriented Architecture (SOA) development, Part 1: Basics of SOA and agile methods	4
Sk05	Service Oriented Architecture Realizing the Business Value	4
So06a	A new Service-Oriented Architecture (SOA) maturity model	4

Figure 3.13: Number of occurrences of the term *Business Value* in references

Label	Title	#
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	142
Mi05b	A case for SOA governance	89
Wi06	SOA Governance: Rules of the Game	48
Ma06c	Service-Oriented Architecture Craves Governance	43
So06a	A new Service-Oriented Architecture (SOA) maturity model	20
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	20
SUN06a	SOA Governance Solution - Accelerating SOA	19
Ma06b	The ICC and SOA Governance	18
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	16
KR05a	10 Steps to SOA	12
KKP05	Reckoning Legislative Compliances with Service Oriented Architecture - A Proposed	12
Ku05	Hot Spots der Software-Entwicklung: Service-orientierte Architekturen - Anspruch und Wirklichkeit	12
Mo06	Web Services Governance	12

Figure 3.14: Number of occurrences of the term *Governance* in references

[Ku05] and [Bi05a]. One of the reasons for this could be, that academic research is more focused on technological groundwork than on business relevance and problems that arise in the adoption of this technology within an organization.

- **Alignment**
  Another interesting topic is the influence of an SOA on the alignment of business and IT within an organization. To find relevant articles, the term alignment was searched case insensitive, resulting in **93 hits** within **44 documents**. Since the term alignment was searched without any reference to business or IT, the resulting hits could also contain the term in another context. The results in figure 3.15 show, that the term is seldom used within the articles, because the article with the most occurrence of this term only mentions it six times. In average al-

Label	Title	#
SUN04	Assessing Your SOA Readiness	6
Jo04	Concepts for Modelling Enterprise Architectures	6
Mi05b	A case for SOA governance	5
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	5
Ku05	Hot Spots der Software-Entwicklung: Service-orientierte Architekturen - Anspruch und Wirklichkeit	5
Sn06b	Integrating legacy Software into a Service oriented Architecture	4
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	4
OYP05	A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration	4
KLS05	Service-Oriented Agility: An initial analysis for the use of Agile methods for SOA development	3
St05	Aligning Business and IT: The Process-Driven Architecture Model	3
FB02	The Web Service Modelling Framework WSMF	3
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	3
KKP05	Reckoning Legislative Compliances with Service Oriented Architecture - A Proposed	3
OY05b	Establishing and Maintaining Compatibility in Service Oriented Business Collaboration	3

Figure 3.15: Number of occurrences of term *Alignment* in references

most each of the 44 articles mentions this keyword twice. This leads to the conclusion, that this topic is mostly just mentioned in the respecting articles without a detailed discussion. But papers that contain this keyword can be considered to dealing with the business relevance and benefits of an SOA.

- **Return on Investment (ROI)**
  One of the most interesting questions when planning to introduce an SOA within a company is, what the monetary benefits of an SOA are, compared to the investment that has to be made for the deployment and maintenance of an SOA. Return on investment (ROI) describes the measure that shows how effectively the capital is used to generate profit. Therefore, the term *return on investment* was searched case insensitive throughout the articles. Since the abbreviation *ROI* is often used, it was also searched and the resulting hits were added. This results in **132 hits** within **53 documents**. Looking at the results of figure 3.16 it has to be taken into account, that the method of searching both terms, return on investment and ROI can result in a total number of hits that does not truly reflect the number of occurrences. Usually when a new term appears in an article, both the full term and the abbreviation are mentioned together, to introduce the abbreviation. However, the results show, that the topic is realized in the articles, although the total number of papers, at least dealing with this issue, should be significantly higher. Especially when considering the importance of this issue for decisions about the introduction of an SOA.

Label	Title	#
Sk06	SOA Starts With Design and Delivery	18
BM05	Learn the Key Success Factors for SOA Deployments	9
FS05a	Service Discovery Based on Business Processes: A Practical Approach Using UML	8
HB04	Reuse Is the Key to SODA ROI Model	6
Li05	The ROI of Your SOA	6
Da04b	What you need to know about service-oriented architecture	4
GMK98	Disaggregating the ROI to IT capital	4
In05a	Data Integration in a Service-Oriented Architecture	4
Mi05a	Business-driven development	4
Ti04	On the business value and the technical challenges of adopting Web services	3
So06a	A new Service-Oriented Architecture (SOA) maturity model	3
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	3
CHT04	Migrating to a service-oriented architecture	3
Ha05a	Service-Oriented Architecture for the Pragmatist	3

Figure 3.16: Number of occurrences of term *Return on Investment* in references

Label	Title	#
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	54
He05b	Semantic Business Process Management: Using Semantic Web Services for Business Process Management	40
Ko05	Dynamic Policy Management on Business Performance Management Architecture	40
Ti06a	Business Process Management on an SOA Foundation	34
ZKG04	Elements of Service-Oriented Analysis and Design	33
Ro05	Der optimale Weg zur Geschäftsagilität: SOA oder BPM?	32
Gi06	The Difference Between IT and Business Enablement	30
Ka05	Analysis and simulation of business solutions in a serviceoriented architecture	27
AHZ04	Cross-Enterprise Business Process Management Architecture - Methods and Tools for Flexible Collaboration	24
LRS02	Web services and business process management	18

Figure 3.17: Number of occurrences of term *Business Process Management* in references

- **Business Process Management / BPM** As described earlier in section 2.4, BPM is a topic, that is not directly linked with SOA. Nevertheless, an SOA can support BPM with its ability to quickly adopt to changes and its close alignment to business processes. A search for the terms Business Process Management and the abbreviation BPM resulted in **574 hits** within **98 documents**. This result shows, that many articles dealing with SOA, at least mention the concept BPM in this context. Most of the hits are created by the articles in the top ten list and some of the occurrences in the articles might be produced by the complete term in combination with the abbreviation. Nevertheless, it seems obvious, that there is a lot of interest in using SOA

Label	Title	#
Ke04	Patterns: Implementing an SOA Using an Enterprise Service Bus	500
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	249
En04	Patterns: Service-Oriented Architecture and Web Services	39
Ke06	Patterns: Extended Enterprise SOA and Web Services	36
So06a	A new Service-Oriented Architecture (SOA) maturity model	16
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	16
Ma06a	Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus	14
SAB05	SOA Maturity Model, der Leitfaden für die Einführung einer Serviceorientierten Architektur	12
Ro04	Understand Enterprise Service Bus Scenarios and Solutions in Service-Oriented Architecture	8
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	7
Pa06	Event Strem Processing and Event Driven Architectures	7
Co04b	SOA - So What? How SOA Expands the vision of Web Services	6
Li06	IT-Riesen kämpfen um die SOA-Krone	6
Sc03	Hype Cycle for Application Integration and Platform Middleware	6

Figure 3.18: Number of occurrences of term *Enterprise Service Bus* in references

to solve BPM related issues.

3.3.2 IT related keywords

- **Enterprise Service Bus**
  In section 2.1 the abstract concept of an service bus was presented, and later on in section 2.4 a concrete instance, the *Enterprise Service Bus (ESB)* was introduced. This concept for communication within an SOA has been encountered multiple times when screening the articles and was therefore included into the keyword search. The term was found **1016 times** within **67 documents**. More than 80 percent of all hits are generated by the first four articles in figure 3.18, which are all published by IBM. The reason for this high number of hits within these articles is on the one hand, that they are all substantially large and on the other hand, that IBM offers an ESB within one of their solutions. Still, the large number of 68 documents that contain this term also shows, that this topic attracts some attention throughout the articles.

- **Service Layer**
  A special layer for services in the architecture of applications, should help improving reusability and accessibility of services. to find articles dealing with the realization and technical details of such a *service layer*,

Label	Title	#
En04	Patterns: Service-Oriented Architecture and Web Services	6
RD05d	Design and Implementation of a Service-Oriented Business Rules Broker	6
BSD03	The Self-Serv environment for web services composition	4
Du05	SOA Without Web Services: a Pragmatic Implementation of SOA for Financial Transactions Systems	4
JDT05	A Service Oriented Architecture Framework for Collaborative Services	4
St05	Aligning Business and IT: The Process-Driven Architecture Model	3
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	3
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	2
CK05	Management of the serviceoriented-architecture life cycle	2
Gu05	Service-Oriented Grid Architecture and Middleware Technologies for Collaborative E-Learning	2
KMM06	Efficient Exploration of Service-Oriented Architectures using Aspects	2
LT03	Dissecting Service-Oriented Architectures	2
Pa05c	An Introduction to Service-Oriented Architecture from a Java Perspective	2
PG03	Service-Oriented Computing	2
RD05b	Towards a Distributed Service-Oriented Business Rules System	2
Ti06a	Business Process Management on an SOA Foundation	2
WK05b	SOA: EAI in neuen Schläuchen?	2
ZCC04	Migration to web services oriented architecture: a case study	2

Figure 3.19: Number of occurrences of term *Service Layer* in references

this term was searched within the primary and secondary references. Thereby, the term was found **60 times** within **26 documents**. The results shown in figure 3.19 shows, that the term is seldom used within the articles and only few contain this term several times. One of the reasons might be, that the articles do not explicitly mention this term, but still deal with its concept. Another reason could be, that this term is not very commonly used, or not considered important by the authors. Anyhow, the list shows articles that deal with that topic, even though, it seems like none of them provides detailed information about the technical realization.

- **Web Service**
  To find articles dealing with Web Services as a concrete realization of an SOA, the term *Web service* was included into the keyword analysis. Not only articles that deal with technical details, but most of the article that were searched, contained this term. Thus, the term Web service was found **11277 times** within **275 articles**. The spelling of the term Web service was found in different variations throughout the articles, e.g. *Web-service*. These different variations were not included into the search, so the number of articles that contain any variation of the term Web service would almost reach the number of the documents

Label	Title	#
Ke04	Patterns: Implementing an SOA Using an Enterprise Service Bus	421
Ke06	Patterns: Extended Enterprise SOA and Web Services	351
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	328
En04	Patterns: Service-Oriented Architecture and Web Services	309
Cu02	Unraveling the Web Services Web	116
CKB04	An introduction to the Web Services architecture and its specifications	96
Fu05	A Study of Service Composition with QoS Management	92
TP02	An Overview of Standards and Related Technology in Web Services	82
HW03	Web services: Foundation and composition	57
Zi04	Second generation web services-oriented architecture in production in the finance industry	51

Figure 3.20: Number of occurrences of term *SOAP* in references

searched. Since the results have produced such high numbers of hits, they were not further analyzed and not added in detail to the articles. The significance of such an analyze can be neglected.

- **SOAP**
  The term *SOAP* was included into the keyword search to find articles that are very affine to development of Web services as a concrete implementation of an SOA. Especially articles with many occurrences of this term can be considered dealing with technical details of Web service implementation. The term SOAP appeared **3000 times** in **181 documents**. Furthermore, the result in figure 3.20 shows, that almost half the hits are generated by the *Patterns* series published by IBM redbooks[20]. These documents are comprising more than 300 pages and deal with technical details of Web services, which drastically increases the numbers of hits in the results. Nonetheless, almost one half of all documents searched mention the term SOAP. Most of these articles can be considered dealing with technology related issues, but also papers which focus on the business related issues tend to at least mention this technology.

- **WSDL**
  Another term that comes across when working with Web services, is *WSDL* as described in section 2.1.2. Much like SOAP, this term should be used within papers, dealing with technology related issues of an SOA, in the context of Web services. The term was searched case sensitive, to avoid additional hits from code examples, that contained the term WSDL as part of the XML structure. When searching for WSDL, the outcome was very similar to the search for the term SOAP,

[20] http://www.redbooks.ibm.com

Label	Title	#
En04	Patterns: Service-Oriented Architecture and Web Services	286
Ke06	Patterns: Extended Enterprise SOA and Web Services	235
Ke04	Patterns: Implementing an SOA Using an Enterprise Service Bus	212
Di05b	Model-Driven Engineering of Web Service Compositions: A Transformation from ISDL to	190
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	98
Si04	Framework for Semantic Web Process Composition, Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce	66
TP02	An Overview of Standards and Related Technology in Web Services	44
Pe03b	Web services orchestration: a review of emerging technologies, tools, and standards	39
Fe03	Secure, Reliable, Transacted Web Services: Architecture and Composition	34
Zi04	Second generation web services-oriented architecture in production in the finance industry	34

Figure 3.21: Number of occurrences of term *WSDL* in references

as shown in figure 3.21. **2298 hits** were counted within **188 documents**. The four IBM redbooks were again within the five best rated articles and generated more than 30 percent of the hits. The reason for this result can be related to the fact, that SOAP and WSDL are often mentioned together. Unfortunately, this result does not provide any further insights than the results of the search for SOAP, but shows the close relation of these two technologies.

- **Case Study**
  Finally the term *case study* was searched in order to find articles that show concrete examples of how an SOA can be implemented and deployed as well as some experiences on existing projects. This term can not clearly be assigned to the IT related keywords, since case studies could just as well be used to show the business perspective. Anyhow, uses cases are often used to show development projects. The search resulted in **252 hits** within **54 documents**. Another drawback of this keywords gets obvious when looking at the results in figure 3.22. The article containing the most hits, describes basic software architecture issues with no concrete relevance for SOA. This is founded in the fact, that the term case study could not be linked with SOA when performing the search.

Label	Title	#
Al97	A Formal Approach to Software Architecture	74
ZCC04	Migration to web services oriented architecture: a case study	18
KPR04	A Framework for Integrating Business Processes and Business Requirements	11
CFR03b	Improving Dependability of Service Oriented Architectures for Pervasive Computing	10
LMS06	Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture	10
Kr04b	Service specification with MSCs and roles	9
Pe03b	Web services orchestration: a review of emerging technologies, tools, and standards	8
KJE05	Service-Oriented Software Engineering (SOSE) Framework	8
Ma05c	Transformation of UML Models for Service-Oriented Software Architectures	7
KM04	Systematic Development and Exploration of Service-Oriented Software Architectures	6
KMM06	Efficient Exploration of Service-Oriented Architectures using Aspects	6

Figure 3.22: Number of occurrences of term *Case Study* in references

Chapter 4

Enterprise SOA

The introduction of an SOA within an enterprise has to deal with a lot of issues, other than just technology related ones. Basic ideas and concepts of an SOA have been depicted earlier in this thesis, now this chapter takes a closer look at an SOA with a special emphasis on the concerns of an enterprise. Thereby, two main perspectives are used to describe relevant issues. The business perspective, on the one hand, concerned about the continuity and the success of the enterprise, and the IT perspective, supporting existing processes in everyday business and enabling new strategies, on the other hand. For both perspectives, a set of articles from the ones selected during the analysis described in chapter 3, were selected to deepen these topics. The goal was to find articles, that provide further details on the topics covered in this thesis. In addition, the view on an SOA the articles illustrate and the basic understanding of SOA will be compared to the basic concepts described earlier in this thesis.
In this chapter, at first the relationship of Business and IT will be described, and approaches will be presented, how an SOA can help to improve this relationship. These topics were briefly touched in chapter 2 and will now be presented in more detail. At second, the articles that were chosen to describe IT relevant topics of an SOA will be presented and their statements will be related to the concepts, benefits and risks of an SOA as described so far. At last, the papers dealing with business relevant topics will be discussed. These articles will also be described in detail together with a critical reflection of their conclusions.

4.1 Business and IT: SOA as a mediator

IT departments are usually not directly involved in the business processes within an enterprise which makes the development of solutions, that fit the companies requirements, a difficult task. Business domain experts specify their requirements with little knowledge about the underlying technologies. These requirements are then implemented within the IT departments, that have to deal with certain technology constraints. In some cases the outcome fails the business needs due to different understandings between these two departments. Besides other potential benefits of an SOA, the improvement in the alignment of business and IT is supposed as one of the key benefits compared to existing concepts. Before describing certain specific issues related to either of these two perspectives, this section takes a closer look at their relationship.

Jonkers et al. [Jo04] describe three architectural layers for describing Enterprise Architectures . These are business layer, application layer and a technology layer. This layering is similar to the layering described in chapter 2, when the topic of a service layer was introduced. The first layer, the business layer deals with organization and processes, focused on the business strategy and goals. The application layer contains the applications that deliver functionalities used to execute business processes. Thereby, the application layer uses the IT infrastructure which is assigned to the technology layer. Like in most layered architectures, "lower layers provide functionality to support the higher layers" [Jo04]. For modeling the functionality within these layers, several different concepts exist, but each of these concepts is only supporting one of these layers. For the business domain, certain modeling techniques like ARIS[1], a set of standards for e-Business with XML called ebXML[2] and the Business Process Modeling Language BPML[3] are used to model organizations and processes. For modeling the application domain as well as the technology domain, e.g. the Unified Modeling Language (UML)[4] has become a widely accepted standard, providing several different concepts for modeling a variety of technical issues.

The article by Jonkers et al. aim at describing a common language for describing "integrated enterprise modeling" [Jo04] (p. 2), which should be able to describe the relationship between the above mentioned layers and

[1] http://www.ids-scheer.com
[2] http://www.ebxml.org/
[3] http://www.bpmi.org
[4] http://www.uml.org/

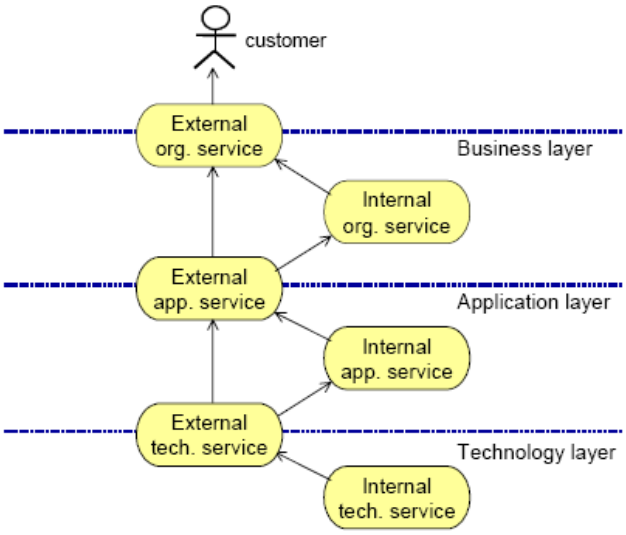


Figure 4.1: Service architecture: hierarchy of services, [Jo04]

their modeling concepts. Thereby they make use of services to describe the relationships among the different layers. These layers form a hierarchy within an enterprise, which allows to look at an enterprise starting from the business processes and activities. [Jo04] (p. 13). Certain actors or roles perform those business activities and processes, and are supported by several business applications. The business activities can "be viewed as services to this business processes" [Jo04] (p. 13). This service-oriented view leads to a hierarchy of services as depicted in figure 4.1. On each layer within this hierarchy, the external services are made available to the layer above. The external services of one layer can depend on both, the internal services of the same layer, or the external services of one layer below. Internal services are used by other services within one layer. Application layer services can use internal services offered by other applications, to enhance their functionality.
The layering of services is similar to the approach that can be found in [Er05a]. Erl defines the business process layer and the application layer, that are connected via the service interface layer, as described in section 2.4. This service interface layer, which is located between the business process layer and the application layer can further be specialized into the following three layers as depicted in figure 4.2:

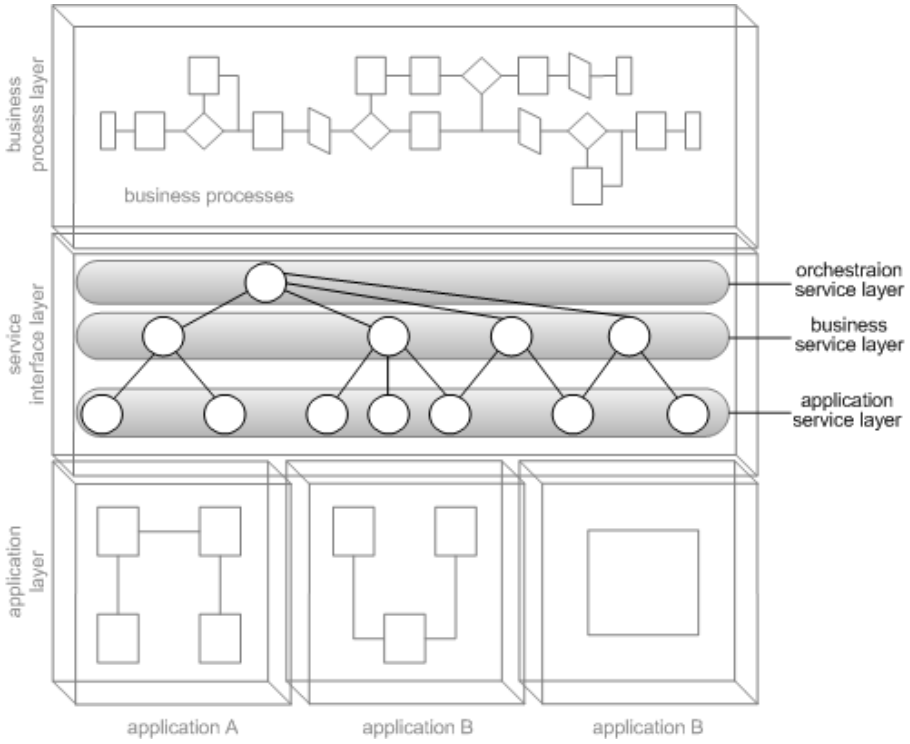


Figure 4.2: Primary service layers as described by [Er05a] (p. 337)

- orchestration service layer
- business service layer
- application service layer

The application service layer provides "ground level foundation that exists to express technology-specific functionality" [Er05a] (p. 337) and can be considered similar to the application layer in the article by Jonkers et al., but in contrast, a technology layer is missing in Erl's model. But since he defines the application service layer to provide reusable functions for processing data, it can be assumed that the functionality of the technology layer is incorporated in the application service layer. The business service layer is equivalent to the business layer in the article by Jonkers et al. Its main purpose is to represent business logic and provide suitable interfaces for expressing business processes. To provide a "parent level of abstraction"

[Er05a] (p. 444), Erl introduces the orchestration service layer. Its main purpose is to manage composition and interaction of other services and encapsulates the linkage of process logic to service interactions. This behavior is expressed in the external services of the business layer of the model described by Jonkers et al.
The layering proposed by the article of Jonkers et al. and the book by Erl is based on the concept of composition that was already described in section 2.2. The key to alignment of business and IT, is the increased abstraction introduced with each layer. It allows business and IT to agree on a common language that includes relevant entities and stakeholders. Jonkers et al. specify those objects, stakeholders and entities in their article and provide a common basis for interconnecting the languages of the individual layers. The resulting descriptions and contracts between business and IT can then be further specified in more detail by the IT departments, but the business needs are clearly expressed and can be understood by the IT departments. On the other hand, constraints and problems of the IT can be comprehended by the business departments, as they are abstracted from technical details. Other characteristics and concepts of an SOA like service descriptions and service contracts contribute to an increased level of alignment between business and IT. Nevertheless, the successful alignment is also a challenging task, that needs to be supported by a proper governance and promotion for the new concept.

4.2 SOA and Business

This section takes a closer look at an SOA from the business perspective and describes certain topics that are relevant from a business point of view during the introduction and usage of an SOA. Due to the high number of references and different topics covered, along with a large diversity of approaches to SOA, three articles that were considerably relevant were selected, to outline this perspective. Apparently, this cannot provide an holistic overview over all the topics involved, but it ensures a certain self-contained approach, instead of a confusing confrontation of different opinions. Nonetheless, the ideas presented in these papers will be critically analyzed and compared to the basics of an SOA that were presented earlier in this thesis.
The information that was gathered during the analysis of the articles described in chapter 3 was used, to choose most relevant articles that deal with the business perspective of an SOA. Thereby, the number of references

to the papers, the number of hits from the keyword search as well as the title of the articles and the year of publication were surveyed, in order to get a good estimation of the content of the articles. Interesting articles were then viewed in more detail by their abstracts and finally, the three most interesting papers were selected.
The first article that was selected is called "Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals" [Bi05a], and was written by N. Bieberstein, S. Bose, L. Walker, and A. Lynch. The article was published in the IBM Systems Journal in 2005. The reasons for selecting this article were manifold. One reason was, that the article was referenced one time by another article and contains seven references to other important articles in the analysis, which leads to the expectation that the article is well founded on existing material. Another reason was the title, which indicates, that the article provides answers to questions regarding the impact of an SOA on the organization and shows solutions for a proper introduction of an SOA, in context with organizational structure and governance. The keyword search resulted in a high number of hits for the term governance and also showed that other business terms are mentioned. In addition, some terms that were considered technical like ESB and service layering were also found. Last, two of the authors of this paper, namely N. Bieberstein and S. Bose were involved in more than one articles included into the analysis, which also increases their credibility.
The second article selected, is called "A new service-oriented architecture (SOA) maturity model" [So06a], and was written by a variety of software vendors. This article was published in 2006, which may be a reason, why this article was not referenced by others, unfortunately this article contains no references to any other work. Nevertheless, one other article that appeared in a technical journal [Ba06], is based on the findings of this article. The main reason for selecting this article was the topic and its promising approach, to value the maturity of an SOA introduction. Moreover, the keyword search showed, that the article deals with any of the business relevant keywords, especially with governance, but also technical keywords are mentioned.
The last article that was selected for the business view on an SOA, is called "On the business value and technical challenges of adopting Web services" [Ti04] and was written by S. Tilley, J. Gerdes, T. Hamilton, S. Huang, M. Müller, D. Smith and K. Wong. The article was published in the Journal of Software Maintenance and Evolution: Research and Practice in 2004. All of the authors have an academic background, and one of them was also involved in three other articles within the scope of the analysis. The article

was further referenced by two other articles in the list of references, which further enhances the credibility and relevance of this article. In addition, this article has the highest number of hits in the search for the keyword "business value" and was rated within the top ten for the keyword "return on investment". Besides this business relevant topics, also the IT specific keywords WSDL and SOAP are used frequently throughout the article.

4.2.1 Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals

The article by Bieberstein et al. [Bi05a] deals with the challenges that the introduction of an SOA induces on an enterprise and shows the impact on different areas of an enterprise. The authors develop a strategy, that enables successful transformation to an SOA. Thereby they tailor some of their ideas to IBM products and integrate them into the solutions. Since those products are of no interest to a general discussion of an SOA, they will be excluded from further discussions, and instead, only the general approach shown in this article will be described. Furthermore, the article often relates to "on demand", which is a term widely used by IBM for marketing purposes, when describing flexible and agile business solutions supported by IT. IBM defines it as a business, that "can respond with flexibility and speed to any customer demand, market opportunity or threat"[5]. Despite promoting own products, the article contains basic information about the transformation to an SOA, that are of general interest.

Transformation strategies

The article proposes several important aspects that have to be considered when transforming the existing IT landscape into an SOA. Basically two approaches are described by the authors: Firstly, to realize short-term business benefits, an incremental approach can be chosen to apply an SOA. According to the authors, the realization of short-term benefits is "currently the primary focus" [Bi05a] (p. 691) for many organizations. Secondly, to make use of the full potential of an SOA, an approach that also "transforms the fundamental layers of an organization" [Bi05a] (p. 691), is necessary. For a successful transition to SOA, a well defined transition strategy is required,

[5] http://www-1.ibm.com/partnerworld/pwhome.nsf/weblook/mkt_bp_ebod.html

that addresses technological, organizational, cultural, and behavioral transformation. This approach is described in the following, dealing especially with IT, organizational, and cultural and behavioral transformation.

IT transformation

At first SOA funding and investment have to be aligned. Therefore, the business value of the benefits of an SOA has to be considered, the ROI regarding the IT efficiencies has to be calculated, and the economic behaviors have to be adapted to support SOA. Especially for the collaboration across departments, that is fostered by an SOA, respective economic directives and common goals have to be introduced. Funding models have to be introduced that allow "business-unit autonomy and enterprise-level directives" [Bi05a] (p. 692) at the same time. Furthermore, directives have to be established, that control the relationship between the IT and business units, to encourage reuse of IT services and provide a set of enterprise wide, common services.
From an IT governance perspective, it is important that enterprise services have defined owners with assigned governance responsibilities. These owner's responsibilities have to be "aligned with the overall enterprise governance" [Bi05a] (p. 693). It is crucial for the success of an SOA, to establish well documented internal technical standards and to ensure that they are followed by the technical teams. The technical community often perceives those standards as more complex and time consuming, therefore, management policies have to be introduced that control the adherence to those standards. Standards for interfaces are furthermore required, to improve decoupling and enable business partner integration and outsourcing opportunities. To allow integration of business partners, it is also important to base enterprise standards on common industry standards, to allow service interoperability. For communication handling between service providers and service requestors in the described framework, Bieberstein et al. propose the use of an ESB as a middleware.
For technology adoption, technology standards can provide templates for projects, to reduce the effort for creating standardized and easy to use services. Conditions, under which certain protocols, methodologies, and deployment platforms are used, can minimize training investments and increase the number of human resources available. The use of common patterns for business, design and architecture can further decrease the development efforts and help building standardized services. Setting up knowledge management systems that allow developers to share their knowledge throughout

the enterprise, further leverages the introduction of an SOA. With the technical adoption of an SOA, some risk is associated. Since SOA is considered a leading-edge technology, an increased need for product support of software vendors or a lack of support for an SOA in the existing infrastructure could increase costs.

Organizational transformation

After the IT transformation, "an organization structure that is perfectly aligned to exploit this new IT service framework is required" [Bi05a] (p. 696). Such an organization structure has to ensure business agility, minimize management overhead and rationalize tasks and associated communication. The transformation of the organization has to be conducted in parallel to the IT transformation. According to Bieberstein et al., such a target organization has to be accomplished by "applying the SOA metaphor to organizational structure" [Bi05a] (p. 697). This means, that tasks and activities within the organization are viewed as units of service, called *team services*, and that employees are assigned to teams, which are providing such a service. These services can then be orchestrated to build high-level tasks. To enable collaboration and coordination of the teams and services, a so called "collaboration-and-orchestration bus (COB)" [Bi05a] (p. 698) is proposed. This can be considered as a logical equivalent to the ESB, as it provides a technical infrastructure for advertising and coordinating team services as well as for monitoring tasks. The COB further offers tools for the design and planning of services, which can be divided into three major tools: the service directory tool (SDT), the asset directory tool (ADT) and the employee directory tool (EDT). The SDT is used to describe and publish all services in an enterprise with service descriptions. These service descriptions are annotated with key characteristics like stakeholders, team members and ratings. The access to the individual services in the SDT is controlled by roles, authorization levels and privacy policies. The ADT manages assets that are produced by services and can be reused. This can be product binaries, technical documents or architectural blueprints. Instead of producing them again with each service request, the ADT provides facilities to house and publish these assets. This minimizes unnecessary communication for asset-producing services. Finally, the EDT lists all employees, together with their personal and professional information. This includes experience and qualifications that can be used to assemble and orchestrate teams dynamically. Furthermore, contact information and linkage to other employees

support human networking and therefore supports "effective teaming and individual growth" [Bi05a] (p. 701).
The above mentioned team services and the COB comprise what Bieberstein et al. introduce as the Human Service Bus (HSB), which is "an optimized organizational service structure" [Bi05a] (p. 698). Services, which execute particular tasks are the central entity within the HSB. For each service, *service agents* are defined, who monitor, mediate or choreograph services. Bieberstein et al. introduce service layers for aggregated and composed services according to their functionality and scope of responsibility. They are team services, departmental services, business unit services, divisional services and group services. A detailed description of the service layers can be found in [Bi05a]. According to Bieberstein et al. the adoption of an HSB provides several benefits. At first, modeling the organization using a service structure and assigning responsibilities provides a better overview and control over the services and removes redundancies, which leads to optimized business operations at reduced costs. At second, through the involvement of customers and partners via collaborative tools, the customer satisfaction and loyalty is increased, leading to a better alignment with the customers. Furthermore, with the integration of external services and possibilities to configure and customize services, business agility is increased. At third, since the services are "geared to support the organizations's tactical and strategic business operations" [Bi05a] (p. 702), new strategies can be designed, reducing opportunity costs. A pool of services that represents the core competencies of an enterprise, can further help to streamline internal business processes. At last, the employees are given more creative freedom in development, since they are measured by service performance and meeting service requirements. Their personal growth opportunities are very flexible. They can choose between specialization, broader responsibilities or developing orchestration skills.

Cultural and behavioral transformation

One problem that is seldom covered in articles and not always properly addressed when migrating to an SOA, is the cultural and behavioral transformation. When change in organization occurs, sociological and psychological perspectives have to be taken into account. From a cultural perspective, it is important, to communicate the changes in detail to the entire company, to speed up the organizational transition. Within stable organizations, employees should be given the necessary time to adopt and practice these changes.

It is also important, to point out, that the changes are necessary and cannot be handled with the existing structure. This helps creating a better understanding for the necessity of the change and reduces the rejection of the change. About the impact on the individual behavior, Bieberstein et al. define several human resource considerations [Bi05a] (p. 702):

- **Fostering teamwork:** Organizational change impacts the human network of the employees. To create team unity within new departments or teams, regular meetings and events are necessary.
- **Designing incentives:** Due to the dynamic and impersonal team assignment, proposed by the HSB, new incentives for different personality types have to be created. This can reach from public recognition to monetary awards.
- **Matching roles and skills to services:** To achieve maximum team performance, it is important to form teams with regard to various team roles, in addition to the skills and experience of the team members. This makes it necessary to assess team performance in addition to individual performance within an SOA.
- **Reinforcing new working styles:** To force the employees to use new working styles, strong tools with a benefit for the users have to be provided. The new behaviors must be well understood and required, and they have to be encouraged by business incentives.
- **Managing individuals:** For a successful SOA, it is important to rely on the skills of the employees and to free managers from strictly controlling their employees. Instead only exceptional events that require action, should be triggered and managed accordingly.
- **Designing measurement metrics:** Monitoring ensures the quality of services. To measure the value added by each individual service, a scale of measurement has to be introduced. Along with the ratings provided by the SDT, the performance of each service, within a project can be measured.
- **Designing appraisal and compensation metrics:** To evaluate the individual contribution of any employee to the company success, new metrics have to be established.

The above mentioned human resource considerations arise when an SOA is introduced, and the respecting transformations of IT, governance, and organization are implemented.

4.2.2 A new service-oriented architecture Maturity Model

The article [So06a] that was written by various software vendors, describes a new approach to assess the stage of an SOA adoption within an organization. Therefore, they introduce an SOA Maturity Model (SOA MM), which is similar to the Capability Maturity Model Integration (CMMI), and shows a roadmap towards a successful adoption of an SOA. "CMMI is a process improvement approach that provides organizations with the essential elements of effective processes" [SEI06]. According to the authors, the article provides a framework that shows the path to an successful SOA adoption, provides a common vocabulary to support business and IT alignment and illustrates possible business benefits. Although, this paper is interspersed with references to the solutions offered by the software vendors that published the paper and lacks academic background, the conceptual model offers a good overview over the relevant actions required for introducing an SOA.

View on SOA

The authors of this paper also describe their understanding of an SOA. They describe SOA as "an approach to designing, implementing, and deploying information systems, such as that the system is created from components implementing discrete business functionality" [So06a] (p. 3). The components that are mentioned in this definition are later on described as services. This definition of an SOA is very general and does not deal with concepts that are specific to an SOA, except the usage of services. Similar definitions can often be found for service-orientation as a design concept. The authors also provide some key factors, about the specialties of an SOA. At first they claim that "SOA is built on standards of the World Wide Web" [So06a] (p. 3). According to the characteristics of a basic SOA, gathered in this thesis, an SOA is not necessary implemented using Web technology and therefore this point only applies to an SOA that is implemented using Web services. At second, they describe services as loosely-coupled which leads to increased flexibility, reuse and recombination of services. These characteristics conform to the characteristics as described in chapter 2. At third,

the authors claim, that "SOA best practices create designs which embody business processes" [So06a] (p. 3), which is said to support the ability to outsource and extend processes. This point, has not yet been covered in this thesis, since it is a rather weak argument that is not specific to an SOA. Other design concepts also provide best practices that allow a targeted use for special problems. Furthermore, the issue of best practices is often used by software vendors, that advertise their solutions as best practice, when in fact it is difficult to talk about best practices in an area that still is new to the market. At last, it is mentioned that the integration of legacy systems is enabled by an SOA and thus preserves existing investments. The authors also extend these characteristics by several "critical elements of an SOA" [So06a] (p. 5) which lead to asset reuse and fast respond to changes. At first, services are claimed to reflect business activities, especially coarse-grained services. This is in fact a critical desing imperative when building service and has already been described in section 2.2.4. At second, the composability of services as a method to create new services is mentioned which is also described in the before mentioned section. At third, the platform independence and location transparency are described to allow heterogeneity. As SOA was described earlier to be independent from concrete implementations and abstracts from technology this corresponds to the description of basic SOA in this thesis. Location transparency is further achieved by utilizing a service repository, that allows the dynamic assignment of services. Finally, the authors demand services to have an interface and to be message-oriented. Again, these are very basic concepts of an SOA and have been stated in the context of basic SOAs in chapter 2. Throughout the description of the different levels of the maturity model, the authors often use Web service technologies for describing the technologies that have to be introduced. This might be founded on their practical experiences, but their conclusions can also be applied to SOA in general.
Altogether, the view on SOA as it is described in the article, basically corresponds to the notion of an SOA as described in this article and found common throughout the literature. Nevertheless, little differences can be found and it again becomes obvious, that the understanding of an SOA, is common, but not standardized and well defined.

SOA Maturity Model

The SOA MM described in the article shows goals for an SOA introduction at various steps and provides guidelines on how an SOA can positively influence

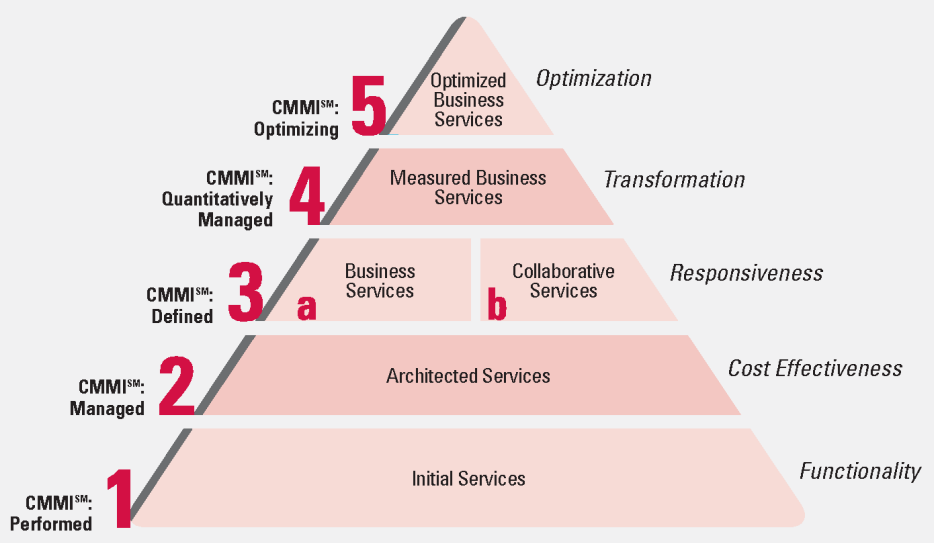


Figure 4.3: SOA Maturity Model levels with key business impact by [So06a] (p. 7)

an enterprise. Figure 4.3 shows the different levels of the SOA MM compared to the levels that are described in the context of CMMI. Each of the described levels also shows the key business impact. Lower levels are preconditions for achieving higher levels of maturity. For example, only when all requirements of level 1 are accomplished, it is possible to reach level 2. Subsequently, the levels are described in detail, according to their definition in the original article.

- **Level 1 - Initial Services** This level represents the first stage of an SOA adoption that is characterized by an initial learning about the technology and initial projects for adopting an SOA. It is used for testing SOA in an laboratory environment and to acquire new skills. According to the authors, "the initial introduction of SOA is driven by the application development organization" [So06a] (p. 10). Along with this first steps in SOA adoption, attempts to quantify the ROI of an SOA should be made. The key benefits for the enterprise at this level are to provide required business functionality and to learn about developing and deploying basic SOA based applications. The authors also suggest to already introduce additional infrastructure components

at this level, to put scalable technology in place to address future challenges. These include a runtime governance service for monitoring and control, an ESB as standard interaction model and a service registry for storing service definitions.

- **Level 2 - Architected Services** At the second level of the SOA MM standards for technical governance are set, under the leadership of the architecture organization. The definition of these standards is based on the learning and feedback of the first maturity level and covers standards for protocols, platforms and policies. According to the authors, the key benefits at this stage are "development and deployment cost reduction" [So06a] (p. 12), based on standard infrastructure and components. The authors propose the usage of further SOA components. A repository should be introduced, which is an extended service registry, that also stores policies and other SOA artifacts like versioning and deployment tools. Further, an extended runtime governance service is mentioned, that identifies services and their dependencies, and thereby supports error detection. Another component is called message transformation service, which transforms messages into the required format and helps integrating services with a different message format. At last, a single sign-on service should be introduced to provide on single point of authentication and authorization accross the organization.

- **Level 3 - Business Services and Collaborative Services** The third level of maturity focuses on the relationship between business and technology. Thereby two complementary paths, that both achieve improved business responsiveness, are possible to achieve this goal. Business Services focus on the improvement of internal business processes, whereas Collaborative Services focus on the improvement of processes for collaboration with external partners.
  To implement Business Services, BPM is important, to manage long-running processes. Another key issue is the enhancement of business processes, which is supported by SOA's ability to easily reconfigure services. Finally, reuse of services, provides the possibility to share common tasks throughout several processes.
  When achieving improved business services through Collaboration Services, it is important to provide options for linking with external partners. Standard SOA protocols, that support business-to-business (B2B) functionality, are one way which is proposed by the authors in

order to achieve cross-enterprise operations. A collaboration service that implements these protocols and manages the transformation between internal and external massages should also be implemented. In addition to standard SOA protocols, it is also necessary to use standard industry protocols for supporting interoperability with external partners.

- **Level 4 - Measured Business Services** Level 4 focuses on measuring and presenting the processes that have been implemented at Level 3. Thereby, a continuous feedback on the business impact and the performance of these processes should be provided. The authors give two examples for key features that can be implemented at this level. A *Real-time Event Stream processing*, to collect events that can be further processed and analyzed, to provide real-time feedback. In addition, *Business Process Visibility* can be used to mange SOA from a business perspective and to gain an overview over business processes along with their supporting services and the underlying IT infrastructure.

- **Level 5 - Optimized Business Services** At Maturity Level 5, the measurements of Level 4 are automatically processed, to optimize business processes. Depending on occurring events, automatic responses are created to build self-correcting processes. According to the authors, enterprises that reach this SOA Maturity Level should be able to "increase revenues, reduce costs and respond to unanticipated events in an automated fashion" [So06a] (p. 20).

Figure 4.4 shows a summary of the above described maturity levels including major benefits, scope, critical technology success factors, critical people and organizational success factors, and relevant standards.

4.2.3 On business value and technical challenges of adopting Web services

The article [Ti04], written by S. Tilley et al. try to give a balanced perspective on the business value and technical challenges for adopting Web services. Although, the term SOA does not appear within the document and the article instead focuses on Web service technology, it was selected, since most relevant findings for Web services can be transfered to SOA as

Maturity Level	Prime Business Benefits	Scope	Critical Technology Success Factors	Critical People & Organizational Success Factors	Selected Relevant Standards
1. Initial Services	New functionality	R&D experimentation, Pilot projects Web site, Portal, Custom integrations, Small number of services	Standards, Legacy Integration	Developers learn service development skills Developer Manager Sponsorship	XML, XSLT, WSDL, SOAP, J2EE, .NET
2. Architected Services	IT cost reduction and control	Multiple integrated applications	Support for heterogeneity and distributed systems, Reliable Messaging, Mediation, Ease of deployment, Database integration, Versioning, Internal Security, Performance management	Architecture group provides leadership, SOA Competency Center CIO Sponsorship	UDDI, WS-ReliableMessaging, WS-Policy, WS-Addressing, XQuery, WS-Security, SAML
3.a. Business Services	Business responsiveness — change business processes quickly and effectively	Business processes across business unit or enterprise	Reuse, Ease of modification, Availability, Business process rules, Event-driven processes, Composite applications	IT Partnership with Business, Partnership across Organizations, SOA Life-cycle Governance, Executive commitment, Event-driven design skills Business Unit Manager Sponsorship	WS-BPEL
3.b. Collaborative Services	Business responsiveness — collaboration with business and trading partners	Services available to external partners, Cross-enterprise	External services enablement, Cross-enterprise security, Translation of cross-enterprise protocols, Long-running transactions		RosettaNet, ebXML, WS-Trust
4. Measured Business Services	Business transformation from reactive to real-time, Meet business performance metrics	Business unit or enterprise, Cross-enterprise	Business Activity Monitoring, Event Stream Processing, Complex Event Processing, Event-driven dashboards and alerts	On-going business process evaluation and response CFO Sponsorship	
5. Optimized Business Services	Business optimization — react and respond automatically	Business unit or enterprise, Cross-enterprise	Event-driven automation for optimization	Continuous improvement culture CEO Sponsorship	

Figure 4.4: SOA Maturity Model levels by [So06a] (p. 7)

well. Especially since the author describes Web services using parts of the basic SOA description, that was shown earlier in this thesis. One goal of this article is "separating marketing hype from business reality" [Ti04] (p. 31), which is also one of the goals of this thesis, as outlined in the introduction. Tilley et al. also use a practical example, to illustrate their opinions. As this is just a theoretical example, with no special business insights, it was not included in further examination.

View on Web services

At first, the authors view on Web service is described, in order to show the differences and commonalities with the understanding of an SOA as it was described earlier in this thesis. The authors refer to the previously mentioned definition of Web services by the W3C, to define Web services. They further mention two different views on Web services. The first, is the role perspective which conforms to the basic SOA description given in section 2.1, depicted by figure 2.1. This again shows the close relation of SOA concepts and Web service concepts, and is another example for the confusion about the usage of these terms. Furthermore, this legitimates the use of the results of this paper in the more general context of an SOA. The second view on Web services provided in this article is a protocol perspective, that shows a protocol stack for the different Web service technologies. It comprises a network, transport, XML messaging, description, and discovery layer, where each of this layer is assigned to Web service standards, e.g. the description layer is assigned to WSDL.
The authors claim, that "Web services do offer a novel approach to engineering and deploying software solutions", especially "for the integration of networked heterogeneous applications across the enterprise" [Ti04] (p. 33). In addition to an identifiable advancement from a technical viewpoint, challenges for developers and users of Web services, like an increased skill set that is demanded from developers as well as increased development costs have to be taken into account to make a sound business decision.

Business value of Web services

According to the authors, "Web services hold the promise of considerable business value" [Ti04] (p. 33). This business value must be balanced with many issues of moving to a Web service architecture respectively to an SOA. This includes issues of cost, return on investment, maintenance and control. The authors refer to two studies that deal with the cost savings of Web service. One of them claims significant cost savings from Web services, whereas the other could not determine whether the expected cost benefits were achieved. This shows the difficulty of assessing the true business benefits of an SOA. Following, two areas of application for Web services are described that can "truly provide business value" [Ti04] (p. 33).
The first application area is Enterprise Application Integration (EAI), since the authors state, that data sharing and information flow within an en-

terprise are a major problem in many organizations. EAI tries to address these problems by integrating components from disparate applications. The problem of EAI can be delineated into three categories: data, control, and presentation integration. Several approaches for EAI exist, but according to Tilley et al. they all fail at the usability, as most of the solutions require extensive programmer involvement. Thus, a lot of knowledge and skills are required, which makes it impossible for end-users to work with. Furthermore, many of the existing solutions are tailored to specific needs, making the systems difficult to manage and hard to understand. Earlier EAI solutions were also vendor specific and did not interoperate well with each other. A lack of commonly available network structure, as it was present in prior EAI applications, can be neglected in Web service environments, as the services can be published over the Internet. In contrast to early EAI applications that focused on data integration, Web services "offer the possibility of full integration along all three vectors of data, control, and presentation" [Ti04] (p. 35). In addition, legacy applications can be wrapped into reusable components and Web services can provide integrated access to these components. This enables just in time integration (JITI), which is "one of the long sought-after goals of information systems research" [Ti04] (p. 35).
The second application area of Web service that is examined in detail, is business-to-business (B2B) e-commerce. In contrast to business-to-consumer (B2C) e-commerce, where end-users access applications, B2B e-commerce focuses on application-to-application communication. Thereby, web services offer network-accessible interfaces that allow applications to communicate with each other over the Internet. This provides a considerable business value, since the potential users for the system can be increased and core competencies can be enabled as B2B services and be provided to other Web services. These services can be centralized within the organization or through outsourcing agreements, promising savings on the IT costs. For the services that are centralized internally, these savings result from improved application quality and consistency. Furthermore, the IT department can improve the control over the IT assets and can "better address the long-term needs of the organization" [Ti04] (p. 36). The services that are outsourced, must not be part of the companies core competencies, which create a competitive advantage. If only those services are outsourced, costs can be reduced due to a higher efficiency and better experiences within the outsourcing organization. Furthermore, experience from other clients and applications of the outsourcing partner can positively influence the own services and applications. Besides the potential benefits of outsourcing Web services, certain potential drawbacks have to be taken into account. This includes a loss of

control over timing and functionality, as well as potential priority problems with the Web service provider, since its priorities "may not be aligned with that of the requestor" [Ti04] (p. 36).

Technical challenges of adopting Web services

To realize the above described benefits and the business value of Web services, some technical challenges must at first be managed. Thereby, Tilley et al. address two main issues, required software engineering skills to implement Web service solutions and the integration of legacy systems.
The first point regarding software engineering skills deals with large amount of knowledge required from developers to work with Web services. Whereas software engineers used to working with other paradigms like object- orientation, mainly focus on algorithms, programming languages, and certain project management skills, software engineers "engaged in Web services development need to know all these things - and more" [Ti04] (p. 37). Many new technologies have to be understood by the developers, to make proper use of theses technologies to create Web service solutions. Especially new areas like distributed component technology, computer security, and Internet standards like XML, which build the foundation for Web services have to be mastered. As Web service development "has the potential to create dramatic change in the nature of software engineering" [Ti04] (p. 38), knowledge which is also broad and deep is required.
Legacy systems are valuable business assets within an enterprise and therefore need to be integrated into the new Web service technology. According to the authors, a typical legacy system is characterized as a monolithic, single-tier and mainframe-based application. Tilley et al. propose a three-step reengineering process to integrate legacy systems into the new technology.

- At the first step, the *decomposition*, an analysis should identify potential components.
- At the second step, the *development*, theses modules are developed into virtual components which can be accessed from the new technology platform.
- At the last step, called deployment, the components are deployed into the network to enable usage in new business context.

This stepwise procedure results in a modernized set of legacy components that can be integrated with the new Web service environment. In addition, to solving the technological problems, it is important to keep in mind, that such a transformation of legacy system inflicts considerable investments in terms of educating the developers and convincing the customers of the benefits. This can especially be difficult due to resistance against a modernization of legacy applications, because the new technology is considered immature and the business benefit is uncertain.

4.3 SOA and IT

Similar to the selection of articles that was done for the business perspective, three papers IT relevant topics have been selected. These papers describe basic SOA concepts that can be used for the development and the deployment of an SOA. Those concepts described in the articles are compared to the basic SOA concepts outlined in this thesis. Furthermore, the selected articles also give some extended thoughts about SOA, that broaden the topics covered so far.
The first article selected, is titled "Service-Oriented Computing: Key Concepts and Principles" [HS05] and was written by Michael N. Huhns and Munindar P. Singh. It was published in the journal IEEE Internet Computing in 2005. Both authors are working at Universities in California which indicates an academic background of the article. The article was selected as a primary reference during the first step of the analysis and additionally referenced by another article. Three referenced papers in this article were further included into the analysis, which shows that this work is well founded on other literature. The keyword analysis only resulted in one hit for the term WSDL and two hits for the term SOAP. Since many of the other technical keywords are describing special concepts, for a description of basic SOA, the terms found are sufficient. Furthermore, the title of the article and a short overview, made this article very interesting and promising, to provide additional viewpoints on a basic SOA.
The second article that was selected is called "Service-Oriented Computing: Concepts, Characteristics and Directions" and was published in the year 2003. The author is Mike P. Papazoglou from a University in the Netherlands. His article was found to be one of the most referenced ones, during the analysis in chapter 3. Altogether, six other papers were referencing this article, which makes especially his view on SOA very interesting, as it can be

assumed to be widely accepted. In addition, Papazoglou contributed to four other papers considered in the analysis, that all cover basic SOA concepts. Together, all his papers included in the analysis, have been referenced 14 times, which makes him one of the most cited authors within the analysis. Since his work is dealing with basic SOA concepts and he is cited many times, his articles seems to be important for finding basic SOA concepts.
The last article that was selected for the IT perspective is titled "Using Architectural Patterns and Blueprints for Service-Oriented Architecture" [St06] and was written by Michael Stal, a member of Siemens Corporate Technology. The article was published in 2006 in the IEEE Software magazine. Since the article was published in April 2006, shortly before the selection of articles for this thesis was made, it is comprehensible, that it was not referenced by any other articles within the analysis. The main reason for selecting this article was the special focus of this article on concrete implementation patterns for an SOA, which was expected to provide further ideas and a more detailed notion about SOA implementation in practice.

4.3.1 Service-Oriented Computing: Key Concepts and Principles

The article by M. Huhns and M. Singh [HS05] deals with basic service-oriented concepts and shows a corresponding engineering methodology. Therefore, they abstract from existing Web service standards, which are "unnecessarily limited" [HS05] (p. 75) and base their work on key service-oriented computing (SOC) concepts. As in their understanding services "provide higher-level abstractions for organizing applications" and "are simply a means for building distributed applications" [HS05] (p. 75), they focus on the concepts for building these applications and the interaction of services. According to the authors an architecture for applications using services is based on three main parts: "a provider, a consumer, and a registry" [HS05] (p. 75). This definition is equal to the SOA description that is depicted in figure 2.1 and commonly referenced to as Web services architecture. The naming of the different parts slightly differs, e.g. the consumer is called service requestor in other references, but the basic parts and connections represent the same concepts.
The article comprises three chapters. The first one deals with different levels of abstraction for SOC, the second chapter describes key elements for an SOA and the last chapter describes a development methodology that is required for engineering an SOA.

Abstractions for SOC

The authors distinguish four cross-cutting levels of abstractions for SOC. The first one is called *Intraenterprise Abstraction Level*, dealing with the interoperation aspects of services within an enterprise. For this level several problems are described. The connectivity among the applications has to be ensured using standard protocols and the components have to understand each other, which can be supported by the utilization of XML. Although, XML can provide a schema for a common syntax, it is not able to provide semantic for the different components. To reconcile the meaning behind the interacting components, developers have to rely on declarative information models, which often have to be constructed at integration time. Another issue concerns the integration of legacy applications, which demands that enterprise policies must be made explicit by an SOA, to simplify the management of the system. Furthermore, the "adherence to interconnection standards is crucial" [HS05] (p. 77), especially for the introduction of new applications. This ensures that the right interface is exposed between existing and new applications. Finally, the authors claim, that SOA provides a solution for all these problems, as it provides abstraction and relates the components to the organization.
The second level of abstraction is the *Interenterprise Abstraction Level*, which is concerned with interoperation of enterprises. In contrast to ad hoc data interchange between enterprises, SOA should lead to more cross-enterprise processes and streamline the interactions between the enterprises. With XML as data format, the processing of data is simplified, although certain problems still have to be solved. One issue that can be enabled by an SOA is dynamic vendor selection. This describes an increased flexibility, which is enabled through the capability to dynamically pick business partners according to any quality-of-service criteria. According to the authors, "SOA provides the ability for interacting parties to choreograph their behaviors" [HS05] (p. 77). This refers to choreography, that was already defined in chapter 2.
The third level of abstraction described is called *Infrastructure Abstraction Level*. This level describes grid services which provide "computing resources as a utility" [HS05] (p. 78). These services are computational resources that can be configured dynamically and are provided on demand by outsourcing suppliers, that offer their computing infrastructure. These modular interfaces based on services can be seen as analogous to Web services. They abstract the infrastructure level of an application.

The fourth level of abstraction is the *Software Component Abstraction Level*, describing the software development approach based on services. Autonomous components with well defined interfaces, as offered by services, help improving software development. Therefore, the use of the abstraction models and the semantic representations that are described by SOC, provide a "semantically rich and flexible computational model" simplifies software development.

Service Oriented Architectures

According to the authors, service oriented computing focuses on architectural models, since "many of the key techniques for its components [...] are already well understood in isolation" [HS05] (p. 78). Thus, the key to a successful SOA lies in a cohesive framework that integrates these components. The authors further specify the following key elements for an SOA:

- **Loose coupling:** Instead of tight transactional properties among components, the components are autonomous and high-level contractual relationships are introduced, to ensure system consistency.
- **Implementation neutrality**: Interacting components are only relying on the interface definition and do not need to know implementation details of each other. This especially refers to programming language independency and platform neutrality.
- **Flexible configurability:** Late binding of components ensures flexibility of the configuration, which can be changed dynamically.
- **Persistence:** Services must exist long enough to handle exceptions and take actions to correct them. Further, they must be able to "engender trust in their behavior" [HS05] (p. 79), which requires a minimum lifetime of the services.
- **Granularity:** The authors demand services to be modeled at a course granularity, that is visible for business contracts, as this reduces dependencies among participants and limits communication. This basically conforms to the discussion in section 2.4.1, but still it is important to also look at the drawbacks of coase-grained services.
- **Teams:** Autonomous parties working on a team to realize development should be considered instead of a central computation. They should "solve problems cooperatively or compete intelligently" [HS05] (p. 79).

All the above mentioned key factors for an SOA are closely related to what was described earlier in this thesis in chapter 2. They show SOA as a standardized way to build information technology.

Engineering an SOA

The authors describe how the development methodology changes using SOA to a service-oriented methodology of development. This methodology "replaces code generation with a combination of service discovery, selection and engagement" [HS05] (p. 79). Thereby, service composition plays a key role for reusing existing parts to create new value. Compared to traditional development methodologies, reuse becomes essential and is not just a convenience. Furthermore, some challenges for compositions these methodologies face are described. One challenge concerns transaction handling of services. Due to the loose coupling of the services, it is a difficult task to build transactions which include these services, to ensure a consistent state across the system. Since there are no specifications available that specify a transaction model, the authors claim, that these are implemented in an ad hoc fashion, which reduces interoperability. Since the appearance of this article some improvement in this area was made and first specifications have been defined[6]. Security is another difficult topic in service-oriented engineering, because many participants are involved, who have very different interaction needs. This may lead to pragmatic solutions and problems with incompatible vocabularies.

4.3.2 Service-Oriented Computing: Concepts, Characteristics and Directions

The article by M. Papazoglou [Pa03], describes basic SOA concepts and further introduces an extended SOA. Services are described as the fundamental elements of SOC. Papazoglou describes the correlation between SOC and SOA as follows: "To build the service model, SOC relies on the Service Oriented Architecture, which is a way of reorganizing software applications and infrastructure into a set of interacting services" [Pa03] (p. 1). This article provides a description of basic SOA, that corresponds to the basics covered in this theses, but also enhances the basic SOA concepts described in the extended SOA. The article is neither focusing on technological details

[6] http://www-128.ibm.com/developerworks/library/specification/ws-tx/

nor covering specific solutions, instead it gives an abstract view on service-oriented concepts and SOA. In addition some extended information about grid services are described later in the article.

Basic SOA

According to Papazoglou, services "constitute the next major step in distributed computing" [Pa03] (p. 1). Therefore, he demands services to be technology neutral, loosely coupled and to support location transparency. He further distinguishes between two types of services: simple and composite services. Where composite services assemble existing services to combine their information and functionality. This helps integrating applications as described earlier in this thesis. The article describes Web services to be a specific kind of service, that uses Internet languages and protocols to expose its features and is implemented using open Internet standards like XML. Web services "share the characteristics of more general services" [Pa03] (p. 2), but since they use the Internet for inter-service communication, they require special attention.
Papazoglou describes SOA as "a way of reorganizing a portfolio of previously siloed software applications and support infrastructure into an interconnected set of services, each accessible through standard interfaces and messaging protocols" [Pa03] (p. 3). This definition reflects most of the points covered in chapter 2. Further, Papazoglou describes the basic SOA as it is depicted in figure 2.1. Service provider, service requestor and service registry "act upon the service artifacts: service description and service implementation" [Pa03] (p. 3). The service provider publishes its service description to the service registry, where the service requestor can find the appropriate service and use the service description to bind with the service provider. Then the service requestor can invoke the service and interact with the service implementation. Service requestor and service provider are logical roles, that can be taken by services, and even allow a service to take both roles at the same time.
The author also specifies the scope of the service functionality. He claims, that a service should represent a complete business function, which can be reused. These business functions encapsulated in a service can be seen as a black box that is described by its interfaces, whereas its concrete implementation is irrelevant for the service requestor. Moreover, the service should be logically decoupled from the service requestor, to support loose coupling and to allow the service to be invoked by any service requestor. The ser-

vice description can be divided into service capabilities, interface, behavior and quality. The service interface description publishes the service signature while the service capability description states the conceptual purpose and the expected results of the service. The service behavior description describes how the service is expected to behave during its execution. At last, the Quality of Service (QoS) description publishes functional and non-functional attributes describing the service quality.
In addition to service deployment as described above, Papazoglou distinguishes service realization as another broad aspect of services. The strategy for service realization, can be chosen among the following alternatives:

- Given the specification of a service, the service is designed and implemented in-house.
- Services can be acquired from a service provider, based on various payment strategies like leasing, purchasing etc.
- After the service is specified, design and implementation of the service can be outsourced.
- Existing legacy software can be accessed using wrappers or adapters. Adapters are similar to wrappers, but in addition they incorporate newly developed code into the software.

Extended SOA

The basic SOA definition "does not address overarching concerns such as management, service orchestration, service transaction management and coordination, security and other concerns" [Pa03] (p. 5). Therefore, Papazoglou introduced the extended SOA (ESOA), which is depicted in figure 4.5. It incorporates the basic SOA definition at its bottom layer called *basic services*.
On top of that layer, the *service composition layer* contains roles and functionality for composite services. *Service aggregators* use basic services to consolidate them to composite services, and acting as a service provider for these composite services. Service clients can use these value added services like any other basic service. Thereby, the composite services can provide the following actions:

- **Coordination:** The composite service manages the dataflow among the component services.

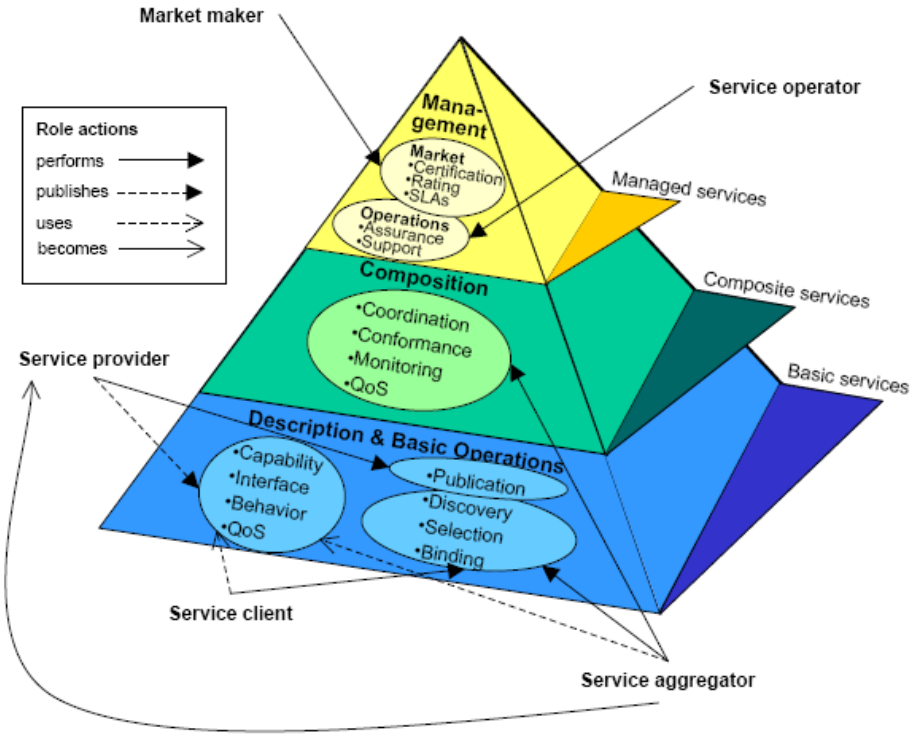


Figure 4.5: Extended Service Oriented architecture [Pa03] (p. 6)

- **Monitoring:** The composite service subscribes to events of the component services and publishes higher-level events.
- **Conformance:** The composite service performs data fusion and imposes constraints on the component services to ensure integrity.
- QoS composition: The composite service derives composite QoS from the QoS of the composite services.

The *service management layer* is the top layer of ESOA framework. It provides managed services to "manage critical applications/solutions and specific markets" [Pa03] (p. 6). In particular, the support for critical applications is provided by the *operations management*. Therefore, it provides performance statistics to asses the effectiveness of services and delivers status notifications for defined conditions. Operations management is thus important to avoid severe risks of service errors. The responsible organization

for operations management is called *service operator* and could either be a service client or a service aggregator. Furthermore, the service management layer also aims at supporting open service marketplaces. Their purpose is to "create opportunities for buyers and sellers to meet and conduct business electronically" [Pa03] (p. 7), or by offering added value services. Therefore, the service management layer includes functionality for market management, to support these marketplace functions. A market maker creates and maintains the marketplace and brings suppliers and vendors together.

Grid services

The functionality of the above described service management can be provided by *grid services*. They provide a high level abstraction model to "allow a collection of services to function as an integral unit and collaborate with others" [Pa03] (p. 7). Grid services are stateful and follow specific conventions to manage collections of service providers and aggregators. Thereby, grid services address critical application and system management concerns. They complement with Web services, as grid services focus on the dynamic discovery and efficient use of distributed computing resources. The Open Grid Services Architecture (OGSA)[7] uses this complementarity and makes the functionality of grid services available using Web services. To provide an abstract infrastructure to enable interaction, aggregation, and coordination of grid services, the logical construct of the service grid bus (SGB) is introduced. The SGB is designed, to provide a management tier and single service connectivity, addressing the following issues:

- **Reach and robustness:** The SGB allows to locate services anywhere and guarantees error free connections to any service. Furthermore, the application should be "robust under transient or long-term failure of one or more service components" [Pa03] (p. 8). This is mainly enabled through asynchronous communications and the support of transparent rerouting in case of failure.
- **Policy and security management:** Services describe their capabilities and requirements via policies, which are a collection of these capabilities and requirements. In addition, it is important to restrict access to the services to authorized stakeholders.

[7] http://www.globus.org/ogsa/

- **Deployment time cost:** The SGB provides facilities to easily aggregate composite services. This process is managed quickly, efficiently and therefore cost-saving.
- **Scalability and performance:** The SGB must perform well, regardless of the performance of the service components or network latencies. Therefore, load balancing, workload distribution and fail-over can be managed by the SGB.

In the application of an open service marketplace, the SGB can be seen as a *business service grid* that "dispatches the best service available from a pool of dynamically assembled service providers" [Pa03] (p. 9). This selection is intended to meet the user's needs and is based on SLAs and QoS constraints.

4.3.3 Using Architectural Patterns and Blueprints for SOA

In his article [St06], M. Stal describes SOA from an architectural perspective, using architectural patterns and blueprints to define SOA. According to the author, "most publications fail to explain SOA, or simply assume it merely defines a synonym for a stack of XML Web service protocols" [St06] (p. 54). Instead of focusing on technology and implementations, he introduces a set of architectural principles, using software patterns. These patterns should help not only to build new applications based on SOA, but also help understanding existing applications. In addition to patterns, also blueprints are presented, "which aren't full-blown patterns but reveal the same properties" [St06] (p. 54).

Driving forces

According to Stal, the main goal of service-orientation is to reduce dependencies between software components. Therefore, services as main entities are introduced, that are accessed by clients. He describes several driving forces that need to be balanced in a service-oriented system:

- **Distribution:** The software environment consists of several components, that are running on different network nodes.
- **Heterogeneity:** Software components reside on heterogeneous environments, that can not be controlled by the service developers.

- **Dynamics:** The environments are highly dynamic, and have to be configurable at runtime.
- **Transparency:** The underlying communication infrastructure should be oblivious to the service providers and consumers.
- **Process-orientation:** Services should be composed to coarse-grained components for the use in coordinated workflows.

When these forces are balanced by following architectural principles, this results in a loosely coupled systems.

Architectural principles

Stal, references to the ternary relationship of an SOA, described in figure 2.1. He claims, that this model "doesn't reflect the differences between standard middleware and the service-oriented approach" [St06] (p. 55). Instead he argues, that this model has to be refined, defining the following aspects in more detail using architectural principles:

- **Interfaces and contracts:** To decouple the service interface from the concrete service implementation, the author suggests the use of the *Bridge pattern* [Ga94] (p. 151). A service interface exposed to the clients, delegates requests to the actual service implementation. This allows changes in the interface or the implementation, without influencing the respectively other part. Distribution and transparency on the client-side can be achieved using the *Proxy pattern* [Ga94] (p. 207). These two example patterns are important for implementing interfaces in a technology agnostic way and with respect to heterogeneity.
- **Communication:** An SOA must support various communication styles, to provide loose coupling and flexibility. Therefore, services communicate asynchronously on the bottom layer through dynamic message routes. Based on this bottom layer several blueprints and patterns can reproduce different communication types, even synchronous communication. Therefore, according to Stal, message exchange within an SOA "doesn't imply that clients and services must interoperate in a message-oriented way" [St06] (p. 57). Instead, it is important for the different peers, to agree upon a common format for message packets. This can be achieved using standard specifications like XML, or agreeing on binary formats.

- **Service lookup and registration:** Dynamic service location is a crucial task in an SOA. Instead of hard-coding locations into the client code, Stal proposes the use of the *Client-Dispatcher-Server pattern* [Bu96]. This pattern introduces the dispatcher as an additional intermediate actor, between clients and services. The dispatcher acts as a repository, where services can register their service and clients can query the dispatcher for services. The dispatcher can further abstract more details from the repository, like the location of the services, to further decouple the clients and services from the repository.

- **State and activation:** To increase loose coupling and scalability, services should be stateless. Thereby, certain optimizations can be implemented like on demand activation of services or service pooling, to increase access time. Some circumstances however can require services to maintain their state across multiple requests, e.g. when services depend on results of previous interactions or when services are bound to concrete instances that represent exact logical entities. For both cases, Stal recommends the usage of additional information that is passed along with each service request, called tokens. These tokens can be used to access state information or to find concrete instances.

- **Processes and their implementations:** Clients in an SOA compose independent services to processes that achieve a common goal. Therefore, the author claims, that a higher layer of abstraction to describe these processes is required and should be provided by a language that supports easy service composition. Since the composition often has to meet additional requirements like transactions, he introduces the *Coordinator pattern* [KJ05], which deals with such issues by introducing a central coordination instance. This pattern could further be extended by the *Strategy Pattern* [Ga94] (p. 315), which allows switching between several coordination strategies.

Outlook

Due to "SOA technologies' instability and immaturity" [St06] (p. 59), the author identified three major deficiencies of an SOA and describes possible solutions:

- **Semantic integration:** Since most of the available SOA technologies "don't sufficiently deal with semantic issues" [St06] (p. 59), Stal pro-

poses to introduce semantical information and processing at several layers, to allow adaptive and self-configuring solutions. The *semantic layer* should provide semantic properties in addition to the syntactic definitions. This could be contextual information, security roles or quality-of-service attributes. A communication layer must then be able to understand and process these additional semantic information. Furthermore, clients, services, and repositories must be enabled to introspect and change this semantic information.

- **Middleware integration:** Adding services to existing applications increases complexity since many functionalities have to be integrated by hand. In addition, "many parts of an SOA application or infrastructure aren't easy to change" [St06] (p. 60). Therefore, the author proposes the use of an SOA middleware, which is often referenced as ESB(see section 2.4.5). This would lead to an integrated solution, which would be configurable and extensible without impacting other components.

- **Further resource types:** Many applications have to access other resource types like databases and files. As they should be hidden from the applications, the author proposes to integrate these resources. To achieve this, he proposes to wrap the access to these resources as services.

Summarizing, the author introduces some basic architectural guidelines for the development of efficient SOA based applications. At first he claims to minimize the communication between client and service. At second, clients and services should be decoupled from infrastructural issues like service discovery. Finally, the developer productivity should be increased, for example by using patterns and blueprints.

Chapter 5

Conclusion

This thesis described the fundamental concepts of an SOA with respect to the different views of business and IT. Thereby, important topics and terms were explained and important characteristics of an SOA were examined. As, many articles dealing with SOA exist and hence many different views and understandings of an SOA can be found, this thesis did not try to provide definitions for an SOA. Instead, a common understanding of relevant concepts was described, as it was found in many articles. To avoid an unpromising discussion about definition details, these concepts were described focusing on only a couple of references that provide a differentiated, yet continuous view on the topic. Nevertheless, many points of current and ongoing discussion have been described, to elaborate issues that need close attention and future refinement. To engross the comprehension about SOA, the described characteristics of an SOA were related to key characteristics of existing and preceding concepts.

After the description of SOA basics, an analysis of articles was made. For this analysis over 400 articles were selected with the goal to find relevant articles dealing with the special characteristics of the two organizational perspectives: business and IT. Furthermore, this analysis should find articles that can be considered as basic work in this area of research. To ensure a broad range of articles and to avoid that important articles were left out, a process consisting of four steps was realized. At first, articles that were considered interesting by the author of this thesis were selected. In the second step, articles that were cited in the selected articles were included into the analysis. In the third step, the entire set of articles was annotated with various attributes, which further describe the articles. Additionally,

the dependency among the articles was depicted, to find articles that can be considered important based on the number of references. In the last step, important keywords were searched throughout the articles and statistically analyzed. The results of this analysis provide an overview over the articles altogether and describe the content of each individual article. This information was used to derive some general conclusions. These conclusions substantiate the initial assumption, that SOA lacks fundamental work describing basic concepts, since no outstanding articles could be found, which could be considered to be a basic reference throughout the articles. Moreover, the understanding of the basic concepts differs among the articles and not even the terminology is consistent. Another conclusion that can be drawn from the results, is that SOA is a very young concept, proven by the fact, that most papers dealing with SOA were published in the last five years.

With the aid of the analysis' results, six articles were selected to describe business and IT relevant issues of an SOA in detail. Especially the alignment of business and IT, was described in detail as well as the possibilities an SOA offers in this area. For each perspective three articles were selected, and their ideas were depicted in detail. Furthermore, these ideas and their view on SOA, were also related to the concepts described earlier in this thesis. The aim of the selected articles was not to provide a holistic overview over all the details involved, instead they were selected in a way, that they address relevant topics in that area, and deepen some of the basic ideas of an SOA. From the business perspective, issues like organizational challenges of introducing an SOA, business value of an SOA introduction, and process guidelines for introducing an SOA were described. The IT perspective, dealt with extended SOA models, SOA engineering, and architectural patterns for an SOA.

The description of basic SOA concepts in this thesis, as well as the topics covered, should provide the reader with a basic understanding of the relevant concepts and act as a starting point for further research in any of the described topics. The results of the statistical analysis can be used to select relevant articles for that purpose.

In the future, the analysis should be repeated, to trace the development of articles around SOA. Thereby, it could be verified, if the trend to an increasing number of articles continues and new articles can be included. Furthermore, the significance of the results can be broadened, by also including references from the articles that were selected in the second step of the analysis. Thereby, more articles which are often cited might be found.

Although, it is almost impossible to predict the further development of SOA,

especially in the current state where the topic is still in hype phase, it seems obvious, that the basic concept has a lot of potential for fundamentally changing IT and business environment and especially the alignment of these two perspectives. Despite the huge potential that lies within the application of an SOA, it is important to consider the risks that are associated with the introduction of an SOA. Therefore, the business value of an SOA has to be measured and the return on investment must be determined.
Currently, many of the SOA projects that are conducted are evaluating the technology. A few, mostly in larger enterprises are already proving the increased business value of this concept, and are delivering first results in form of best practices. Obviously there is a strong demand for more standardization and concepts that abstract from those best practices and vendor specific solutions. When this is accomplished, the technology can be broadly supported and introduction risks minimized. A first step in this direction can be observed by an increasing number of standardizations relating to SOA.
Finally, it is important to develop strategies and concepts, that describe how an SOA can be used in coordination with existing other concepts, to leverage the full potential of an SOA. For example an integration with Event Driven Architectures (EDA) as described in [SN03] might provide significant benefits.

List of Figures

1.1 Gartner Hype Cycle for Emerging Technologies 4

2.1 Basic Service-Oriented Architecture [Pa03] 8
2.2 Artifacts of an SOA, [KBS05] 10
2.3 Levels of loose and tight coupling [KBS05] (p. 47) 17
2.4 UpdateEverything operation encapsulates service composition [Er05a] (p. 301) 19
2.5 Orchestration in dependence on [Er05a] (p. 201) 26
2.6 Choreography in dependence on [Er05a] (p. 208) 27
2.7 The service interface layer, [Er05a] (p.282) 29

3.1 Stepwise procedure of analysis 37
3.2 Disposability of articles 43
3.3 Visualization of disposability 45
3.4 Distribution in time 46
3.5 Authors of articles 47
3.6 Types of references 49
3.7 Number of ingoing and outgoing references 51
3.8 Referenced specifications 53
3.9 Complete Graph 54
3.10 Primary and secondary references in the graph 55

3.11 Nodes containing no references to other articles 56
3.12 An article with many ingoing references 56
3.13 Number of occurrences of the term *Business Value* in references 58
3.14 Number of occurrences of the term *Governance* in references 58
3.15 Number of occurrences of term *Alignment* in references . . . 59
3.16 Number of occurrences of term *Return on Investment* in references 60
3.17 Number of occurrences of term *Business Process Management* in references 60
3.18 Number of occurrences of term *Enterprise Service Bus* in references 61
3.19 Number of occurrences of term *Service Layer* in references . . 62
3.20 Number of occurrences of term *SOAP* in references 63
3.21 Number of occurrences of term *WSDL* in references 64
3.22 Number of occurrences of term *Case Study* in references . . . 65

4.1 Service architecture: hierarchy of services, [Jo04] 68
4.2 Primary service layers as described by [Er05a] (p. 337) 69
4.3 SOA Maturity Model levels with key business impact by [So06a] (p. 7) 79
4.4 SOA Maturity Model levels by [So06a] (p. 7) 82
4.5 Extended Service Oriented architecture [Pa03] (p. 6) 93

Bibliography

[Ap03] Apperly, H. et al.: *Service- and Component-Based Development: Using the Select Perspective and UML*. Addison-Wesley, 2003.

[AAS04] Abrams, C.; Anderson, R. P.; Smith, D. M.: *How Service-Oriented Architecture Will Affect SMBs*, 2004.

[Ac05] Acharya, M. et al.: *SOA in the Real World - Experiences*. In *Service Oriented Computing - ICSOC 2005*, 2005.

[AH06] Abrams, C.; Hill, J. B.: *Achieving Agility: SOA Will Build Organizational Agility, but Watch the Hype*, 2006.

[AHZ04] Adam, O.; Hofer, A.; Zhang, S.: *Cross-Enterprise Business Process Management Architecture - Methods and Tools for Flexible Collaboration*. MIOS, 2004.

[AKL99] Armour, F. J.; Kaisler, S. H.; Liu, S. Y.: *A Big-Picture Look at Enterprise Architectures*. *IT Professional*, 1(1):7, 1999.

[Al97] Allen, R.: *A Formal Approach to Software Architecture*. Phd thesis, Carnegie Mellon University, 1997.

[Al04] Alonso, G. et al.: *Web Services - Concepts, Architectures and Applications*. Springer-Verlag, Berlin, Heidelberg, 2004.

[An03] Andrews, T. et al.: *Business Process Execution Language for Web Services Version 1.1*, 2003. http://www.ibm.com/developerworks/library/specification/ws-bpel (Accessed: 2006-08-09)

[Ar03] Arsanjani, A. et al.: *Web Services: Promises and Compromises*. *ACM Queue*, 1(1):10, 2003.

[Ar04] Arsanjani, A.: *Service-oriented modeling and architecture - How to identify, specify, and realize services for your SOA*, 2004. http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1 (Accessed: 2006-06-22)

[Ar05] Arkin, A. et al.: *OASIS Web Service Business Process Execution Language V2.0 Working Draft*, 2005. http://www.oasis-open.org/committees/download.php/12791/wsbpel-specification-draft-May-20-2005.html (Accessed: 2006-08-07)

[AB05] AI-Rawahi, N.; Baghdad, Y.: *Approaches to Identify and Develop Web Services as Instance of SOA Architecture*. IEEE, 2005.

[Ba02] Baglietto, P. et al.: *Deployment of Service Oriented Architecture for a Business Community*. In *International ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference (EDOC'02)*, IEEE, 2002.

[Ba03a] Baresi, L. et al.: *Modeling and Validation of Service-Oriented Architectures: Application vs. Style*. In *ESEC/FSE'03*, Helsinki, Finland, 2003. IEEE.

[Ba03b] Barry, D.: *Web services and Service-Oriented Architecture*. Morgan Kaufmann, 2003. http://www.barryandassociates.com/managers-guide.html (Accessed: 2006-08-07)

[Ba05] Baghdadi, Y.: *A Business Model for Deploying Web Services: A Data-Centric Approach Based on Factual Dependencies*. 2005.

[Ba06] Bachman, J.: *Model offers measure for SOA success. Networkworld.com*, 2006.

[BCK03] Bass, L.; Clements, P.; Kazman, R.: *Software Architecture in Practice*. Addison-Wesley, 2003.

[BD00] Brügge, B.; Dutoit, A.: *Object-Oriented Software Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, 2000.

[Be01] Bennett, K. et al.: *An Architectural Model for Service-Based Software with Ultra Rapid Evolution*. In *Proc. IEEE Int'l Conf. Software Maintenance (ICSM 01)*. IEEE CS Press, 2001.

[Be02] Bennett, K. et al.: *Prototype Implementations of an Architectural Model for Service-Based Flexible Software.* In *Proc. 35th Hawaii Int'l Conf. System Sciences (HICSS 02).* IEEE CS Press, 2002.

[Be03] Benattallah, B. et al.: *Conceptual modeling of web service conversations.* In *15th Int. Conf. on Advanced Information Systems (CAiSE)*, Klagenfurt, Austria, 2003. Springer.

[BGA01] Buyya, R.; Giddy, J.; Abramson, D.: *A Case for Economy Grid Architecture for Service-Oriented Grid Computing.* In *International Heterogeneous Computing Workshop (HCW 2001).* IEEE, 2001.

[BH05] Bernhardt, M.; Hinz, O.: *Creating Value with Interactive Pricing Mechanisms - a Web Service-Oriented Architecture.* In *International Conference on E-Commerce Technology (CEC'05)*, IEEE, 2005.

[Bi04] Birman, K. P.: *Like it or not, web services are distributed objects. Commun. ACM*, 47(12), 2004.

[Bi05a] Bieberstein, N. et al.: *Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM SYSTEMS JOURNAL*, 44(4):18, 2005.

[Bi05b] Bieberstein, N. et al.: *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap.* Prentice Hall PTR, Upper Saddle River, NJ, 2005.

[BK04] Broy, M.; Krüger, I. H.: *Services and service oriented software architectures - methodological foundations*, 2004.

[Bl02] Bloomberg, J.: *Web Services Management: The Key to Service-Oriented Architectures.* Report, 2002.

[BL06] Bichler, M.; Lin, K.: *Service-Oriented Computing. IT Systems Perspectives*, 2006.

[BLM05] Botta, A.; Lazzerini, B.; Marcelloni, F.: *Integrating service-oriented technologies to support business processes.* In *International Conference on E-Commerce Technology Workshops (CECW'05)*, IEEE, 2005.

[BM05] Barnes, M.; Malinverno, P.: *Learn the Key Success Factors for SOA Deployments*, 2005.

[BNZ06] Bhuiyan, J.; Nepal, S.; Zic, J.: *Checking Conformance between Business Processes and Web Service Contract in Service Oriented Applications.* In *Australian Software Engineering Conference (ASWEC'06)*, IEEE, 2006.

[Bo01] Bond, A.: *ODSI: Enterprise Service Co-ordination.* In *DOA'O1*, Italy, 2001.

[Bo04a] Booth, D. et al.: *Web Services Architecture, W3C Working Group Note*, 2004. http://www.w3.org/TR/ws-arch (Accessed: 2006-08-07)

[Bo04b] Booth, D. et al.: *Web Services Architecture*, 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211 (Accessed: 2006-08-07)

[Br02] Brown, A.; Johnston, S.; Kelly, K.: *Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications*, 2002.

[Br03] Broy, M.: *Service-Oriented Systems Engineering: Specification and Design of Services and Layered Architectures*, 2003.

[BR03] Brown, K.; Reinitz, R.: *Web Services Architectures and Best Practices*, 2003. http://www-128.ibm.com/developerworks/websphere/techjournal/0310_brown/brown.html (Accessed: 2006-08-07)

[BRV04] Birman, K.; Renesse, R. v.; Vogels, W.: *Adding High Availability and Autonomic Behavior to Web Services.* In *26th International Conference on Software Engineering (ICSE'04)*, Scotland, UK, 2004.

[BSD03] Benatallah, B.; Sheng, Q.; Dumas, M.: *The Self-Serv environment for web services composition. IEEE Internet Computing*, 7(1):8, 2003.

[BSM05] Barnes, M.; Sholler, D.; Malinverno, P.: *Benefits and Challenges of SOA in Business Terms*, 2005.

[Bu96] Buschman, F. et al.: *Pattern-Oriented Software Architecture - A System of Patterns*. John Wiley & Sons, 1996.

[Bu02] Burbeck, S.: *The Tao of e-business services*, 2000. http://www-128.ibm.com/developerworks/library/ws-tao/index.html#author (Accessed: 2006-08-07)

[Bu05] The Burton Group: *Service-Oriented Architecture - Developing the Enterprise Roadmap*, 2005.

[Ca03] Casati, F. et al.: *Business-Oriented Management of Web Services. COMMUNICATIONS OF THE ACM*, 46(10):6, 2003.

[Ca04] Canfora, G.: *Software Evolution in the Era of Software Services.* In *International Workshop on Principles of Software Evolution (IWPSE'04)*. IEEE Press, 2004.

[Ca05a] Campbell, D.: *Service Oriented Database Architecture: App Server-Lite?* In *SIGMOD 2005*, Baltimore, Maryland, USA, 2005.

[Ca05b] Capgemini: *Service-Oriented Enterprise: How To Make Your Business Fast, Flexible and Responsive*, 2005.

[Ca06] Carr, D. F.: *Service-Oriented Architecture. Baseline*, April:1, 2006.

[CBD05] Chung, S.; Bylin, Z.; Davalos, S.: *Service-Oriented Development and Integration: Toward Web Services-Based Business Information Systems.* In *International Conference on Web Services (ICWS'05)*, IEEE, 2005.

[CC06] CIO; Computerworld: *CIO and Computerworld Research: The Forecast for SOA*. Report, 2006. http://www2.cio.com/research/surveyreport.cfm?id=106 (Accessed: 2006-06-22)

[CDH05] Chang, E.; Dillon, T. S.; Hussain, F. K.: *Trust and Reputation Relationships in Service-Oriented Environments.* In *International Conference on Information Technology and Applications (ICITA'05)*, IEEE, 2005.

[Ce02] Cerami, E.: *Web Services Essentials.* O'Reilly & Associates, 2002.

[CFR03a] Cotroneo, D.; Flora, C. d.; Russo, S.: *An Enhanced Service Oriented Architecture for Developing Web-based Applications. J. Web Eng.*, 2003.

[CFR03b] Cotroneo, D.; Flora, C. d.; Russo, S.: *Improving Dependability of Service Oriented Architectures for Pervasive Computing.* In *Eighth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2003),*, 2003.

[Ch01] Christensen, E. et al.: *Web Services Description Language (WSDL)*, 2001. http://www.w3.org/TR/wsdl (Accessed: 2006-08-08)

[Ch02a] Champion, M. et al.: *Web Services Architecture, W3C Working Draft*, 2002. http://www.w3.org/TR/2002/WD-ws-arch-20021114 (Accessed: 2006-08-08)

[Ch02b] Chakraborty, D. et al.: *A reactive service composition architecture for pervasive computing environments.* In *PWC '02: Proceedings of the IFIP TC6/WG6.8 Working Conference on Personal Wireless Communications*, 2002.

[Ch03a] Chung, M. J. et al.: *A Service-Oriented Framework for Collaborative Product Commerce.* In (IEEE, Hrsg.): *International Conference on Computer Supported Cooperative Work in Design Proceedings*, 2003.

[Ch03b] Chinnici, R. et al.: *Web Services Description Language (WSDL) Version 1.2*, 2003. http://www.w3.org/TR/wsdl12 (Accessed: 2006-08-07)

[Ch04a] Chappell, D.: *Enterprise Service Bus.* O'Reilly, 2004.

[Ch04b] Chinnici, R. et al.: *Web Services Description Language (WSDL) Version 2.0*, 2004. http://www.w3.org/TR/wsdl20 (Accessed: 2006-08-07)

[Ch05a] Chen, F. et al.: *Feature Analysis for Service-Oriented Reengineering.* In *Asia-Pacific Software Engineering Conference (APSEC'05)*, IEEE, 2005.

[Ch05b] Cherbakov, L. et al.: *Impact of service orientation at the business level. IBM SYSTEMS JOURNAL*, 44(4):16, 2005.

[Ch05c] Charlesworth, I.: *Think business processes for SOA success*, 2005.

[CHT04] Channabasavaiah, K.; Holley, K.; Tuggle, E. M. Jr.: *Migrating to a service-oriented architecture*, 2004.

[Ci06] Cisco Systems: *Building an infrastructure to enable SOA*, 2006.

[CIO06] *CIO.de News*, 2006. www.cio.de/news (Accessed: 2006-05-10)

[CK01] Conti, M.; Kumar, M.: *Quality of Service in Web Services.* In *Annual Hawaii International Conference on System Sciences*, 2001.

[CK05] Cox, D. E.; Kreger, H.: *Management of the serviceoriented-architecture life cycle. IBM SYSTEMS JOURNAL*, 44(4):18, 2005.

[CKB04] Cabrera, L. F.; Kurt, C.; Box, D.: *An introduction to the Web Services architecture and its specifications*, 2004. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/introWSA.asp (Accessed: 2006-08-07)

[CKK02] Clements, P.; Kazman, R.; Klein, M.: *Evaluating Software Architectures - Methods and Case Studies.* Addison-Wesley, 2002.

[CLM03] Chung, J.; Lin, K.; Mathieu, R. G.: *Web Services Computing: Advancing Software Interoperability. IEEE Computer*, 36(10):3, 2003.

[CM04] Charfi, A.; Mezini, M.: *Hybrid Web Service Composition: Business Processes Meet Business Rules.* In *2nd International Conference on Service Oriented Computing*, 2004.

[CN04] Cheesman, J.; Ntinolazos, G.: *SOA Reference Model - Part 2 - the Flexible Service Runtime. CBDi Journal*, 2004. http://www.cbdiforum.com/secure/interact/2004-06/Flexible_Service_Runtime.php (Accessed: 2006-08-08)

[Co04a] Colan, M.: *Service-Oriented Architecture expands the vision of Web services, Part 1. IBM developerWorks*, 2004. http://www-106.ibm.com/developerworks/webservices/library/ws-soaintro.html (Accessed: 2006-08-07)

[Co04b] Colan, M.: *SOA - So What? How SOA Expands the vision of Web Services*, 2004.

[Co05] Colombo, M. et al.: *Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[Cr05] Crawford, C. H. et al.: *Toward an On Demand Service-Oriented Architecture. IBM Systems Journal*, 44(1):28, 2005. http://www.research.ibm.com/journal/sj/441/crawford.html (Accessed: 2006-08-07)

[CSS06] Corporation, A.; Software, P.; Software, S.: *SOA Worst Practices, Volume I How to prevent your SOA from being DOA.* 2006.

[CT05] Capilla, R.; Topaloglu, N. Y.: *Product Lines for Supporting the Composition and Evolution of Service Oriented Applications.* In *International Workshop on Principles of Software Evolution (IWPSE'05)*, IEEE, 2005.

[Cu02] Curbera, F. et al.: *Unraveling the Web Services Web. IEEE Internet Computing*, 6(2):8, 2002.

[Cu03] Curbera, F. et al.: *The Next Step in Web Services. Communications of the ACM*, 46(10), 2003.

[Cu05a] Cuomo, G.: *IBM SOA "on the Edge".* In *SIGMOD 2005*, Baltimore, Maryland, USA, 2005.

[Cu05b] Curbera, F. et al.: *Toward a Programming Model for Service-Oriented Computing.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[CV05] Cibrán, M. A.; Verheecke, B.: *Dynamic business rules for web service composition.* In *Second Dynamic Aspects Workshop (DAW)*, 2005.

[CYN05] Chung, S.; Young, P.; Nelson, J.: *Service-Oriented Software Reengineering: Bertie3 as Web Services.* In *Proceedings of the 2005 IEEE International Conference on Web Services (ICWS'05)*, Orlando, FL, USA, 2005. IEEE Computer Society.

[Da04a] Davydov, M. M.: *Beyond SOA: Principles of Service Engineering. JAVAPro*, 2004. http://www.ftponline.com/javapro/2004_09/magazine/ features/mdavydov (Accessed: 2006-08-07)

[Da04b] Datz, T.: *What you need to know about service-oriented architecture. CIO Magazine*, 2004. http://www.cio.com/archive/011504/soa.html (Accessed: 2006-08-07)

[DD04] Dijkmann, R.; Dumas, M.: *Service-oriented Design: A Multi-viewpoint Approach. International Journal of Cooperative Information Systems*, 13(4):32, 2004.

[Di05a] Dirgahayu, T.: *Model-Driven Engineering of Web Service Compositions: A Transformation from ISDL to BPEL.* Dissertation, University of Twente, The Netherlands, 2005.

[Di05b] Dittberner: *Analysis of SOA & Integration Middleware Projects at Large U.S. Wireline Carriers*, 2005.

[DOT06] *The DOT Language*, 2006. http://www.graphviz.org/doc/info/lang.html (Accessed: 2006-09-15)

[Du05] Duan, Z. et al.: *SOA Without Web Services: a Pragmatic Implementation of SOA for Financial Transactions Systems.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[EDS06] EDS: *Beyond the Hype of Web Services-What Is It and How Can It Help Enterprises Become Agile*, 2006.

[EL04] Elfataray, A.; Layzell, P.: *Negotiating in the Service-Oriented Environment. Communications of the ACM*, 47(8), 2004.

[EMM06] Epstein, J.; Matsumoto, S.; McGraw, G.: *Software Security and SOA: Danger, Will Robinson! IEEE SECURITY & PRIVACY*, 4(1):4, 2006.

[En04] Endrei, M. et al.: *Patterns: Service-Oriented Architecture and Web Services*, 2004. http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246303.html?Open (Accessed: 2006-08-09)

[Er04] Erl, T.: *Service-oriented Architecture: A Field Guide to Integrating XML and Web Services.* Prentice Hall, 2004.

[Er05a] Erl, T.: *Service-Oriented Architecture - Concepts, Technology and Design*. Prentice Hall, 2005.

[Er05b] Erlanger, L.: *Making SOA work*. 2005.

[EWA06] Emig, C.; Weisser, J.; Abeck, S.: *Development of SOA-Based Software Systems - an Evolutionary Programming Approach*. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)*, IEEE, 2006.

[FB02] Fensel, D.; Bussler, C.: *The Web Service Modelling Framework WSMF. Electronic Commerce: Research and Applications*, 2002. http://www-128.ibm.com/developerworks/webservices/library/ws-securtrans (Accessed: 2006-08-08)

[Fe03] Ferguson, D. et al.: *Secure, Reliable, Transacted Web Services: Architecture and Composition*, 2003.

[Fe06] Fenn, J.: *Findings From the 'All Company' Research Meeting: What's the Value of Service-Oriented Architecture?*, 2006.

[FF03] Ferris, C.; Farrell, J.: *What Are Web Services? Communications of the ACM*, 46(6):1, 2003.

[FHH04] Fang, J.; Hu, S.; Han, Y.: *A Service Interoperability Assessment Model for Service Composition*. In *International Conference on Services Computing (SCC'04)*, 2004.

[FLB05] Filho, A. M. S.; Barros, R. S. M.; Liesenberg, H. K. E.: *Interaction Pattern Gathering in Service-oriented Applications*. In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, IEEE, 2005.

[Fo02] Foster, I. et al.: *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, 2002.

[Fo03] Foster, H. et al.: *Model-Based Verification of Web Service Compositions*. In *Proc. of the 18th International Conference on Automated Software Engineering (ASE'03)*, 2003.

[Fo04] Foster, H. et al.: *Compatibility Verification for Web Service Choreography*. In *ICWS*, 2004.

[Fo05] Foster, I.: *Service-oriented Science. Science*, 308, 2005.

[Fo06] Fox, S.: *Reigning In SOA. Infoworld*, 2006.

[FP02] Frankel, D.; Parodi, I.: *Using Model-Driven Architecture to Develop Web Services*, 2002.

[Fr04] Fricke, P.: *Oracle's Service-Oriented Architecture Strategy*, 2004.

[FS05a] Fatolahi, A.; Shams, F.: *Service Discovery Based on Business Processes: A Practical Approach Using UML.* IEEE, 2005.

[FS05b] Ferguson, D. F.; Stockton, M. L.: *Service-oriented architecture: Programming model and product architecture. IBM SYSTEMS JOURNAL*, 44(4):28, 2005.

[FT05a] Friedman, T.; Thompson, J.: *Management Update: Data Services: The Intersection of Data Integration and SOA*, 2005.

[FT05b] Foster, I.; Tuecke, S.: *The Different Faces of IT as Service. Queue*, 3(6):8, 2005.

[Fu05] Fung, C. K. et al.: *A Study of Service Composition with QoS Management.* In *International Conference on Web Services (ICWS'05)*, IEEE, 2005.

[FWK02] Fremantle, P.; Weerawarana, S.; Khalaf, R.: *Enterprise Services. Communications of the ACM*, 45:5, 2002.

[Ga02] : *Component Based Development Helps Enable SODA*, 2002.

[Ga06] Gao, T. et al.: *Toward QoS Analysis of Adaptive Service-Oriented Architecture.* In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, IEEE, 2006.

[Ga94] Gamma, E. et al.: *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, Upper Saddle River, 1994.

[GB04] Gold, N.; Bennett, K.: *Program Comprehension for Web Services.* In *12th IWPC.* IEEE Computer Society, 2004.

[Ge06] Genovese, Y.: *Most Benefits of Service-Oriented Architecture for Business Applications Are Longer Term*, 2006.

[Gi06] Gilbert, P.: *The Difference Between IT and Business Enablement*, 2006. http://www.ebizq.net/topics/bpm/features/7070.html (Accessed: 2006-06-22)

[GMK98] Gurbaxani, V.; Melville, N.; Kraemer, K.: *Disaggregating the ROI to IT capital*. Dissertation, University of California, 1998.

[Go02] Gottschalk, K. et al.: *Introduction to Web Services Architecture. IBM Systems Journal*, 41(2):8, 2002.

[Go03] McGovern, J.: *Java Web Services Architecture*. Morgan Kauffman, New York, 2003.

[Go04] Gold, N. et al.: *Understanding Service-Oriented Software. IEEE SOFTWARE*, 2004.

[Gr03] Grey, W. et al.: *An Analytic Approach for Quantifying the Value of e-Business Initiatives. IBM Systems Journal*, 42(3):14, 2003.

[Gr05] Grauer, M. et al.: *About a service-oriented software architecture for simulation and optimization in sheet metal forming*. In *International Conference on intelligent Processing and Manufacturing of Materials (on CD)*, Monterey, USA, 2005.

[Gr06a] Grauer, M. et al.: *The potential of service oriented computing for small and medium enterprises - analysis of value chains*. In *Multikonerenz Wirtschaftsinformatik 2006*, 2006.

[Gu05] GuiLing, W. et al.: *Service-Oriented Grid Architecture and Middleware Technologies for Collaborative E-Learning*. In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[Ha03a] Haas, H.: *Designing the architecture for Web services*, 2003. http://www.w3.org/2003/Talks/0521-hh-wsa (Accessed: 2006-08-08)

[Ha03b] Hashimi, S.: *Service-Oriented Architecture Explained*, 2003. http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html (Accessed: 2006-08-08)

[Ha05a] Hailstone, R.: *Service-Oriented Architecture for the Pragmatist*. 2005.

[Ha05b] Hasselmeyer, P.: *On Service Discovery Process Types*. In *Service Oriented Computing - ICSOC 2005*, 2005.

[Ha06] Havenstein, H.: *SQA Hurdles Forcing Changes In IT Units. Computerworld*, 40(17):3, 2006.

[HB02] Hagel, J.; Brown, J. S.: *Out of the Box: Strategies for Achieving Profits Today and Growth Tomorrow Through Web Services.* Harper Business School Press. Cambridge, MA, 2002.

[HB03] Hamadi, R.; Benatallah, B.: *A Petri-Net-Based Model for Web Service Composition.* In *Proc. of the Fourteenth Australasian database conference on Database technologies*, Adelaide, Australia, 2003.

[HB04] Hotle, M.; Blechar, M.: *Reuse Is the Key to SODA ROI Model. Gartner Research Note*, 2004.

[HB05] Halley, M. R.; Bashioum, C.: *Enterprise Transformation to a Service Oriented Architecture: Successful Patterns.* In *International Conference on Web Services (ICWS'05)*, IEEE, 2005.

[HCD06] Hussain, F. K.; Chang, E.; Dillon, T. S.: *Defining Reputation in Service Oriented Environment.* In *International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)*, IEEE, 2006.

[HCZ05] Huang, S.; Chen, H.; Zhang, L.-J.: *Progressive Auction Based Resource Allocation in Service-Oriented Architecture.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[He02] Helal, S.: *Standards for Service Discovery and Delivery. IEEE Pervasive Computing*, 1(3):6, 2002.

[He03] He, H.: *What Is Service-Oriented Architecture*, 2003. http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html (Accessed: 2006-06-22)

[He05a] Henrie, K. S.: *At your Service. CIO Insight*, 10:5, 2005.

[He05b] Hepp, M. et al.: *Semantic Business Process Management: Using Semantic Web Services for Business Process Management.* In *IEEE ICEBE 2005*, Beijing, China, 2005.

[He05c] Helbig, J.: *Wertbeitrag der IT.* In *Vorlesung strategisches IT Management, SS 05*, Technische Universität München, Chair Informatik XIX - Prof. F. Matthes, 2005.

[He06] Heiss, J. J.: *SOA Best Practices: A Conversation With Sun Microsystems Distinguished Engineer Mark Hapner*, 2006.

[HG05] Hepner, M.; Gamble, R.: *Establishing Connectors as Integration Services*. In *Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, IEEE, 2005.

[HLC05] Huang, Y.; Li, Y.; Chao, K.-M.: *A Framework for Service-Oriented Business Integration under Uncertainty*. In *IEEE International Conference on E-Business Engineering*, 2005.

[HS02] Herzum, P.; Sims, O.: *Business Component Factory: a comprehensive overview of business component development for the enterprise*. John Wiley & Sons, 2002.

[HS05] Huhns, M. N.; Singh, M. P.: *Service-Oriented Computing: Key Concepts and Principles. IEEE INTERNET COMPUTING*, JANUARY / FEBRUARY:7, 2005.

[Hu03] Huang, Y.: *A Template-based Approach for Service-Oriented e-business Solutions*. In *3rd International Conference on Electronic Commerce Engineering*, 2003.

[Hu05] Huang, Y. et al.: *A Stochastic Service Composition Model for Business Integration*. In *International Conference on Next Generation Web Services Practices (NWeSP'05)*, IEEE, 2005.

[HW03] Hündling, J.; Weske, M.: *Web services: Foundation and composition. EM - Electronic Markets*, 2003.

[IBM00] IBM: *Web Services Architecture Overview*, 2000-2003. http://www-128.ibm.com/developerworks/webservices/library/w-ovr/?dwzone=webservices (Accessed: 2006-08-07)

[IBM01] IBM: *Web Services Conceptual Architecture (WSCA 1.0)*, 2001.

[IBM02a] IBM: *Web Services Security (OASIS)*, 2002. http://www-106.ibm.com/developerworks/webservices/library/ws-secure (Accessed: 2006-08-07)

[IBM02b] IBM: *Web Services Transactions specifications (OASIS)*, 2002. http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/ (Accessed: 2006-08-07)

[IBM02c] IBM: *Security in a Web Services World: A Proposed Architecture and Roadmap*, 2002. http://www-106.ibm.com/developerworks/library/ws-secroad (Accessed: 2006-08-07)

[IBM02d] IBM: *Business Process Execution Language for Web Services Version 1.0*, 2002. http://www-106.ibm.com/developerworks/library/ws-bpel (Accessed: 2006-08-07)

[IBM03] IBM: *Business Process Execution Language for Web Services*, 2003. http://www.ibm.com/developerworks/library/ws-bpel (Accessed: 2006-08-07)

[IBM04a] IBM: *Service Oriented Architecture*, 2004.

[IBM04b] IBM: *SOA Reference Architecture: A Service-Based Foundation for Business Integration.* 2004.

[IBM04c] IBM: *New to SOA and Web Services*, 2004. http://www-128.ibm.com/developerworks/webservices/newto (Accessed: 2006-08-07)

[IBM04d] IBM: *IBM Service-Oriented Modeling and Architecture*, 2004.

[IBM04e] IBM: *Building SOA with Web services using Websphere Studio*, 2004. http://www-128.ibm.com/developerworks/edu/ws-dw-ws-soa-1-i.html?ca=dnt-536 (Accessed: 2006-08-09)

[IBM05a] IBM: *Five SOA projects that can pay for themselves in six months.*, 2005.

[IBM05b] IBM: *Providing what you need to get started with SOA.*, 2005.

[IBM06] IBM: *SOA and Web Services*, 2006. http://www.ibm.com/developerworks/webservices (Accessed: 2006-08-08)

[In05a] Informatica: *Data Integration in a Service-Oriented Architecture*, 2005.

[In05b] : *Service-Oriented Enterprise, The Technology Path to Business Transformation*, 2005.

[In06] Intargia: *Service-Architektur zielgerichtet nutzen*, 2006.

[It06] iteratec: *SOA - Service-orientierte Architekturen: Praxistest bestanden?*, 2006.

[JDT05] Jorstad, I.; Dustdar, S.; Thanh, D. v.: *A Service Oriented Architecture Framework for Collaborative Services.* In *International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*, 2005.

[Je05] Jennings, T.: *Developing Composite Applications - Comparing the Total Cost of Ownership*, 2005.

[Ji05] Jiang, Y. et al.: *On Procedure Strategy of Constructing SOA's Modeling Language.* In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, IEEE, 2005.

[JMP04] Juric, M.; Matthew, B.; Poornachandra, S.: *Business Process Execution Language for Web Services.* Packt Pub., Birmingham, UK, 2004.

[Jo04] Jonkers, H.: *Concepts for Modelling Enterprise Architectures. Int. Journal of Cooperative Information Systems*, 13(3):30, 2004.

[Jo05] Jones, S.: *Toward an Acceptable Definition of Service. IEEE SOFTWARE*, 2005.

[JRM05] Jaeger, M. C.; Rojec-Goldmann, G.; Muhl, G.: *QoS Aggregation in Web Service Compositions.* In *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '05)*, 2005.

[JS05] Jammes, F.; Smit, H.: *Service-Oriented Paradigms in Industrial Automation. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 1(1):9, 2005.

[JW05] Jiang, M.; Willey, A.: *Service-Oriented Architecture for Deploying and Integrating Enterprise Applications.* In *Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, IEEE, 2005.

[Ka03] Kaye, D.: *Loosely Coupled-The Missing Pieces of Web Services.* RDS Press, 2003.

[Ka05] Kano, M. et al.: *Analysis and simulation of business solutions in a serviceoriented architecture. IBM SYSTEMS JOURNAL*, 44(4):22, 2005.

[Ka06] Karacsony, K.: *Four Steps to SOA*. *Computerworld*, March:3, 2006.

[KBS05] Krafzig, D.; Banke, K.; Slama, D.: *Enterprise SOA*. Prentice Hall, 2005.

[KC05] Kim, T.-h.; Chang, C. K.: *Service-Oriented Design with Aspects (SODA)*. In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[Ke04] Keen, M. et al.: *Patterns: Implementing an SOA Using an Enterprise Service Bus*. IBM, first. Auflage, 2004.

[Ke05] Keen, M. et al.: *Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture*. IBM, first. Auflage, 2005.

[Ke06] Keen, M. et al.: *Patterns: Extended Enterprise SOA and Web Services*. IBM, first. Auflage, 2006.

[KEA05] Kumar, A.; El-Geniedy, A.; Agarwal, S.: *A Generalized Framework for Providing QoS Based Registry in Service Oriented Architecture*. In *International Conference on Services Computing (SCC'05)*, 2005.

[KJ05] Kircher, J.; Jain, P.: *Pattern-Oriented Software Architecture - Patterns for Resource Management*. John Wiley & Sons, 2005.

[KJE05] Karhunen, H.; Jantti, M.; Eerola, A.: *Service-Oriented Software Engineering (SOSE) Framework*. In *Services Systems and Services Management, 2005. Proceedings of ICSSSM '05. 2005 International Conference on*, IEEE, 2005.

[KKP05] Kulkarni, N. N.; Kumar, K. M. S.; Padmanabhuni, D. S.: *Reckoning Legislative Compliances with Service Oriented Architecture - A Proposed Approach*. In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[KKO03] Klein, M.; König-Ries, B.; Obreiter, P.: *Stepwise refinable service descriptions: Adapting DAML-S to staged service trading*. In *First International Conference on Service-Oriented Computing, ICSOC*. Springer, 2003.

[KKS06] Kalasapur, S.; Kumar, M.; Shirazi, B.: *Evaluating Service Oriented Architectures (SOA) in Pervasive Computing.* In *International Conference on Pervasive Computing and Communications (PERCOM'06)*, IEEE, 2006.

[KL03] Khalaf, R.; Leymann, F.: *On Web Services Aggregation.* In *International Workshops on Technologies for E-services in Conjunction with the VLDB conference*, 2003.

[KLS05a] Krogdahl, P.; Luef, G.; Steindl, C.: *Service-Oriented Agility: An initial analysis for the use of Agile methods for SOA development.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[KLS05b] Krogdahl, P.; Luef, G.; Steindl, C.: *Service-oriented agility: Methods for successful Service-Oriented Architecture (SOA) development, Part 1: Basics of SOA and agile methods*, 2005. http://www-128.ibm.com/developerworks/webservices/library/ws-agile1 (Accessed: 2006-06-22)

[KM04] Krüger, I. H.; Mathew, R.: *Systematic Development and Exploration of Service-Oriented Software Architectures.* In *Working IEEE/IFIP Conference on Software Architecture (WICSA'04)*, 2004.

[KMM06] Krüger, I. H.; Mathew, R.; Meisinger, M.: *Efficient Exploration of Service-Oriented Architectures using Aspects.* In *ICSE'06*, Shanghai, China, 2006.

[Ko05] Koyanagi, T. et al.: *Dynamic Policy Management on Business Performance Management Architecture.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[KP03] Kreger, H.; Philips, J.: *Toward Web Services Management Standards-An Architectural Approach to IT Systems Design. Web Services Journal*, 3(10):5, 2003.

[KPR04] Kazhamiakin, R.; Pistore, M.; Roveri, M.: *A Framework for Integrating Business Processes and Business Requirements.* In *Intl Enterprise Distributed Object Computing Conf (EDOC 2004)*, 2004.

[Kr95] Kruchten, P.: *Architectural Blueprints-The "4+1" View Model of Software Architecture. IEEE Software*, 12:15, 1995.

[Kr04a] Krüger, I. H. et al.: *Towards a Process and Tool-Chain for Service-Oriented Automotive Software Engineering.* In *Proceedings of the ICSE 2004 Workshop on Software Engineering for Automotive Systems (SEAS)*, 2004.

[Kr04b] Krüger, I. H.: *Service specification with MSCs and roles.* In *IASTED International Conference on Software Engineering*, 2004.

[KR05a] Knorr, E.; Rist, O.: *10 Steps to SOA. Infoworld*, 2005.

[Kr05b] Krcmar, H.: *Informationsmanagement.* Springer, Berlin Heidelberg, 4. Auflage, 2005.

[Kr06a] Krill, P.: *IBM'S New Governance Push. Infoworld*, 2006.

[Kr06b] Krill, P.: *SOA Governance Power Play. Infoworld*, 2006.

[Ku05] Kuhrmann, M. et al.: *Hot Spots der Software-Entwicklung: Service-orientierte Architekturen - Anspruch und Wirklichkeit*, 2005.

[Ku06] Kumnick, A.: *SOA: A Phoenix rises out of COBOL's Ashes. Canadian Underwriter*, 2006.

[KZ04] Kral, J.; Zemlicka, M.: *Architecture and modeling of service-oriented systems.* In *SOFSEM 2005 Communications*, Bratislava, Slovakia, 2004. Slovak Society for Computer Science.

[KZ05] Kral, J.; Zemlika, M.: *Implementation of Business Processes in Service-Oriented Systems.* In *International Conference on Services Computing (SCC'05)*. IEEE, 2005.

[La02] Landry, S.: *The Role of Web Services in the Financial Service Industry.* Report, Gartner Group, 2002.

[LA02] Levi, K.; Arsanjani, A.: *A Goal-driven approach to enterprise component identification and specification. Communications of the ACM*, 45(10), 2002.

[La03] Langdon, C.: *The State of Web services. Computer*, July:2, 2003.

[La06a] Lager, M.: *Making SOA pay right away*, 2006. http://www.destinationcrm.com/articles/default.asp?ArticleID=5903 (Accessed: 2006-06-22)

[La06b] Lamont, J.: *Service-oriented architecture: A way of Life. KM-World*, February:3, 2006.

[La06c] Lager, M.: *SOA Simple. CUSTOMER RELATIONSHIP MANAGEMENT*, February:6, 2006.

[Le00] Lee, K. et al.: *Feature-based Approach to Object-Oriented Engineering of Applications for Reuse. Journal of Software Practice and Experience*, 30:22, 2000.

[Le03] Leymann, F.: *Web Services - Distributed Applications without Limits, Business, Technology and Web.* Leipzig, 2003.

[Le04] Leymann, F.: *The Influence of Web Services on Software: Potentials and Tasks.* In *34th Annual Meeting of the German Computer Society*, Ulm, Germany, 2004. Springer.

[Le05a] Leymann, F.: *The (Service) Bus: Services Penetrate Everyday Life.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[Le05b] Lefever, B.: *Service-Oriented Solutions Approach (SOSA).* Report, Computer Associates International Inc., 2005.

[Le05c] Leymann, F.: *Combining Web Services and the Grid: Towards Adaptive Enterprise Applications.* In *CAiSE/ASMEA '05*, Porto, Portugal, 2005.

[Li03] Linthicum, D. S.: *Next Generation Application Integration.* Addison-Wesley, 2003.

[Li04] Litoiu, M.: *Migrating to Web Services: a Performance Engineering Approach. Journal of Software Maintenance and Evolution Research and Practice*, 16(1-2):20, 2004.

[Li05] Linthicum, D. S.: *The ROI of Your SOA*, 2005. http://www.ebizq.net/topics/soa/features/6092.html (Accessed: 2006-06-22)

[Li06] Liebhart, D.: *IT-Riesen kämpfen um die SOA-Krone. Computerwoche*, Sonderdruck:4, 2006.

[LMC05] Lee, Y.-C.; Ma, C.-M.; Chou, S.-C.: *A Service-Oriented Architecture for Design and Development of Middleware*. In *Asia-Pacific Software Engineering Conference (APSEC'05)*, IEEE, 2005.

[LMS05] Lewis, G.; Morris, E.; Smith, D.: *SMART: The Service-Oriented Migration and Reuse Technique*. In *CMU/SEI-05-TN-029*. Software Engineering Institute, 2005.

[LMS06] Lewis, G.; Morris, E.; Smith, D.: *Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture*. In *Conference on Software Maintenance and Reengineering (CSMR'06)*, IEEE, 2006.

[Lo06a] Loosely Coupled: *Managing SOA Performance*, 2006.

[LRS02] Leymann, F.; Roller, D.; Schmidt, M.: *Web services and business process management. IBM Systems Journal*, 41(2), 2002.

[LT03] Lublinsky, B.; Tyomkin, D.: *Dissecting Service-Oriented Architectures. Business Integration Journal*, 2003.

[Lu05a] Lu, X.: *An Investigation on Service-Oriented Architecture for Constructing Distributed Web GIS Application*. In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[Lu05b] Ludwig, H. et al.: *Template Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle*. In *Service Oriented Computing - ICSOC 2005*, 2005.

[Ma03] Marchese, M.: *Service oriented architectures for supporting environments in eGovernment applications*. In *Applications and the Internet Workshops*, IEEE, 2003.

[Ma04a] MacKenzie, M.: *Electronic Business Service Oriented Architecture: Service Broker - Proxy Pattern*, 2004.

[Ma04b] Mathew, R.: *Systematic Definition, Implementation and Evaluation of Service-Oriented Software*. Dissertation, University of California, 2004.

[Ma05a] Maximilien, E. M.: *Toward Web Services Interaction Styles*. In *International Conference on Services Computing (SCC'05)*, 2005.

[Ma05b] Ma, K. J.: *Web Services: What's Real and What's Not? IT Pro*, 2005.

[Ma05c] Machado, R. J. et al.: *Transformation of UML Models for Service-Oriented Software Architectures.* In *International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, IEEE, 2005.

[Ma05d] Manolescu, D. A.: *Orchestration Patterns in Service Oriented Architectures*, 2005. http://www.orchestrationpatterns.com (Accessed: 2006-08-07)

[Ma06a] Maréchaux, J.-L.: *Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus*, 2006.

[Ma06b] Malinverno, P.: *The ICC and SOA Governance*, 2006.

[Ma06c] Malinverno, P.: *Service-Oriented Architecture Craves Governance*, 2006.

[Ma06d] Margulius, D.: *Sizing up your Business Processes.* 2006.

[MBE03] Medjahed, B.; Bouguettaya, A.; Elmagarmid, A. K.: *Composing Web services on the Semantic Web. VLDB J.*, 12(4), 2003.

[Me04] Menascé, A.: *Composing Web Services. A QoS View. IEEE Internet Computing*, 8(6):3, 2004.

[MG03] *Practical Approaches to Service Oriented Architectures*, 2003. www.metagroup.com (Accessed: 2006-08-08)

[Mi05a] Mitra, T.: *Business-driven development*, 2005.

[Mi05b] Mitra, T.: *A case for SOA governance*, 2005.

[MKC04] Mukhi, N. K.; Konuru, R.; Curbera, F.: *Cooperative Middleware Specialization for Service Oriented Architectures.* In *WWW2004*, New York, New York, USA, 2004.

[MLD05] Marin, C.; Lalanda, P.; Donsez, D.: *A MDE Approach for Power Distribution Service Development.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[MM04] Milanovic, N.; Malek, M.: *Current Solutions for Web Service Composition. IEEE Internet Computing Magazine*, Nov-Dec, 2004.

[Mo02] Mohan, C.: *Dynamic e-business: Trends in web services.* In *third VLDB workshop on Technologies for E-Services.* Springer, 2002.

[Mo06] Moore, J.: *Web Services Governance. Baseline*, February:1, 2006.

[MPP02] Mecella, M.; Presicce, F. P.; Pernici, B.: *Modeling e-service orchestration through Petri nets.* In *3rd Intl. Workshop on Technologies for E-Services (TES).* Springer, 2002.

[MPT99] Marton, A.; Piccinelli, G.; Turfin, C.: *Service provision and composition in virtual business communities.* In *18th IEEE - ISRDS, Int. Workshop on Electronic Commerce*, Lausanne, Switzerland, 1999.

[MR03] Mes, A. d.; Rongen, E.: *Technical Note: Web service credentials. IBM Systems Journal*, 42:6, 2003.

[MSR05] Margaria, T.; Seffen, B.; Reitenspieß, M.: *Service-Oriented Design: The Roots.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[MVA05] Mayerl, C.; Vogel, T.; Abeck, S.: *SOA-based Integration of IT Service Management Applications.* In *International Conference on Web Services (ICWS'05)*, IEEE, 2005.

[Na02] Nakajima, S.: *Model-Checking Verification for Reliable Web Service.* In *OOPSLA 2002 Workshop on Objectoriented Web Services*, Seatle, Washington, 2002.

[Na03] Natis, Y. V.: *Service-Oriented Architecture Scenario*, 2003.

[Na05] Natis, Y. V.: *Applied SOA: Conquering IT Complexity Through Software Architecture*, 2005.

[Ne02] Newcomer, E.: *Understanding Web Services. XML, WSDL, SOAP, and UDDI.* Addison-Wesley, Boston, MA, 2002.

[Ne04] Newcomer, E.: *Understanding SOA with Web Services.* Addison Wesley Professional, 2004.

[Ne05] Nepal, S. et al.: *A Service-oriented Workflow Language for Robust Interacting Applications.* In *CoopIS*, 2005.

[NG05] Niblett, P.; Graham, S.: *Events and service-oriented architecture: The OASIS Web Services Notification specifications. IBM SYSTEMS JOURNAL*, 44(4):18, 2005.

[NM02] Narayanan, S.; McIlraith, S. A.: *Simulation, Verification and Automated Composition of Web Services. ACM*, 2002.

[No06] Nolle, T.: *Can There Be Too Much SOA? BUSINESS COMMUNICATIONS REVIEW*, April:3, 2006.

[NP03] Nelson, E.; Prasad, K.: *Automotive Infotronics: An emerging domain for Service-Based Architecture.* In *SBSE'03, Service-Based Software Engineering, Proceedings of the FM2003 Workshop, Technical Report TUM-I0315*, Technical University of Munich, 2003. I.H. Krüger, B. Schätz, M. Broy, and H. Hussman.

[NS03] Natis, Y. V.; Schulte, R. W.: *Introduction to Service-Oriented Architecture*, 2003.

[Gr04] O'Grady, S.: *SOA Meets Compliance: Compliance Oriented Architecture*, 2004.

[OA04] *Web Services Business Process Execution Language Version 2.0, Draft 1*, 2004. http://www.oasisopen.org/apps/org/workgroup/wsbpel (Accessed: 2006-08-09)

[OHE03] Oaks, P.; ter Hofstede, A.; Edmond, D.: *Capabilities: Describing What Services Do.* In *1st International Conference on Service-oriented Computing*, Trento, Italy, 2003. Springer-Verlag.

[OY05a] Orriens, B.; Yang, J.: *Bridging the Gap between Business and IT in Service Oriented Business Collaboration.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[OY05b] Orriens, B.; Yang, J.: *Establishing and Maintaining Compatibility in Service Oriented Business Collaboration.* In *7th International Conference on Electronic Commerce*, Xi'an, China, 2005.

[OYP05] Orriens, B.; Yang, J.; Papazoglou, M. P.: *A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[Pa02] Paolucci, M. et al.: *Semantic Matching of Web Services Capabilities*. In *1st International Semantic Web Conference*, Sardinia, Italy, 2002.

[Pa03] Papazoglou, M. P.: *Service-Oriented Computing: Concepts, Characteristics and Directions*. In *Conference on Web Information Systems Engineering*, Washington DC, USA, 2003. IEEE.

[Pa05a] Pasley, J.: *How BPEL and SOA Are Changing Web Services Development. IEEE INTERNET COMPUTING*, June:8, 2005.

[Pa05b] Patrick, P.: *Impact of SOA on Enterprise Information Architectures*. In *SIGMOD 2005*, Baltimore, Maryland, USA, 2005.

[Pa05c] Panda, D.: *An Introduction to Service-Oriented Architecture from a Java Perspective*, 2005. http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html (Accessed: 2006-08-08)

[Pa06] Palmer, M.: *Event Strem Processing and Event Driven Architectures*, 2006.

[PB03] Perks, C.; Beveridge, T.: *Guide to Enterprise IT Architecture*. Springer-Verlag, New York, 2003.

[Pe03a] Peltz, C.: *Web Services Orchestration and Choreography. IEEE Computer*, Oktober:6, 2003.

[Pe03b] Peltz, C.: *Web services orchestration: a review of emerging technologies, tools, and standards*, 2003.

[Pe04] Petinot, Y. et al.: *A service-oriented architecture for digital libraries*. In *International conference on Service oriented computing (ICSOC '04)*, ACM Press, 2004.

[PG03] Papazoglou, M. P.; Georgakopoulos, D.: *Service-Oriented Computing. Communications of the ACM*, 46(10):5, 2003.

[Pi04] Pistore, M. et al.: *Planning and monitoring web service composition*. In *Proc. 11th Int. Conf. on Artificial Intelligence: Methodology, Systems, Architectures*, 2004.

[PL03] Perrey, R.; Lycett, M.: *Service-Oriented Architecture*. In *Applications and the Internet Workshops*, IEEE, 2003.

[Pl05a] Plummer, D. C.: *Software Architectures Will Evolve From SOA and Events to Service Virtualization*, 2005.

[Pl05b] Plummer, D. C.: *Management Update: Six Missteps That Can Result in SOA Strategy Failure*, 2005.

[Pl05c] plenum Management Consulting: *Whitepaper IT-Architektur*, 2005.

[Po06] Pollet, T. et al.: *Telecom Services Delivery in a SOA*. In *Conference on Advanced Information Networking and Applications (AINA'06)*, IEEE, 2006.

[PS02] Piccinelli, G.; Stammers, E.: *From E-Processes to E-Networks: an E-Service-oriented Approach*. In *International Conference on Internet Computing*, 2002.

[PSZ01] Piccinelli, G.; Salle, M.; Zirpins, C.: *Service-oriented Modelling for e-Business Applications Components*. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. WET ICE 2001. Proceedings. Tenth IEEE International Workshops on*, IEEE, 2001.

[PW04] Parastatidis, S.; Webber, J.: *Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications*. In *SCC'04*, 2004.

[PY04] Papazoglou, M. P.; Yang, J.: *Design Methodology for Web Services and Business Process*. In *International Workshop on Technologies for E-Services*, 2004.

[QDD04] Quartel, D.; Dijkman, R.; Dumas, M.: *Methodological support for service-oriented design with ISDL*. In *International Conference on Service-Oriented Computing*, New York, USA, 2004. ACM Press.

[QDS05] Quartel, D.; Dijkman, R.; Sinderen, M. v.: *An approach to relate business and application services using ISDL*. In *International EDOC Enterprise Computing Conference (EDOC'05)*, IEEE, 2005.

[Qu05] Qureshi, K. A.: *Enterprises Application Integration*. In *International Conference on Emnerging Technologies*, 2005.

[Ra03] Ran, S.: *A Model for Web services Discovery with QoS. ACM SIGecom Exchange*, 4(1), 2003.

[RB05] Roser, S.; Bauer, B.: *A Categorization of Collaborative Business Process Modeling Techniques.* In *International Conference on E-Commerce Technology Workshops (CECW'05)*, IEEE, 2005.

[RD05a] Rosu, D.; Dan, A.: *Managing End-to-End Lifecycle of Global Service Policies.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[RD05b] Rosenberg, F.; Dustdar, S.: *Towards a Distributed Service-Oriented Business Rules System.* In *European Conference on Web Services (ECOWS'05)*, IEEE, 2005.

[RD05c] Rosenberg, F.; Dustdar, S.: *Business Rules Integration in BPEL - A Service-Oriented Apporach.* In *7th International IEEE Conference on E-Commerce Technology (CEC 2005)*, 2005.

[RD05d] Rosenberg, F.; Dustdar, S.: *Design and Implementation of a Service-Oriented Business Rules Broker.* In *1st IEEE International Workshop on Serviceoriented Solutions for Cooperative Organizations (SoS4CO'05)*, 2005.

[RD05e] Rosenberg, F.; Dustdar, S.: *Business Rules Integration in BPEL - A Service-Oriented Approach.* In *International Conference on E-Commerce Technology (CEC'05)*, IEEE, 2005.

[Ri05] Rittmann, S. et al.: *Integrating Service Specifications at Different Levels of Abstraction.* In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, 2005.

[Ro04] Robinson, R.: *Understand Enterprise Service Bus Scenarios and Solutions in Service-Oriented Architecture*, 2004. http://www-106.ibm.com/developerworks/webservices/library/ws-esbscen (Accessed: 2006-08-08)

[Ro05] Rother, T.: *Der optimale Weg zur Geschäftsagilität: SOA oder BPM?* 2005.

[SA05] SAP: *Enterprise Services Architecture: Das Konzept für den Wandel*, 2005.

[SB96] Succi, G.; Baruchelli, F.: *Analysing the Return of Investment of Reuse*. Dissertation, Uuiversita di Trento, 1996.

[Sc92] Scheer, A.: *Architecture for Integrated Information Systems - Foundations of Enterprise Modeling*. Springer-Verlag, Now York, Secaucus, USA, 1992.

[Sc02] Schmidt, R.: *Web services based execution of business rules*. In *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, 2002.

[Sc03] Schulte, R.: *Hype Cycle for Application Integration and Platform Middleware*. Report, 2003.

[Sc05] Schmidt, M.-T. et al.: *The Enterprise Service Bus: Making service-oriented architecture real. IBM SYSTEMS JOURNAL*, 44(4):17, 2005.

[Sc06] Schaad, A.: *Security in Enterprise Resource Planning Systems and Service-Oriented Architectures*. In *SACMAT'06*, Lake Tahoe, California, USA, 2006.

[SDS04] Stojanovic, Z.; Dahanayake, A.; Sol, H.: *Modeling and Design of Service-Oriented Architecture*. In *International Conference on Systems, Man and Cybernetics*. IEEE, 2004.

[SEH02] O'Sullivan, J.; Edmond, D.; ter Hofstede, A. H. M.: *What's in a Service? Towards accurate description of non-functional service properties. Distributed and Parallel Databases Journal*, 12(Special Issue on E-Services):26, 2002.

[SEH05a] O'Sullivan, J.; Edmond, D.; Hofstede, A. H. M. t.: *The Price of Services*. In *Service Oriented Computing - ICSOC 2005*, 2005.

[SEH05b] O'Sullivan, J.; Edmond, D.; Hofstede, A. H. M. t.: *Formal description of non-functional service properties*, 2005. http://www.bpm.fit.qut.edu.au/about/docs/non-functional.jsp (Accessed: 2006-08-10)

[SEI06] Software Engineering Institute: *What is CMMI?*, 2006. http://www.sei.cmu.edu/cmmi/general/general.html (Accessed: 2006-10-10)

[SG96] Shaw, M.; Garlan, D.: *Software Architecture, Perspectives on an Emerging Discipline*. Prentice Hall, 1996.

[Sh02] Sheth, A. et al.: *QoS for Service-oriented Middleware*. In *Conference on Systemics, Cybernetics and Informatics*, 2002.

[Sh03] Shaiva, V.: *Designing Adaptive Components for a Services Oriented Architecture*. IEEE, 2003.

[Sh04] Shan, T. C.: *Building a service-oriented eBanking Platform*. In *International Conference on Services Computing (SCC'04)*, IEEE, 2004.

[SH05] Singh, M. P.; Huhns, M. N.: *Service-oriented Computing - Semantics, Processes, Agents*. John Wiley & Sons, 2005.

[Si04] Sivashanmugam, K. et al.: *Framework for Semantic Web Process Composition, Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce. Special Issue of the International Journal of Electronic Commerce*, 2004.

[Si05] Sikka, V.: *Data and Metadata Management in Service-Oriented Architectures: Some Open Challenges*. In *SIGMOD 2005*, Baltimore, Maryland, USA, 2005.

[Sk06] Skyway Software: *SOA Starts With Design and Delivery*, 2006.

[SLC05] Sorathia, V.; Laliwala, Z.; Chaudhary, S.: *Towards Agricultural Marketing Reforms: Web Services Orchestration Approach*. In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[Sm04] Smith, D.: *Web Services Enable Service-Oriented and Event-Driven Architectures. Business Integration Journal*, May:3, 2004.

[SM05] Sassen, A. M.; Macmillan, C.: *The service engineering area: An overview of its current state and a vision of its future*, 2005.

[SM06] Stantchev, V.; Malek, M.: *Architectural translucency in service-oriented architectures. IEE Proc.-Softw.*, 153(1):7, 2006.

[SN03] Schulte, R. W.; Natis, Y. V.: *Event-Driven Architecture Complements SOA*, 2003.

[Sn06a] Sneed, H. M.: *Wrapping Legacy Software for Reuse in a SOA*. In *Multikonerenz Wirtschaftsinformatik 2006*, 2006.

[Sn06b] Sneed, H. M.: *Integrating legacy Software into a Service oriented Architecture*. In *Conference on Software Maintenance and Reengineering (CSMR'06)*, IEEE, 2006.

[Sk05] Skyway Software: *Service Oriented Architecture Realizing the Business Value*, 2005.

[So06a] Sonic Software: *A new Service-Oriented Architecture (SOA) maturity model*, 2006.

[So06b] SOAinitiative der Computerwoche: *Zentrale Ergebnisse der SOA-Studie 2006*. 2006.

[Sp02] Sprott, D.: *Service Oriented Process Matters*, 2002.

[Sp05] Specht, T. et al.: *Modeling Cooperative Business Processes and Transformation to a Service Oriented Architecture*. In *International Conference on E-Commerce Technology (CEC'05)*, IEEE, 2005.

[SS04] Stantchev, V.; Scherz, M.: *Challenges of service-oriented architectures*. In *IPSI '04*, Venice, Italy, 2004.

[St97] Steffen, B. et al.: *Hierarchical Service Definition*. In *Annual Review of Communic.*, Chicago, 1997. Int. Engineering Consortium.

[St05] Strnadl, C. F.: *Aligning Business and IT: The Process-Driven Architecture Model*. In *EUROCON 2005*, Serbia & Montenegro, Belgrade,, 2005. IEEE.

[St06] Stal, M.: *Using Architectural Patterns and Blueprints for Service-Oriented Architecture. IEEE SOFTWARE*, 2006.

[SUN02] SUN: *Using Web Services Effectively*, 2002. http://java.sun.com/blueprints/webservices/using/ webservbp.html (Accessed: 2006-08-09)

[SUN04] SUN: *Assessing Your SOA Readiness*, 2004.

[SUN06a] SUN: *SOA Governance Solution - Accelerating SOA*, 2006.

[SUN06b] SUN: *The SOA Platform Guide: Evaluate, Extend, Embrace*, 2006.

[SUN06c] SUN: *Sun SOA Opportunity Assessment - A Pragmatic Approach*, 2006.

[SUN06d] SUN: *Systematic Development and the Service Oriented Architecture*, 2006.

[SVA04] Schulte, R. W.; Valdes, R.; Andrews, W.: *SOA and Web Services Offer Little Vendor Independence*, 2004.

[SVW04] Svensson, E.; Vetter, C.; Werner, T.: *Data Consistency in a Heterogeneous IT Landscape: A Service Oriented Architecture Approach.* In *Intl Enterprise Distributed Object Computing Conf (EDOC 2004)*, IEEE, 2004.

[SW04] Schuschel, H.; Weske, M.: *Automated Planning in a Service-Oriented Architecture.* In *International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*, IEEE, 2004.

[Ta03] Tan, P. S.: *A Standards-based Web Service Framework. Synthesis Journal*, 2003.

[TAC05] Taylor, K.; Austin, T.; Cameron, M.: *Charging for Information Services in Service-Oriented Architectures.* In *International Conference on e-Technology, e-Commerce and e-Service (EEE-05).*, 2005.

[TBB03] Turner, M.; Budgen, D.; Brereton, P.: *Turning Software into a Service. IEEE Computer*, 36(10):7, 2003.

[TCP05] Tsai, W. T.; Chen, Y.; Paul, R.: *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* In *Tenth IEEE International Workshop on Object-oriented Realtime Dependable Systems (WORDS 05)*, Sedona, 2005.

[TG03] Tich, M.; Giese, G. H.: *Seamless UML Support for Service-Based Software Architectures.* In *Third International Workshop, FIDJI 2003*, 2003.

[Th05] Thompson, J. et al.: *Management Update: Predicts 2006: The Strategic Impact of SOA Broadens*, 2005.

[Ti04] Tilley, S. et al.: *On the business value and the technical challenges of adopting Web services. Journal of Software Maintenance and Evolution*, 16(1), 2004.

[Ti06a] TIBCO: *Business Process Management on an SOA Foundation*, 2006.

[Ti06b] TIBCO: *Service-Oriented Architecture and Event-Driven Architecture*, 2006.

[TJ05] Thanh, D. v.; Jorstad, I.: *A Service-Oriented Architecture Framework for Mobile Services.* In *Advanced Industrial Conference on Telecommunications*, IEEE, 2005.

[TJH05] Tao, X.; Jiang, C.; Han, Y.: *Applying SOA to Intelligent Transportation System.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[TP02] Tsalgatidou, A.; Pilioura, T.: *An Overview of Standards and Related Technology in Web Services. International Journal of Distributed and Parallel Databases*, 12(2):28, 2002.

[Tr04] Traverso, P.: *Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development.* In *2d International Conference on Service Oriented Computing*, New York, USA, 2004.

[Ts04] Tsai, W. T. et al.: *A Software Reliability Model for Web Services.* In *8th IASTED International Conference on Software Engineering and Applications*, Cambridge, MA, 2004.

[Ts05] Tsai, W. T.: *Service-Oriented System Engineering: A New Paradigm.* In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, IEEE, 2005.

[Tu04] Tucker, R.: *Enterprise Architecture Technology Assessment.* Report, 2004. http://www.mitre.org (Accessed: 2006-08-07)

[TYL05] Tan, P. S.; Yin, Z. L.; Lee, S. G.: *Realising Service-Oriented Architecture through Context-Based Dynamic Service Invocation.* In *International Conference on Industrial Informatics (INDIN)*, IEEE, 2005.

[Vi02] Vinoski, S.: *Web Services Interaction Models. Current practice. Internet Computing*, 6(3):3, 2002.

[Vi06] Vinoski, S.: *The Social Side of Services. IEEE INTERNET COMPUTING*, 2006.

[VM06] Vasilescu, E.; Mun, S. K.: *Service Oriented Architecture (SOA) Implications for Large Scale Distributed Health Care Enterprises.* In *Distributed Diagnosis and Home Healthcare (D2H2) Conference*, Arlington, Virginia, USA, 2006.

[Vo03] Vogels, W.: *Web services are not distributed objects. IEEE Internet Computing*, 7(6):8, 2003.

[W3C02a] W3C: *W3 Consortium Web Services Initiative*, 2002. http://www.w3.org/2002/ws (Accessed: 2006-08-07)

[W3C02b] W3C: *W3C Web Services Description Working Group*, 2002. http://www.w3.org/2002/ws/desc (Accessed: 2006-08-07)

[W3C02c] W3C: *Web Services Description Language (WSDL)*, 2002. www.w3.org/TR/2002/WDwsdl12-20020709 (Accessed: 2006-08-07)

[W3C04a] W3C Working Group: *Web Services Architecture*, 2002-2004. http://www.w3.org/TR/ws-arch (Accessed: 2006-08-08)

[W3C04b] : *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*, 2004. http://www.w3.org/TR/2004/WD-wsdl20-primer-20041221 (Accessed: 2006-08-09)

[W3C06c] W3C: *Web Services Architecture*, 2006. http://www.w3.org/2002/ws/arch/2/08/wd-wsa-arch-20020821.html (Accessed: 2006-08-08)

[Wa04] Wang, G. et al.: *Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architecture.* In *Proceedings of the 8th IEEE Enterprise Distributed Object Computing Conference*, 2004.

[Wa05] Wang, C. et al.: *A Policy-Based Approach for QoS Specification and Enforcement in Distributed Service-Oriented Architecture.* In *International Conference on Services Computing (SCC'05)*, IEEE, 2005.

[WEI06] Wong-Bushby, I.; Egan, R.; Isaacson, C.: *A Case Study in SOA and Re-Architecture at Company ABC.* In *Hawaii International Conference on System Sciences*, Hawaii, 2006. IEEE.

[WC03] Webber, J.; Chatterjee, S.: *Developing Enterprise Web Services: An Architect's Guide.* Prentice Hall PTR, 2003.

[WDL04] Wu, Z.; Deng, S.; Li, Y.: *Introducing EAI and Service Components into Process Management.* In *International Conference on Services Computing (SCC'04)*, IEEE, 2004.

[We04] Weill, P.: *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results.* Harvard Business School Press. Cambridge, MA, 2004.

[We05] Weerawarana, S. et al.: *Web Services Platform Architecture.* Prentice Hall, 2005.

[Wi06] Windley, P. J.: *SOA Governance: Rules of the Game.* 2006.

[WK05a] Wang, X.; Kang, J.: *Service-Oriented Business Integration and Management in Telecom.* IEEE, 2005.

[WK05b] Weyerhäuser, M.; Krischer, O.: *SOA: EAI in neuen Schläuchen?*, 2005.

[WR04] Weill, P.; Ross, J. W.: *IT Governance.* Harvard Business School Press, Boston, Massachusetts, 2004.

[WT04] Wonohoesodo, R.; Tari, Z.: *A Role based Access Control for Web Services.* In *IEEE International Conference on Services Computing (SCC'04)*, 2004.

[WYH05] Wang, J.; Yu, J.; Han, Y.: *A Service Modeling Approach with Business-Level Reusability and Extensibility.* In *International Workshop on Service-Oriented System Engineering (SOSE'05)*, 2005.

[Ya03] Yang, J.: *Web Service Componentization: Towards Service Reuse and Specialization. Communications of the ACM*, 46(10):6, 2003.

[YIM05a] Yanchuk, A.; Ivanyukovich, A.; Marchese, M.: *A Lightweight Formal Framework for Service-Oriented Applications Design.* In *Service Oriented Computing - ICSOC 2005*, 2005.

[YIM05b] Yanchuk, A.; Ivanyukovich, A.; Marchese, M.: *Towards a mathematical foundation for service-oriented applications design. Technical report dit-05-059*, 2005.

[YK97] Youngdahl, W. E.; Kellogg, D. L.: *Relationship between Service Customers' Quality Assurance Behaviors, Satisfaction, and Effort: A Cost of Quality Perspective. Journal of Operations Management*, 15(1):14, 1997.

[Yu04] Yu, J. et al.: *Developing End-User Programmable Service-Oriented Applications with VINCA.* In *2nd Ljungby Workshop on Information Logistics*, Ljungby, Sweden, 2004.

[Za87] Zachman, J. A.: *A Framework for Information Systems Architecture. IBM Systems Journal*, 26(3), 1987.

[Za96] Zachman, J. A.: *Concepts of the Framework for Enterprise Architecture*, 1996.

[Za03] Zapthink: *Service-Oriented Consulting: Facilitating the Service-Oriented Enterprise.* Report, 2003. http://www.zapthink.com/report.html?id=ZTR-WS109 (Accessed: 2006-08-09)

[ZCC04] Zhang, J.; Chung, J.-Y.; Chang, C. K.: *Migration to web services oriented architecture: a case study.* In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing.* ACM Press, 2004.

[Ze03a] Zeng, L. et al.: *Quality Driven Web Services Definition*, 2003.

[Ze03b] Zeng, L. et al.: *Flexible Composition of Enterprise Web Services. Electronic Markets - The International Journal of Electronic Commerce and Business Media*, 13(2), 2003.

[Ze04] Zeng, L. et al.: *QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering*, 30(5):17, 2004.

[Zi04] Zimmermann, O. et al.: *Second generation web services-oriented architecture in production in the finance industry.* In *OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, ACM Press, 2004.

[Zi05] Zimmermann, O. et al.: *Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario: Rationale, Concepts, Lessons Learned.* In *OOPSLA*, San Diego, California, USA, 2005. ACM.

[ZKG04] Zimmermann, O.; Krogdahl, P.; Gee, C.: *Elements of Service-Oriented Analysis and Design*, 2004. http://www-128.ibm.com/developerworks/webservices/library/ws-soad1 (Accessed: 2006-08-07)

[ZTP03] Zimmermann, O.; Tomlinson, M.; Peuser, S.: *Perspectives on Web Services - Applying SOAP, WSDL and UDDI to Real-World Projects.* Springer-Verlag, Heidelberg, 2003.

[Zy02] Zyl, J. v.: *Application Assembly using Web services.* In *Annual International Computer Software and Applications Conference (COMPASC'02)*, IEEE Press, 2002.

[ZY04a] Zhang, Z.; Yang, H.: *Incubating Services in Legacy Systems for Architectural Migration.* In *Asia-Pacific Software Engineering Conference (APSEC'04)*, 2004.

[ZY04b] Zhang, Z.; Yang, H.: *One-Stone-Two-Birds: Legacy System Reengineering and Web Services Development - A Component-Based and Service-Oriented Approach.* In *Postgraduate Research Conference in Electronics, Photonics, Communications & Networks, and Computing Science (PREP 2004)*, Hatfield, UK, 2004.

Appendix A

Results of analysis

Label	Title	Year	Type	Ref-In	Ref-Out
AAS04	How Service-Oriented Architecture Will Affect SMBs	2004	Primary	0	0
AB05	Approaches to Identify and Develop Web Services as Instance of SOA Architecture	2005	Primary	0	8
Ac05	SOA in the Real World - Experiences	2005	Primary	0	1
AH06	Achieving Agility: SOA Will Build Organizational Agility, but Watch the Hype	2006	Primary	0	0
AHZ04	Cross-Enterprise Business Process Management Architecture - Methods and Tools for Flexible Collaboration	2004	Secondary	1	0
AKL99	A Big-Picture Look at Enterprise Architectures	1999	Secondary	1	0
Al04	Web Services - Concepts, Architectures and Applications	2004	Secondary	10	0
Al97	A Formal Approach to Software Architecture	1997	Secondary	1	0
An03	Business Process Execution Language for Web Services Version 1.1	2003	Secondary	15	0
Ap03	Service- and Component-Based Development: Using the Select Perspective and UML	2003	Secondary	1	0
APS06	SOA Worst Practices, Volume I How to prevent your SOA from being DOA	2006	Primary	0	0
Ar03	Web Services: Promises and Compromises	2003	Secondary	1	0
Ar04	Service-oriented modeling and architecture - How to identify, specify, and realize services for your SOA	2004	Primary	4	4
Ar05	OASIS Web Service Business Process Execution Language V2.0 Working Draft	2005	Secondary	2	0
Ba02	Deployment of Service Oriented Architecture for a Business Community	2002	Primary	0	3
Ba03a	Modeling and Validation of Service-Oriented Architectures: Application vs. Style	2003	Primary	0	3
Ba03b	Web services and Service-Oriented Architecture	2003	Secondary	2	0
Ba05	A Business Model for Deploying Web Services: A Data-Centric Approach Based on Factual Dependencies	2005	Secondary	0	0
Ba06	Model offers measure for SOA success	2006	Primary	0	0
BCK03	Software Architecture in Practice	2003	Secondary	1	0
Be01	An Architectural Model for Service-Based Software with Ultra Rapid Evolution	2001	Secondary	1	0
Be02	Prototype Implementations of an Architectural Model for Service-Based Flexible Software	2002	Secondary	1	0
Be03	Conceptual modeling of web service conversations	2003	Secondary	1	0
BGA01	A Case for Economy Grid Architecture for Service-Oriented Grid Computing	2001	Secondary	1	0
BH05	Creating Value with Interactive Pricing Mechanisms - a Web Service-Oriented Architecture	2005	Primary	0	3
Bi04	Like it or not, web services are distributed objects	2004	Secondary	1	0
Bi05a	Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals	2005	Primary	1	7
Bi05b	Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap	2005	Secondary	2	0
BK04	Services and service oriented software architectures - methodological foundations	2004	Secondary	1	0
Bl02	Web Services Management: The Key to Service-Oriented Architectures	2002	Secondary	1	0
BL06	Service-Oriented Computing	2006	Primary	0	0
BLM05	Integrating service-oriented technologies to support business processes	2005	Primary	0	3
BM05	Learn the Key Success Factors for SOA Deployments	2005	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
AAS04						1	3	3			
AB05							8	13			
Ac05						3			2		
AH06						3	3	2			
AHZ04						24					
AKL99											
Al04											
Al97				74							
An03											
Ap03											
APS06	2	5			2		15	10			
Ar03							4	7			
Ar04		5						1	2		
Ar05											
Ba02				4			7	9			
Ba03a				2			1				
Ba03b											
Ba05											
Ba06		1				3	1		2		
BCK03											
Be01								1			
Be02			1								
Be03							10	9			
BGA01				1							
BH05											
Bi04											
Bi05a	3	16	4		3				7		2
Bi05b											
BK04											
Bl02											
BL06				1		1	1	7	2		
BLM05							14	5			
BM05	1	5			9	4					

Label	Title	Year	Type	Ref-In	Ref-Out
BNZ06	Checking Conformance between Business Processes and Web Service Contract in Service Oriented Applications	2006	Primary	0	6
Bo01	ODSI: Enterprise Service Co-ordination	2001	Secondary	1	0
Bo04a	Web Services Architecture, W3C Working Group Note	2004	Secondary	3	0
Bo04b	Web Services Architecture	2004	Secondary	1	0
Br02	Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications	2002	Secondary	2	0
BR03	Web Services Architectures and Best Practices	2003	Secondary	0	0
Br03	Service-Oriented Systems Engineering: Specification and Design of Services and Layered Architectures	2003	Primary	0	0
BRV04	Adding High Availability and Autonomic Behavior to Web Services	2004	Secondary	2	0
BSD03	The Self-Serv environment for web services composition	2003	Secondary	1	0
BSM05	Benefits and Challenges of SOA in Business Terms	2005	Primary	0	0
Bu02	The Tao of e-business services	2000	Secondary	1	0
Bu05	Service-Oriented Architecture - Developing the Enterprise Roadmap	2005	Secondary	1	0
Bu96	Pattern-Oriented Software Architecture - A System of Patterns	1996	Secondary	1	0
Ca03	Business-Oriented Management of Web Services	2003	Primary	2	3
Ca04	Software Evolution in the Era of Software Services	2004	Secondary	1	0
Ca05a	Service Oriented Database Architecture: App Server-Lite?	2005	Primary	0	0
Ca05b	Service-Oriented Enterprise: How To Make Your Business Fast, Flexible and Responsive	2005	Primary	0	0
Ca06	Service-Oriented Architecture	2006	Primary	0	0
CBD05	Service-Oriented Development and Integration: Toward Web Services-Based Business Information Systems	2005	Primary	0	2
CC06	CIO and Computerworld Research: The Forecast for SOA	2006	Primary	0	0
CDH05	Trust and Reputation Relationships in Service-Oriented Environments	2005	Primary	0	0
Ce02	Web Services Essentials	2002	Secondary	2	0
CFR03a	An Enhanced Service Oriented Architecture for Developing Web-based Applications	2003	Secondary	2	0
CFR03b	Improving Dependability of Service Oriented Architectures for Pervasive Computing	2003	Secondary	0	0
Ch01	Web Services Description Language (WSDL)	2001	Secondary	14	0
Ch02a	Web Services Architecture, W3C Working Draft	2002	Secondary	1	0
Ch02b	A reactive service composition architecture for pervasive computing environments	2002	Secondary	1	0
Ch03a	A Service-Oriented Framework for Collaborative Product Commerce	2003	Primary	0	4
Ch03b	Web Services Description Language (WSDL) Version 1.2	2003	Secondary	2	0
Ch04a	Enterprise Service Bus	2004	Secondary	8	0
Ch04b	Web Services Description Language (WSDL) Version 2.0	2004	Secondary	1	0
Ch05a	Feature Analysis for Service-Oriented Reengineering	2005	Primary	0	6
Ch05b	Impact of service orientation at the business level	2005	Primary	0	6
Ch05c	Think business processes for SOA success	2005	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
BNZ06							3	2			
Bo01						1	3				
Bo04a											
Bo04b											
Br02							11	6			1
BR03							34	22			
Br03											
BRV04							1	1			
BSD03							8	4			4
BSM05		2			2						
Bu02	2						2	1			
Bu05											
Bu96											
Ca03							9	2			
Ca04				1		1	1	5			
Ca05a							6	3			1
Ca05b	4	1	1	4					1		
Ca06											
CBD05											
CC06	1				1				1		
CDH05	1	1									
Ce02											
CFR03a				2			22	9			
CFR03b				10							
Ch01											
Ch02a											
Ch02b											
Ch03a								4			
Ch03b											
Ch04a											
Ch04b											
Ch05a				3	1		2	3			
Ch05b	6	1	1	3		1			1		
Ch05c											

Label	Title	Year	Type	Ref-In	Ref-Out
CHT04	Migrating to a service-oriented architecture	2004	Primary	1	0
Ci06	Building an infrastructure to enable SOA	2006	Primary	0	0
CK01	Quality of Service in Web Services	2001	Secondary	1	0
CK05	Management of the serviceoriented-architecture life cycle	2005	Primary	1	4
CKB04	An introduction to the Web Services architecture and its specifications	2004	Secondary	1	0
CKK02	Evaluating Software Architectures - Methods and Case Studies	2002	Secondary	1	0
CLM03	Web Services Computing: Advancing Software Interoperability	2003	Secondary	2	0
CM04	Hybrid Web Service Composition: Business Processes Meet Business Rules	2004	Secondary	2	0
CN04	SOA Reference Model - Part 2 - the Flexible Service Runtime	2004	Secondary	1	0
Co04a	Service-Oriented Architecture expands the vision of Web services, Part 1	2004	Secondary	3	0
Co04b	SOA - So What? How SOA Expands the vision of Web Services	2004	Primary	0	0
Co05	Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems	2005	Primary	0	5
Cr05	Toward an On Demand Service-Oriented Architecture	2005	Secondary	1	0
CT05	Product Lines for Supporting the Composition and Evolution of Service Oriented Applications	2005	Primary	0	6
Cu02	Unraveling the Web Services Web	2002	Secondary	2	0
Cu03	The Next Step in Web Services	2003	Secondary	4	0
Cu05a	IBM SOA "on the Edge"	2005	Primary	0	0
Cu05b	Toward a Programming Model for Service-Oriented Computing	2005	Primary	0	4
CV05	Dynamic business rules for web service composition	2005	Secondary	1	0
CYN05	Service-Oriented Software Reengineering: Bertie3 as Web Services	2005	Secondary	1	0
Da04a	Beyond SOA: Principles of Service Engineering	2004	Secondary	1	0
Da04b	What you need to know about service-oriented architecture	2004	Secondary	1	0
Di04	Service-oriented Design: A Multi-viewpoint Approach	2004	Secondary	3	0
Di05a	Analysis of SOA & Integration Middleware Projects at Large U.S. Wireline Carriers	2005	Secondary	1	0
Di05b	Model-Driven Engineering of Web Service Compositions: A Transformation from ISDL to BPEL	2005	Secondary	1	0
Du05	SOA Without Web Services: a Pragmatic Implementation of SOA for Financial Transactions Systems	2005	Primary	0	10
EDS06	Beyond the Hype of Web Services-What Is It and How Can It Help Enterprises Become Agile	2006	Secondary	1	0
EL04	Negotiating in the Service-Oriented Environment	2004	Secondary	1	0
EMM06	Software Security and SOA: Danger, Will Robinson!	2006	Primary	0	0
En04	Patterns: Service-Oriented Architecture and Web Services	2004	Secondary	6	0
Er04	Service-oriented Architecture: A Field Guide to Integrating XML and Web Services	2004	Secondary	3	0
Er05a	Service-Oriented Architecture - Concepts, Technology and Design	2005	Primary	0	0
Er05b	Making SOA work	2005	Primary	0	0
EWA06	Development of SOA-Based Software Systems - an Evolutionary Programming Approach	2006	Primary	0	3
FB02	The Web Service Modelling Framework WSMF	2002	Secondary	3	0
Fe03	Secure, Reliable, Transacted Web Services: Architecture and Composition	2003	Secondary	1	0
Fe06	Findings From the 'All Company' Research Meeting: What's the Value of Service-Oriented	2006	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
CHT04	1				3		1	1			
Ci06	1	2	1				1	1			
CK01											
CK05						1		3	1		2
CKB04							96	27			
CKK02											
CLM03							3	2			
CM04						1					
CN04											
Co04a							3	4			
Co04b							14	14	6		
Co05											
Cr05		4		1	2		3	11			
CT05											
Cu02							116	33			
Cu03						2	5	14			
Cu05a	1						3	2			
Cu05b							6	15	5		
CV05											
CYN05				1							
Da04a						3	4	2			
Da04b	2	1	1		4	1	2	2	2		
Di04							2	4			
Di05a											
Di05b				1			4	190			
Du05				1			19	6			4
EDS06					1		1				
EL04							1	3			1
EMM06			1				1	1			
En04	3		1		1	3	309	286	39		6
Er04											
Er05a											
Er05b		2			1		1		1		
EWA06						3	5	8	1		
FB02			3			3	22	16			
Fe03							13	34			
Fe06	4										

Label	Title	Year	Type	Ref-In	Ref-Out
FF03	What Are Web Services?	2003	Secondary	1	0
FHH04	A Service Interoperability Assessment Model for Service Composition	2004	Primary	0	2
FLB05	Interaction Pattern Gathering in Service-oriented Applications	2005	Primary	0	2
Fo02	The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration	2002	Secondary	1	0
Fo03	Model-Based Verification of Web Service Compositions	2003	Secondary	2	0
Fo04	Compatibility Verification for Web Service Choreography	2004	Secondary	1	0
Fo05	Service-oriented Science	2005	Secondary	1	0
Fo06	Reigning In SOA	2006	Primary	0	0
FP02	Using Model-Driven Architecture to Develop Web Services	2002	Secondary	1	0
Fr04	Oracle's Service-Oriented Architecture Strategy	2004	Secondary	1	0
FS05a	Service Discovery Based on Business Processes: A Practical Approach Using UML	2005	Primary	0	1
FS05b	Service-oriented architecture: Programming model and product architecture	2005	Primary	2	6
FT05a	Management Update: Data Services: The Intersection of Data Integration and SOA	2005	Primary	0	0
FT05b	The Different Faces of IT as Service	2005	Secondary	1	0
Fu05	A Study of Service Composition with QoS Management	2005	Primary	0	9
FWK02	Enterprise Services	2002	Secondary	1	0
Ga02	Component Based Development Helps Enable SODA	2002	Secondary	1	0
Ga06	Toward QoS Analysis of Adaptive Service-Oriented Architecture	2006	Primary	0	2
GB04	Program Comprehension for Web Services	2004	Secondary	2	0
Ge06	Most Benefits of Service-Oriented Architecture for Business Applications Are Longer Term	2006	Primary	0	0
Gi06	The Difference Between IT and Business Enablement	2006	Primary	0	0
GMK98	Disaggregating the ROI to IT capital	1998	Primary	0	0
Go02	Introduction to Web Services Architecture	2002	Secondary	1	0
Go03	Java Web Services Architecture	2003	Secondary	0	0
Go04	Understanding Service-Oriented Software	2004	Primary	0	4
Gr03	An Analytic Approach for Quantifying the Value of e-Business Initiatives	2003	Secondary	1	0
Gr04	SOA Meets Compliance: Compliance Oriented Architecture	2004	Secondary	1	0
Gr05	About a service-oriented software architecture for simulation and optimization in sheet metal forming	2005	Secondary	1	0
Gr06	The potential of service oriented computing for small and medium enterprises - analysis of value chains	2006	Primary	0	5
Gu05	Service-Oriented Grid Architecture and Middleware Technologies for Collaborative E-Learning	2005	Primary	0	0
Ha03a	Designing the architecture for Web services	2003	Secondary	1	0
Ha03b	Service-Oriented Architecture Explained	2003	Secondary	1	0
Ha05a	Service-Oriented Architecture for the Pragmatist	2005	Primary	0	0
Ha05b	On Service Discovery Process Types	2005	Primary	0	1
Ha06	SQA Hurdles Forcing Changes In IT Units	2006	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
FF03							1	8			
FHH04							1	1			
FLB05									1		
Fo02			1				14	28			
Fo03								2			
Fo04											
Fo05											
Fo06		7									
FP02							12	23			
Fr04	1					3	1	1			
FS05a				2	8						
FS05b	3					2	22	32	3		
FT05a		3									
FT05b							2	2			
Fu05						1	92	2			1
FWK02						5	32	24			
Ga02					2						
Ga06				5			4	6			
GB04				1			3	10			
Ge06					1	3					
Gi06	1		1			30					
GMK98					4						
Go02							8	16			
Go03											
Go04					1		1				
Gr03	13			5	2						
Gr04		2									
Gr05											
Gr06											
Gu05				1			1	2			2
Ha03a											
Ha03b							2	2			
Ha05a		2			3				1		
Ha05b											
Ha06		1			1				1		

Label	Title	Year	Type	Ref-In	Ref-Out
HB02	Out of the Box: Strategies for Achieving Profits Today and Growth Tomorrow Through Web Services	2002	Secondary	1	0
HB03	A Petri-Net-Based Model for Web Service Composition	2003	Secondary	1	0
HB04	Reuse Is the Key to SODA ROI Model	2004	Secondary	1	0
HB05	Enterprise Transformation to a Service Oriented Architecture: Successful Patterns	2005	Primary	0	1
HCD06	Defining Reputation in Service Oriented Environment	2006	Primary	0	0
HCZ05	Progressive Auction Based Resource Allocation in Service-Oriented Architecture	2005	Primary	0	2
He02	Standards for Service Discovery and Delivery	2002	Secondary	1	0
He03	What Is Service-Oriented Architecture	2003	Primary	0	0
He05a	At your Service	2005	Primary	0	0
He05b	Semantic Business Process Management: Using Semantic Web Services for Business Process Management	2005	Secondary	1	0
He06	SOA Best Practices: A Conversation With Sun Microsystems Distinguished Engineer Mark Hapner	2006	Primary	0	0
HG05	Establishing Connectors as Integration Services	2005	Primary	0	0
HLC05	A Framework for Service-Oriented Business Integration under Uncertainty	2005	Secondary	1	0
HS02	Business Component Factory: a comprehensive overview of business component development for the enterprise	2002	Secondary	1	0
HS05	Service-Oriented Computing: Key Concepts and Principles	2005	Primary	1	3
Hu03	A Template-based Approach for Service-Oriented e-business Solutions	2003	Secondary	1	0
Hu05	A Stochastic Service Composition Model for Business Integration	2005	Primary	0	6
HW03	Web services: Foundation and composition	2003	Secondary	1	0
IBM00	Web Services Architecture Overview	2000-2	Secondary	3	0
IBM01	Web Services Conceptual Architecture (WSCA 1.0)	2001	Secondary	1	0
IBM02a	Web Services Security (OASIS)	2002	Secondary	2	0
IBM02b	Web Services Transactions specifications (OASIS)	2002	Secondary	1	0
IBM02c	Security in a Web Services World: A Proposed Architecture and Roadmap	2002	Secondary	1	0
IBM02d	Business Process Execution Language for Web Services Version 1.0	2002	Secondary	5	0
IBM03	Business Process Execution Language for Web Services	2003	Secondary	11	0
IBM04a	Service Oriented Architecture	2004	Primary	0	0
IBM04b	SOA Reference Architecture: A Service-Based Foundation for Business Integration	2004	Primary	0	4
IBM04c	New to SOA and Web Services	2004	Secondary	1	0
IBM04d	IBM Service-Oriented Modeling and Architecture	2004	Secondary	1	0
IBM04e	Building SOA with Web services using Websphere Studio		Secondary	1	0
IBM05a	Five SOA projects that can pay for themselves in six months.	2005	Primary	0	0
IBM05b	Providing what you need to get started with SOA.	2005	Primary	0	0
IBM06	SOA and Web Services	2006	Secondary	1	0
In05a	Data Integration in a Service-Oriented Architecture	2005	Primary	0	0
In05b	Service-Oriented Enterprise, The Technology Path to Business Transformation	2005	Secondary	1	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
HB02											
HB03							5	4			
HB04					6						
HB05		8									
HCD06											
HCZ05											
He02							5				
He03							25	3			
He05a											
He05b						40					
He06	1	1					1	1			
HG05							2	2			
HLC05								1			
HS02											
HS05							2	1			
Hu03											
Hu05								1			
HW03	2					7	57	15			
IBM00							1	1			
IBM01											
IBM02a											
IBM02b											
IBM02c							21	2			
IBM02d											
IBM03											
IBM04a									1		
IBM04b		3				7	5	4	5		
IBM04c		11	1				1		3		
IBM04d	1										
IBM04e								1			
IBM05a	2	1			1		24	2	1		
IBM05b	1	5									
IBM06											
In05a		2	1		4	3	3	3	2		
In05b							1				

Label	Title	Year	Type	Ref-In	Ref-Out
In06	Service-Architektur zielgerichtet nutzen	2006	Primary	0	0
It06	SOA - Service-orientierte Architekturen: Praxistest bestanden?	2006	Primary	0	0
JDT05	A Service Oriented Architecture Framework for Collaborative Services	2005	Primary	0	3
Je05	Developing Composite Applications - Comparing the Total Cost of Ownership	2005	Primary	0	0
Ji05	On Procedure Strategy of Constructing SOA's Modeling Language	2005	Primary	0	1
JMP04	Business Process Execution Language for Web Services	2004	Secondary	2	0
Jo04	Concepts for Modelling Enterprise Architectures	2004	Secondary	1	0
Jo05	Toward an Acceptable Definition of Service	2005	Primary	2	2
JRM05	QoS Aggregation in Web Service Compositions	2005	Secondary	1	0
JS05	Service-Oriented Paradigms in Industrial Automation	2005	Primary	0	1
JW05	Service-Oriented Architecture for Deploying and Integrating Enterprise Applications	2005	Primary	0	4
Ka03	Loosely Coupled-The Missing Pieces of Web Services	2003	Secondary	1	0
Ka05	Analysis and simulation of business solutions in a serviceoriented architecture	2005	Primary	0	4
Ka06	Four Steps to SOA	2006	Primary	0	0
KBS05	Enterprise SOA	2005	Primary	3	4
KC05	Service-Oriented Design with Aspects (SODA)	2005	Primary	0	1
Ke04	Patterns: Implementing an SOA Using an Enterprise Service Bus	2004	Primary	5	0
Ke05	Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture	2005	Primary	0	0
Ke06	Patterns: Extended Enterprise SOA and Web Services	2006	Primary	0	0
KEA05	A Generalized Framework for Providing QoS Based Registry in Service Oriented Architecture	2005	Primary	0	6
KJE05	Service-Oriented Software Engineering (SOSE) Framework	2005	Primary	0	3
KKP05	Reckoning Legislative Compliances with Service Oriented Architecture - A Proposed Approach	2005	Primary	0	3
KKS06	Evaluating Service Oriented Architectures (SOA) in Pervasive Computing	2006	Primary	0	1
KL03	On Web Services Aggregation	2003	Secondary	1	0
KLS05a	Service-Oriented Agility: An initial analysis for the use of Agile methods for SOA development	2005	Primary	0	4
KLS05b	Service-oriented agility: Methods for successful Service-Oriented Architecture (SOA) development, Part 1: Basics of SOA and agile methods	2005	Primary	0	3
KM04	Systematic Development and Exploration of Service-Oriented Software Architectures	2004	Primary	1	4
KMM06	Efficient Exploration of Service-Oriented Architectures using Aspects	2006	Primary	0	5
Ko05	Dynamic Policy Management on Business Performance Management Architecture	2005	Primary	0	1
KP03	Toward Web Services Management Standards-An Architectural Approach to IT Systems Design	2003	Secondary	1	0
KPR04	A Framework for Integrating Business Processes and Business Requirements	2004	Primary	0	4
Kr04a	Towards a Process and Tool-Chain for Service-Oriented Automotive Software Engineering	2004	Secondary	1	0
Kr04b	Service specification with MSCs and roles	2004	Secondary	1	0
KR05a	10 Steps to SOA	2005	Primary	0	0
Kr06a	IBM'S New Governance Push	2006	Primary	0	0
Kr06b	SOA Governance Power Play	2006	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
In06							5	5			
It06		20	3			54	11	5	16		3
JDT05						1					4
Je05	1	1	2			1	1	1			
Ji05											
JMP04											
Jo04		1	6			2					
Jo05							8	7			
JRM05											
JS05			1			1	7	4			
JW05							1				
Ka03											
Ka05	2		1	4		27					
Ka06	1	1			1						
KBS05											
KC05											
Ke04	2	3			2	3	421	212	500		
Ke05		142	5		1	4	328	98	249		
Ke06				1	2	3	351	235	36		
KEA05								2			
KJE05				8	1				2		1
KKP05		12	3			6					
KKS06											
KL03						2	1	15			
KLS05a	2	1	3		1	5					
KLS05b	4	1			1						
KM04				6			1	1			
KMM06				6			2	1			2
Ko05						40					
KP03				2			4	5			
KPR04				11		2					
Kr04a							1				
Kr04b				9			2				
KR05a	2	12	1			12	5	2	1		
Kr06a		9									
Kr06b		8									

Label	Title	Year	Type	Ref-In	Ref-Out
Kr95	Architectural Blueprints-The "4+1" View Model of Software Architecture	1995	Primary	0	0
Ku05	Hot Spots der Software-Entwicklung: Service-orientierte Architekturen - Anspruch und Wirklichkeit	2005	Primary	0	0
Ku06	SOA: A Phoenix rises out of COBOL's Ashes	2006	Primary	0	0
KZ04	Architecture and modeling of service-oriented systems	2004	Secondary	1	0
KZ05	Implementation of Business Processes in Service-Oriented Systems	2005	Primary	0	4
LA02	A Goal-driven approach to enterprise component identification and specification	2002	Secondary	1	0
La02	The Role of Web Services in the Financial Service Industry	2002	Secondary	1	0
La03	The State of Web services	2003	Secondary	1	0
La06a	Making SOA pay right away	2006	Primary	0	0
La06b	Service-oriented architecture: A way of Life	2006	Primary	0	0
La06c	SOA Simple	2006	Primary	0	0
Le00	Feature-based Approach to Object-Oriented Engineering of Applications for Reuse	2000	Secondary	1	0
Le03	Web Services - Distributed Applications without Limits, Business, Technology and Web	2003	Secondary	2	0
Le04	The Influence of Web Services on Software: Potentials and Tasks	2004	Secondary	1	0
Le05a	The (Service) Bus: Services Penetrate Everyday Life	2005	Primary	0	6
Le05b	Service-Oriented Solutions Approach (SOSA)	2005	Secondary	1	0
Le05c	Combining Web Services and the Grid: Towards Adaptive Enterprise Applications	2005	Secondary	1	0
Li03	Next Generation Application Integration	2003	Secondary	3	0
Li04	Migrating to Web Services: a Performance Engineering Approach	2004	Secondary	3	0
Li05	The ROI of Your SOA	2005	Primary	0	0
Li06	IT-Riesen kämpfen um die SOA-Krone	2006	Primary	0	0
LMC05	A Service-Oriented Architecture for Design and Development of Middleware	2005	Primary	0	3
LMS05	SMART: The Service-Oriented Migration and Reuse Technique	2005	Secondary	1	0
LMS06	Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture	2006	Primary	0	4
Lo06a	Managing SOA Performance	2006	Primary	0	0
LRS02	Web services and business process management	2002	Secondary	3	0
LT03	Dissecting Service-Oriented Architectures	2003	Secondary	1	0
Lu05a	An Investigation on Service-Oriented Architecture for Constructing Distributed Web GIS Application	2005	Primary	0	2
Lu05b	Template Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle	2005	Primary	0	0
Ma03	Service oriented architectures for supporting environments in eGovernment applications	2003	Primary	0	0
Ma04a	Electronic Business Service Oriented Architecture: Service Broker - Proxy Pattern	2004	Secondary	1	0
Ma04b	Systematic Definition, Implementation and Evaluation of Service-Oriented Software	2004	Secondary	1	0
Ma05a	Toward Web Services Interaction Styles	2005	Primary	0	0
Ma05b	Web Services: What's Real and What's Not?	2005	Primary	0	0
Ma05c	Transformation of UML Models for Service-Oriented Software Architectures	2005	Primary	0	0
Ma05d	Orchestration Patterns in Service Oriented Architectures	2005	Secondary	1	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
Kr95											
Ku05		12	5	2	1	8	30	10	3		
Ku06					1						
KZ04											
KZ05						2	1		2		
LA02	1		1	2							
La02											
La03											
La06a					1						
La06b						1			1		
La06c							3		1		
Le00											
Le03											
Le04											
Le05a						2		3			
Le05b				3	1						1
Le05c											
Li03											
Li04											
Li05					6						
Li06						2			6		
LMC05								1			
LMS05							4	2			
LMS06				10			3	1			
Lo06a		4					1		1		
LRS02						18	11	9			
LT03			1			2					2
Lu05a							3	7			
Lu05b											
Ma03		1		1							
Ma04a							13	2			1
Ma04b											
Ma05a											
Ma05b							42	16			
Ma05c				7							
Ma05d											

Label	Title	Year	Type	Ref-In	Ref-Out
Ma06a	Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service	2006	Primary	0	0
Ma06b	The ICC and SOA Governance	2006	Primary	0	0
Ma06c	Service-Oriented Architecture Craves Governance	2006	Primary	1	0
Ma06d	Sizing up your Business Processes	2006	Primary	0	0
MBE03	Composing Web services on the Semantic Web	2003	Secondary	1	0
MPP02	Modeling e-service orchestration through Petri nets	2002	Secondary	1	0
Me04	Composing Web Services. A QoS View	2004	Secondary	1	0
MG03	Practical Approaches to Service Oriented Architectures	2003	Secondary	1	0
Mi05a	Business-driven development	2005	Primary	0	0
Mi05b	A case for SOA governance	2005	Primary	0	1
MKC04	Cooperative Middleware Specialization for Service Oriented Architectures	2004	Primary	1	3
KKO03	Stepwise refinable service descriptions: Adapting DAML-S to staged service trading	2003	Secondary	1	0
MLD05	A MDE Approach for Power Distribution Service Development	2005	Primary	0	3
MM04	Current Solutions for Web Service Composition	2004	Secondary	1	0
Mo02	Dynamic e-business: Trends in web services	2002	Secondary	1	0
Mo06	Web Services Governance	2006	Primary	0	0
MPT99	Service provision and composition in virtual business communities	1999	Secondary	1	0
MR03	Technical Note: Web service credentials	2003	Secondary	1	0
MSR05	Service-Oriented Design: The Roots	2005	Primary	0	0
MVA05	SOA-based Integration of IT Service Management Applications	2005	Primary	0	2
Na02	Model-Checking Verification for Reliable Web Service	2002	Secondary	1	0
Na03	Service-Oriented Architecture Scenario	2003	Primary	0	0
Na05	Applied SOA: Conquering IT Complexity Through Software Architecture	2005	Primary	0	0
Ne02	Understanding Web Services. XML, WSDL, SOAP, and UDDI	2002	Secondary	1	0
Ne04	Understanding SOA with Web Services	2004	Secondary	1	0
Ne05	A Service-oriented Workflow Language for Robust Interacting Applications	2005	Secondary	1	0
NG05	Events and service-oriented architecture: The OASIS Web Services Notification specifications	2005	Primary	0	3
NM02	Simulation, Verification and Automated Composition of Web Services	2002	Secondary	2	0
No06	Can There Be Too Much SOA?	2006	Primary	0	0
NP03	Automotive Infotronics: An emerging domain for Service-Based Architecture	2003	Secondary	1	0
NS03	Introduction to Service-Oriented Architecture	2003	Primary	0	0
OA04	Web Services Business Process Execution Language Version 2.0, Draft 1	2004	Secondary	1	0
OHE03	Capabilities: Describing What Services Do	2003	Secondary	1	0
OY05b	Establishing and Maintaining Compatibility in Service Oriented Business Collaboration	2005	Secondary	1	0
OY05a	Bridging the Gap between Business and IT in Service Oriented Business Collaboration	2005	Primary	0	3
OYP05	A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration	2005	Primary	1	13
Pa02	Semantic Matching of Web Services Capabilities	2002	Secondary	1	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
Ma06a									14		
Ma06b	3	18									
Ma06c		43	1		2						
Ma06d	1	2	1						1		
MBE03				3			14	22			
MPP02											
Me04											
MG03											
Mi05a			1		4	11					
Mi05b	2	89	5								
MKC04							2	20			
KKO03											
MLD05											
MM04							3	3			
Mo02						3	11	4			
Mo06		12						2			
MPT99											
MR03							4	8			
MSR05							3	3			
MVA05								3			
Na02								8			
Na03						6	2	3			
Na05			2			1	5		3		
Ne02											
Ne04											
Ne05				3		1					
NG05							4	4	3		
NM02								1			
No06							3				
NP03				2			1	1			
NS03							2	3			
OA04											
OHE03			1					4			
OY05b			3	2							
OY05a											
OYP05			4			2		2			
Pa02							4	7			

Label	Title	Year	Type	Ref-In	Ref-Out
Pa03	Service-Oriented Computing: Concepts, Characteristics and Directions	2003	Secondary	6	0
Pa05a	How BPEL and SOA Are Changing Web Services Development	2005	Primary	0	2
Pa05b	Impact of SOA on Enterprise Information Architectures	2005	Primary	1	1
Pa05c	An Introduction to Service-Oriented Architecture from a Java Perspective	2005	Secondary	1	0
Pa06	Event Strem Processing and Event Driven Architectures	2006	Primary	0	1
PB03	Guide to Enterprise IT Architecture	2003	Secondary	1	0
Pe03a	Web Services Orchestration and Choreography	2003	Secondary	2	0
Pe03b	Web services orchestration: a review of emerging technologies, tools, and standards	2003	Secondary	2	0
Pe04	A service-oriented architecture for digital libraries	2004	Secondary	1	0
PG03	Service-Oriented Computing	2003	Primary	5	0
Pi04	Planning and monitoring web service composition	2004	Secondary	1	0
PL03	Service-Oriented Architecture	2003	Primary	0	2
Pl05a	Software Architectures Will Evolve From SOA and Events to Service Virtualization	2005	Primary	0	0
Pl05b	Management Update: Six Missteps That Can Result in SOA Strategy Failure	2005	Primary	0	0
Pl05c	Whitepaper IT-Architektur	2005	Primary	0	0
Po06	Telecom Services Delivery in a SOA	2006	Primary	0	3
PS02	From E-Processes to E-Networks: an E-Service-oriented Approach	2002	Secondary	1	0
PSZ01	Service-oriented Modelling for e-Business Applications Components	2001	Primary	0	1
PW04	Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications	2004	Seconary	1	0
PY04	Design Methodology for Web Services and Business Process	2004	Secondary	2	0
QDD04	Methodological support for service-oriented design with ISDL	2004	Secondary	4	0
QDS05	An approach to relate business and application services using ISDL	2005	Primary	0	9
Ra03	A Model for Web services Discovery with QoS	2003	Secondary	2	0
RB05	A Categorization of Collaborative Business Process Modeling Techniques	2005	Primary	0	2
RD05a	Managing End-to-End Lifecycle of Global Service Policies	2005	Primary	0	0
RD05b	Towards a Distributed Service-Oriented Business Rules System	2005	Primary	0	6
RD05c	Business Rules Integration in BPEL - A Service-Oriented Apporach	2004	Secondary	1	0
RD05d	Design and Implementation of a Service-Oriented Business Rules Broker	2005	Secondary	2	6
RD05e	Business Rules Integration in BPEL - A Service-Oriented Approach	2005	Primary	0	0
Ri05	Integrating Service Specifications at Different Levels of Abstraction	2005	Primary	0	4
Ro04	Understand Enterprise Service Bus Scenarios and Solutions in Service-Oriented Architecture	2004	Secondary	2	0
Ro05	Der optimale Weg zur Geschäftsagilität: SOA oder BPM?	2005	Primary	0	0
SA05	Enterprise Services Architecture: Das Konzept für den Wandel	2005	Primary	0	0
SB96	Analysing the Return of Investment of Reuse	1996	Primary	0	0
Sc02	Web services based execution of business rules	2002	Secondary	1	0
Sc03	Hype Cycle for Application Integration and Platform Middleware	2003	Secondary	1	0
Sc05	The Enterprise Service Bus: Making service-oriented architecture real	2005	Primary	3	3

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
Pa03							2	4			
Pa05a						2	4	31	3		
Pa05b									5		
Pa05c							15	9			2
Pa06	1								7		
PB03											
Pe03a				4		4		21			
Pe03b	1			8		9	8	39			
Pe04							2	10			
PG03							1	1			2
Pi04				1				1			
PL03			1								
Pl05a									4		
Pl05b		2							1		
Pl05c		7									
Po06							1	10	3		
PS02			1			1	3	3			
PSZ01											
PW04							16	9			
PY04							3	7			
QDD04											
QDS05								1			
Ra03							9	5			
RB05						10		1			
RD05a	7							1			
RD05b							3	1	1		2
RD05c				1		1			2		
RD05d								1			6
RD05e				1		1			3		
Ri05							3	1			
Ro04							5	4	8		
Ro05					1	32					
SA05											
SB96											
Sc02							32	1			
Sc03					2	10	4	3	6		
Sc05						1	14	5	4		

Label	Title	Year	Type	Ref-In	Ref-Out
Sc06	Security in Enterprise Resource Planning Systems and Service-Oriented Architectures	2006	Primary	0	0
Sc92	Architecture for Integrated Information Systems - Foundations of Enterprise Modeling	1992	Secondary	1	0
SDS04	Modeling and Design of Service-Oriented Architecture	2004	Primary	0	3
SEH02	What's in a Service? Towards accurate description of non-functional service properties	2002	Secondary	2	0
SEH05a	The Price of Services	2005	Primary	0	2
SEH05b	Formal description of non-functional service properties	2005	Secondary	1	0
SG96	Software Architecture, Perspectives on an Emerging Discipline	1996	Secondary	3	0
Sh02	QoS for Service-oriented Middleware	2002	Secondary	1	0
Sh03	Designing Adaptive Components for a Services Oriented Architecture	2003	Primary	0	1
Sh04	Building a service-oriented eBanking Platform	2004	Primary	1	2
SH05	Service-oriented Computing - Semantics, Processes, Agents	2005	Secondary	3	0
Si04	Framework for Semantic Web Process Composition, Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce	2004	Secondary	1	0
Si05	Data and Metadata Management in Service-Oriented Architectures: Some Open Challenges	2005	Primary	0	0
Sk05	Service Oriented Architecture Realizing the Business Value	2005	Primary	0	0
Sk06	SOA Starts With Design and Delivery	2006	Primary	0	0
SLC05	Towards Agricultural Marketing Reforms: Web Services Orchestration Approach	2005	Primary	0	5
Sm04	Web Services Enable Service-Oriented and Event-Driven Architectures	2004	Secondary	1	0
SM05	The service engineering area: An overview of its current state and a vision of its future	2005	Secondary	1	0
SM06	Architectural translucency in service-oriented architectures	2006	Primary	0	3
SN03	Event-Driven Architecture Complements SOA	2003	Primary	1	0
Sn06a	Wrapping Legacy Software for Reuse in a SOA	2006	Primary	0	6
Sn06b	Integrating legacy Software into a Service oriented Architecture	2006	Primary	0	6
So06a	A new Service-Oriented Architecture (SOA) maturity model	2006	Primary	0	0
So06b	Zentrale Ergebnisse der SOA-Studie 2006	2006	Primary	0	0
Sp02	Service Oriented Process Matters	2002	Secondary	1	0
Sp05	Modeling Cooperative Business Processes and Transformation to a Service Oriented Architecture	2005	Primary	0	2
SS04	Challenges of service-oriented architectures	2004	Secondary	1	0
St05	Aligning Business and IT: The Process-Driven Architecture Model	2005	Primary	0	7
St06	Using Architectural Patterns and Blueprints for Service-Oriented Architecture	2006	Primary	0	4
St97	Hierarchical Service Definition	1997	Secondary	1	0
SUN02	Using Web Services Effectively	2002	Secondary	1	0
SUN04	Assessing Your SOA Readiness	2004	Primary	0	0
SUN06a	SOA Governance Solution - Accelerating SOA	2006	Primary	0	1
SUN06b	The SOA Platform Guide: Evaluate, Extend, Embrace	2006	Primary	0	0
SUN06c	Sun SOA Opportunity Assessment - A Pragmatic Approach	2006	Primary	0	0
SUN06d	Systematic Development and the Service Oriented Architecture	2006	Primary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
Sc06											
Sc92											
SDS04	2						4	6			
SEH02						1		6			
SEH05a						1					
SEH05b						1	2	5			
SG96											
Sh02							4	7			
Sh03			2				4	2			
Sh04											
SH05											
Si04					1	2	1	66			
Si05											
Sk05	4	2	2				2		1		
Sk06		1			18	1					
SLC05				1		2	2	4			
Sm04							1	2			
SM05							1	2	1		
SM06				3				1	2		
SN03						8	3	4			
Sn06a					1	1					
Sn06b	5		4	1			4	15			
So06a	4	20	2		3	3	3	2	16		
So06b					1						
Sp02											
Sp05						1	2	9			
SS04									2		
St05	2	2	3		2	7	1	1			3
St06							1		1		
St97											
SUN02											
SUN04	1	3	6		2	1	4	4			
SUN06a		19					1				
SUN06b	1	7			1		7	1	3		
SUN06c		1	1		2	2					
SUN06d		4				6	3	3	1		

Label	Title	Year	Type	Ref-In	Ref-Out
SVA04	SOA and Web Services Offer Little Vendor Independence	2004	Primary	0	0
SVW04	Data Consistency in a Heterogeneous IT Landscape: A Service Oriented Architecture Approach	2004	Primary	0	3
SW04	Automated Planning in a Service-Oriented Architecture	2004	Primary	0	7
Ta03	A Standards-based Web Service Framework	2003	Secondary	1	0
TAC05	Charging for Information Services in Service-Oriented Architectures	2005	Primary	0	2
TBB03	Turning Software into a Service	2003	Secondary	2	0
TCP05	Specification-Based Verification and Validation of Web Services and Service-Oriented Operating	2005	Secondary	1	0
TG03	Seamless UML Support for Service-Based Software Architectures	2003	Secondary	1	0
Th05	Management Update: Predicts 2006: The Strategic Impact of SOA Broadens	2005	Primary	0	0
Ti04	On the business value and the technical challenges of adopting Web services	2004	Secondary	2	0
Ti06a	Business Process Management on an SOA Foundation	2006	Primary	0	0
Ti06b	Service-Oriented Architecture and Event-Driven Architecture	2006	Primary	0	0
TJ05	A Service-Oriented Architecture Framework for Mobile Services	2005	Primary	0	3
TJH05	Applying SOA to Intelligent Transportation System	2005	Primary	0	1
TP02	An Overview of Standards and Related Technology in Web Services	2002	Secondary	1	0
Tr04	Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development	2004	Secondary	1	0
Ts04	A Software Reliability Model for Web Services	2004	Secondary	1	0
Ts05	Service-Oriented System Engineering: A New Paradigm	2005	Primary	0	4
Tu04	Enterprise Architecture Technology Assessment	2004	Secondary	1	0
TYL05	Realising Service-Oriented Architecture through Context-Based Dynamic Service Invocation	2005	Primary	0	7
Vi02	Web Services Interaction Models. Current practice	2002	Secondary	1	0
Vi06	The Social Side of Services	2006	Primary	0	0
VM06	Service Oriented Architecture (SOA) Implications for Large Scale Distributed Health Care Enterprises	2006	Primary	0	2
Vo03	Web services are not distributed objects	2003	Secondary	2	0
W3C02a	W3 Consortium Web Services Initiative	2002	Secondary	2	0
W3C02b	W3C Web Services Description Working Group	2002	Secondary	1	0
W3C02c	Web Services Description Language (WSDL)	2002	Secondary	2	0
W3C04a	Web Services Architecture	2002-2	Secondary	4	0
W3C04b	Web Services Description Language (WSDL) Version 2.0 Part 0: Primer	2004	Secondary	1	0
Wa04	Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architecture	2004	Secondary	2	0
Wa05	A Policy-Based Approach for QoS Specification and Enforcement in Distributed Service-Oriented Architecture	2005	Primary	0	1
WC03	Developing Enterprise Web Services: An Architect's Guide	2003	Secondary	1	0
WDL04	Introducing EAI and Service Components into Process Management	2004	Primary	0	1
We04	IT Governance: How Top Performers Manage IT Decision Rights for Superior Results	2004	Secondary	2	0
We05	Web Services Platform Architecture	2005	Secondary	0	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
SVA04						1	24	19			
SVW04							17		4		
SW04						2					
Ta03						5	20	12			
TAC05							4				
TBB03							6	13			1
TCP05							8	7			
TG03											
Th05		8			2	1					
Ti04	19				3		14	8			
Ti06a		1				34			1		2
Ti06b						2					
TJ05											
TJH05				1				4			
TP02							82	44			
Tr04				2				1			
Ts04				1			5	5			
Ts05				1			3	2			
Tu04											
TYL05											
Vi02					1		4	9			
Vi06	1										
VM06			2				3	2			
Vo03							35	7			
W3C02a											
W3C02b											
W3C02c											
W3C04a											
W3C04b											
Wa04							1	1			
Wa05											
WC03											
WDL04						1		5			
We04											
We05											

Label	Title	Year	Type	Ref-In	Ref-Out
WEI06	A Case Study in SOA and Re-Architecture at Company ABC	2006	Primary	0	3
Wi06	SOA Governance: Rules of the Game	2006	Primary	0	0
WK05a	Service-Oriented Business Integration and Management in Telecom	2005	Primary	0	4
WK05b	SOA: EAI in neuen Schläuchen?	2005	Primary	0	0
WT04	A Role based Access Control for Web Services	2004	Secondary	1	0
WYH05	A Service Modeling Approach with Business-Level Reusability and Extensibility	2005	Primary	0	6
Ya03	Web Service Componentization: Towards Service Reuse and Specialization	2003	Secondary	1	0
YIM05a	A Lightweight Formal Framework for Service-Oriented Applications Design	2005	Primary	0	8
YIM05b	Towards a mathematical foundation for service-oriented applications design	2005	Secondary	1	0
YK97	Relationship between Service Customers' Quality Assurance Behaviors, Satisfaction, and Effort: A Cost of Quality Perspective	1997	Secondary	1	0
Yu04	Developing End-User Programmable Service-Oriented Applications with VINCA	2004	Secondary	1	0
Za03	Service-Oriented Consulting: Facilitating the Service-Oriented Enterprise	2003	Secondary	1	0
Za87	A Framework for Information Systems Architecture	1987	Secondary	3	0
Za96	Concepts of the Framework for Enterprise Architecture	1996	Secondary	1	0
ZCC04	Migration to web services oriented architecture: a case study	2004	Secondary	2	0
Ze03a	Quality Driven Web Services Definition	2003	Secondary	1	0
Ze03b	Flexible Composition of Enterprise Web Services	2003	Secondary	3	0
Ze04	QoS-aware middleware for Web services composition	2004	Secondary	2	0
Zi04	Second generation web services-oriented architecture in production in the finance industry	2004	Secondary	2	0
Zi05	Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario: Rationale, Concepts, Lessons Learned	2005	Primary	0	7
ZKG04	Elements of Service-Oriented Analysis and Design	2004	Secondary	5	0
ZTP03	Perspectives on Web Services - Applying SOAP, WSDL and UDDI to Real-World Projects	2003	Secondary	2	0
Zy02	Application Assembly using Web services	2002	Secondary	1	0
ZY04a	Incubating Services in Legacy Systems for Architectural Migration	2004	Primary	0	8
ZY04b	One-Stone-Two-Birds: Legacy System Re-engineering and Web Services Development - A Component-Based and Service-Oriented Approach	2004	Secondary	1	0

Label	Business value	Governance	Alignment	Case Study	ROI	BPM	SOAP	WSDL	ESB	Web Service	Service Layer
WEI06				2		1	5	2	1		
Wi06		48					3	2			
WK05a						1	4	3	2		
WK05b		5					2		2		2
WT04							7	1			
WYH05								1			
Ya03											
YIM05a											
YIM05b							2	2			
YK97				1			1	1			
Yu04								4			
Za03											
Za87	2										
Za96											
ZCC04				18	1		16	5			2
Ze03a							7	3			
Ze03b				1			4	5			
Ze04						1	2	2			
Zi04					1	1	51	34	1		
Zi05	2		1			1	15	25	1		
ZKG04						33	1	2			
ZTP03											
Zy02					1		3	3			
ZY04a					1		5	5			
ZY04b											

Book	28
Not available / Not found	18
WebPage	22
PDF	314
HTML	26
Printout	5
Not searchable	2
	415

Appendix B

Complete graph