Service Engineering in Business Ecosystems

Holger Kett[1], Konrad Voigt[2], Gregor Scheithauer[3], Jorge Cardoso[4]

[1]*Fraunhofer IAO,* [2,4]*SAP Research,* [3]*University of Bamberg*

Service providers are increasingly forced to cooperate to strengthen their core competencies and to create partner networks to offer new services. In order to facilitate the collaboration process, business ecosystems have been building which allow service providers to offer their own as well as combined e-services with other providers over the Internet. This paper introduces the Inter-enterprise Service Engineering methodology (ISE) which has been developed in the Texo use case[1] of the Theseus project[2] and provides a comprehensive, interdisciplinary, and model-driven approach to e-service development in business ecosystems. One major aim of ISE is to integrate business as well as technical stakeholders in the e-service development process and support them with appropriate models, methods, and tools.

1. Introduction

All industrial nations feature large and fast growing service sectors. In Germany, for example, 72 percent of the labour force in 2007 earns its living within the service sector. This sector has been the fastest growing sector in Germany for more than 15 years[3]. In other industrial countries the situation and the trend are comparable (Maglio et al., 2006). Due to this trend, service providers reveal a strong interest for service innovations and their development which will enable them to achieve a competitive market position and to participate in future economic growth (Rai & Sambamurthy, 2006).

In order to reach a competitive advantage, companies should not offer a broad range of services but rather focus on selected core competencies. Ideally, core competencies will enable to offer services which their customers perceive as better than the ones from competitors. However, customers tend to require not only individual services, but they ask for complete and integrated solutions (Bieger & Rüegg-Stürm, 2002). This trend urges companies to collaborate with partners, even with former competitors, to offer the appropriate and required services.

[1] http://www.theseus-programm.de/scenarios/de/texo

[2] The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference “01MQ07012”. The authors take the responsibility for the contents.

[3] See Ergebnisse der Erwerbstätigenrechnung in der Abgrenzung der Volkswirtschaftlichen Gesamtrechnungen (VGR) at http://www.destatis.de

Having this background, business ecosystems have been developed where independent companies collaborate to offer services concertedly. The ecosystems consist of highly networked social and technical resources which produce an economic value by cooperating (Kagermann & Österle, 2006). In this context, services are an integral part of a value chain network which will take up a more important role in the future. In order to realize this potential, services need to become tradable goods similar to products (Janiesch et al., 2008).

An appropriate infrastructure is needed to offer and provide these services via the Internet. The ability to combine services is a central and crucial feature to develop new and innovative services on the basis of existing ones. Services of different providers can be combined and integrated. Here, the focus lays on web-based services (so-called e-services) which are accessible via the Internet (Janiesch et al., 2008).

2. Objective and Structure

Many methodologies exist to support the development of services. Some of the approaches are business oriented (e.g. New Service Development and Service Engineering (Daun & Klein, 2004)) while others have a stronger emphasis on more technical aspects (e.g. software developing methodologies based on the OSI model or on Service-Oriented Architectures). Depending on the orientation, most of the approaches focus either on business or on technical aspects of the service realization. Therefore, the approaches lack an interdisciplinary perspective (Buhl et al., 2008). Nevertheless, new interdisciplinary approaches are emerging in the context of Service Science which aims to academically examine the field of service development (Chesbrough & Spohrer, 2006). So far, we could not identify suitable approaches from Service Science which address the development of e-services in business ecosystems.

The objective of this paper is to introduce a new methodology which supports the development process of e-services from the creation of a business model to the implementation. The methodology is called Inter-enterprise Service Engineering (ISE) and provides a basis for companies which are interested in participating in the cooperative e-service development process within business ecosystems.

This paper is structured as followed. In section 3, the strength and weaknesses of current business and technical oriented approaches are shown and the need for an interdisciplinary Service Engineering methodology which covers business as well as technical aspects is explained. The ISE methodology is introduced in section 4 based on selected methods of Service Engineering and Computer Science. Their integration into a broad and interdisciplinary methodology is described. Section 5 presents our conclusions and our future work.

3. State of the Art in Service Development

Prior to participating in business ecosystems, companies need to have an understanding of e-service development and integration of business partners and their e-services. In order to facilitate this process an appropriate methodology should be drawn which provides guidelines for the e-service development process for companies joining business ecosystems. We have examined current approaches for their suitability to e-service development. Depending on the different types of stakeholders who are involved in the service development process, e.g. business developers, business analysts, IT analysts, and IT developers, the underlying understanding of a service differs. The involved stakeholders and their service definitions can be divided into two main types which are influenced by a business and a technical understanding of services:

A **business-oriented service definition** contains often the following features (Corsten, 1997;Rai & Sambamurthy, 2006;Chesbrough & Spohrer, 2006)):

- ***Immateriality***,
- ***Production and consumption happen simultaneously*** (Uno-actu-principle),
- ***Integration of the customer*** as an external factor in the service delivery process.

When analyzing e-services from a business perspective, the traditional service view and features may not apply. For example, the Uno-actu-principle and the integration of the customer might become less applicable or even obsolete if e-services can be offered by providers and bought by customers in a similar way to products in e-shops. In this case, e-services are already produced prior to consuming by costumers (Scheer et al., 2003).

From a **technical-oriented service definition** mostly the functionality of the e-service interfaces needs to comply with certain features (Papazoglou, 2003;Papazoglou et al., 2006):

- ***Self descriptive:*** The annotation of meta-data allows the assignment of descriptions or quality criteria to services.
- ***Platform independent:*** The service is independent of the underlying soft- and hardware.
- ***Composition:*** Distributed applications and their functionality can be realized by composition and linkage.
- ***Application of standards:*** The functionality of a service is offered via network and standardized languages and protocols.
- ***Loose coupling:*** Neither context information nor further information of the internal functionality are needed by the service provider and consumer. Here, a dynamic real-time (at the time of usage) integration of the services by using accepted invocation mechanisms should be sufficient.
- ***Transparency of location:*** The service can be found and invoked independently of the location of the service provider by registration in a service directory.

The technical-oriented service definition mainly focuses on implementational aspects of e-services. The business context is widely neglected.

To identify the strengths and weaknesses of business- and technical-oriented approaches for service development, we have setup a set of criteria for their comparison. This set of criteria has been built on an aggregation of the main parts of existing service development approaches and our understanding of the essential parts of e-service development. Thus, the following criteria are compared for each type of service development approach to identify its strengths and weaknesses:

- ***Methodology:*** Provides a clear procedure which guides stakeholders through the development process.
- ***Service innovation***: Involves understanding the market and to derive ideas for new services.
- ***Business model***: Builds the basis for the business-oriented service description.
- ***Business processes***: Introduce the service delivery process.
- ***Architecture***: Provides a set of abstraction levels, views, meta models, and patterns to classify and distinguish components of services from an IT perspective,
- ***Service ecosystems***: Indicates that services are a result of a collaboration of several business partners and thus, the service development cannot only rely on techniques dedicated only for service development within one organization,
- ***Model-driven architecture (MDA)***: Offers a technology to structurally develop, integrate and support the perspectives of different stakeholders in the service development process based on models. MDA allows the transformation from one model to another model.

3.1. Business-oriented Service Development Approaches

When having a closer look at business-oriented service development approaches two main streams of scientific activities exist: New Service Development (NSD) and Service Engineering (SE) (Daun & Klein, 2004). Since the 80's NSD has mainly been driven by Anglo-American researchers with a strong marketing focus, whereas, SE has been pushed by German researchers for the past 15 years. The main aim of SE is to provide a systematic service development by applying engineering know-how from traditional product development.

Both streams include a variety of different methodological approaches which slightly differ in their phases. In order to compare the orientation between both streams, the phases of each stream have been aggregated in Table 1. On the one hand, NSD approaches include the development of a business and/or service strategy and the testing of services after development. On the other hand, some of the SE methods explicitly mention Service Provisioning and the Displacement of services which are phased out.

The focus of the approaches is mainly on traditional services which feature a direct customer contact and customer interaction. Here, a part of the service delivery process takes place in interaction with the customers. Especially those process steps in interaction with customers need to be carefully planned and implemented since they strongly influence the customer's perception. However, when e-services are sold similarly to products, those aspects may become less relevant since the e-service is already produced before it is requested and the interaction between provider and customer decreases.

	New Service Development	Service Engineering
Phases	Business/Service Strategy Development	-
	Idea Generation	Idea Search and Evaluation
	Screening and Evaluation	
	Business Analysis	Requirements Analysis
	Service Development	Service Concept
	Testing	-
	Introduction	Implementation and Introduction
	-	Service Provisioning
	-	Displacement

Table 1: Comparison of aggregated phases of models of NSD and SE[4]

In conclusion, the business-oriented approaches of service development show the following weaknesses (Daun & Klein, 2004):

- ***Insufficient details:*** In most cases only the high-level processes are described, whereas, concrete activities and methods are missing.
- ***Lack of IT-centric approaches:*** IT-issues are discussed to develop or support the provisioning of traditional services on the basis of a business-oriented service definition (Buhl et al., 2008). The development of pure e-services and the consideration of technical aspects as listed in the technical service definition are not in the focus.

Table 2 shows the comparison of the strengths and weaknesses of the business-oriented approaches.

	Methodology	Service Innovation	Business Model	Business Process	Architecture	Service Ecosystem	MDA
NSD	●	●	●	●	–	–	–
SE	●	●	●	●	–	–	–

(● aspect is covered, – aspect is not covered)

Table 2: Strengths and weaknesses of the business-oriented approaches

3.2. Technical oriented Service Development Approaches

In this section, four well-known and common approaches to technical service-oriented development are presented: SMART, SOAD, SOMA, and Motion. Each ap-

[4] The comparison is based on (Daun & Klein, 2004).

proach is briefly introduced with a focus on their strengths and weaknesses according to the criteria which were mentioned above.

Service-oriented Migration and Reuse Technique

The Service-Oriented Migration and Reuse Technique (SMART) was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (Lewis et al., 2005). SMART's strengths lay in migrating legacy applications into a Service-Oriented Architecture SOA. Enterprises can apply SMART to analyze their legacy systems and to evaluate their feasibility to integrate a SOA. The analysis includes stakeholder context, capability descriptions (as is), SOA description (to be), gap analysis, and development of a migration strategy. SEI arguments that the approach allows to adapt legacy systems to services without affecting the involved systems while exposing functionality to a large number of client applications using standard-based interfaces.

SMART lacks support for a more business-driven approach and neglects business models and business processes. Moreover, the methodology is not intended to be offered outside a company in a service ecosystem as a product nor does it embody innovation or model transformation.

Service-Oriented Analysis and Design

The Service-Oriented Analysis and Design (SOAD) methodology was developed by IBM (Zimmermann et al., 2004). SOAD results from an analysis of Object-oriented Analysis and Design (OOAD), Enterprise Architecture (EA) frameworks, and Business Process Modeling (BPM) for SOA deployment. SOAD's strength lies in the combination of the previous mentioned approaches. Additionally to the mentioned approaches, SOAD offers ad hoc service composition, semantic brokering, service discovery, quality factors, and reusability. It addresses the business operational level.

Though SOAD does not merely cover technical integration strategies, it offers no innovation phase for new services, nor does it include business models. Additionally, the solution is not intended for service ecosystems nor does it offer model transformation.

Service-oriented Modeling and Architecture

Arsaniani argues that object-oriented approaches lack support for services, flows, and components (Arsanjani, 2004). Therefore, IBM developed the approach for Service-Oriented Modeling and Architecture (SOMA) which fills these gaps. Additionally, SOMA supports the concept of service ecosystems and offers a separation of providers and consumers. SOMA's architecture provides seven layers of abstraction and a methodology which guides the modeling procedures.

While SOMA offers a business-driven top-down pattern, business models and service innovation are neglected. Additionally, this approach does not offer means for model-driven development.

Motion

The Motion (Sehmi & Schwegler, 2006) methodology was developed by Microsoft as a technology-agnostic architecture to expose organizations' business models. However, for Motion a business model merely refers to capabilities. The principle of the approach is to identify capabilities as a basis for stable projects and then derive business processes. This routine allows constructing an established view of businesses. The constructed view of an organization (i.e. the Motion Business Architecture) directs a fast decision making that is otherwise hard to reach. Motion's strengths is its well structured methodology and architecture which comprises a set of stakeholders, business capabilities, a life cycle, concepts for product and service, collaboration, and contexts.

Motion addresses the early stages of service engineering, such as business and collaboration within a service ecosystem. The approach lacks support for service innovation, business models, and model-driven development.

Table 3 shows the comparison of the strengths and weaknesses of business-oriented approaches.

	Methodology	Service Innovation	Business Model	Business Process	Architecture	Service Ecosystem	MDA
SMART	●	–	–	–	–	–	–
SOAD	–	–	–	●	●	–	–
SOMA	●	–	–	●	●	●	–
Motion	●	–	–	●	●	●	–

(● aspect is covered, – aspect is not covered)

Table 3: Strength and weaknesses of the technical oriented approaches

4. Integrated Methodology for Service Development in Business Ecosystems

The examination of both streams of service development approaches in business ecosystems shows weaknesses either on the business or on the technical side. Therefore, the integrated methodology ISE has been developed by the authors to create and combine e-services in the context of business ecosystems. The ISE approach, its challenges and its advantages as well as limitations are introduced in the next sections.

4.1. Challenges of Bridging the Business and Technical Worlds of Service Development

Information representation

Nowadays, organizations generate a wealth of information describing strategies, objectives, goals, business processes, standards, IT infrastructures, etc. This information, which can be classified has unstructured, semi-structured, and structured, constitutes a precious source to develop new e-services (the set of all the information produced by an organizations is often termed as *organizational memory*

(Vasconcelos et al., 2003)). Unstructured information usually characterizes documents such as Word files, spreadsheets, and presentations. Semi-structured information is generally associated with data instance which include schema information, such as XML. Finally, structured information can typify relational databases and formal models.

Depending on their background and skills, people tend to use different structures to define information. For example, business stakeholders tend to use unstructured data to describe future strategies, financial data, SWOT, Porter’s Five Forces, and balanced scorecards. On the other hand, IT professionals usually rely on semi-structured or structured information to describe formal models such as UML activity diagrams, MOF and BPMN processes (see Fig. 1).

It is clear that unstructured information drives numerous business processes but often organizations cannot leverage this information efficiently, leading to inconsistent communications between stakeholders, duplication of effort, poor decision-making, and higher costs (Herschel and Jones, 05). The difference between the various structure levels that are used to embody information creates a representational gap that needs to be closed in order to unify and bridge the business and IT perspectives on e-services. If this gap is not closed, unstructured information generated by business stakeholders will not be readily available to IT as part of the engineering process of e-services.

(Herschel and Jones, 05) Richard T. Herschel, Nory E. Jones, Knowledge management and business intelligence: the importance of integration, Journal of Knowledge Management, pp. 45–55, 9(4), Emerald Group Publishing Limited, 2005, ISSN: 1367-3270.

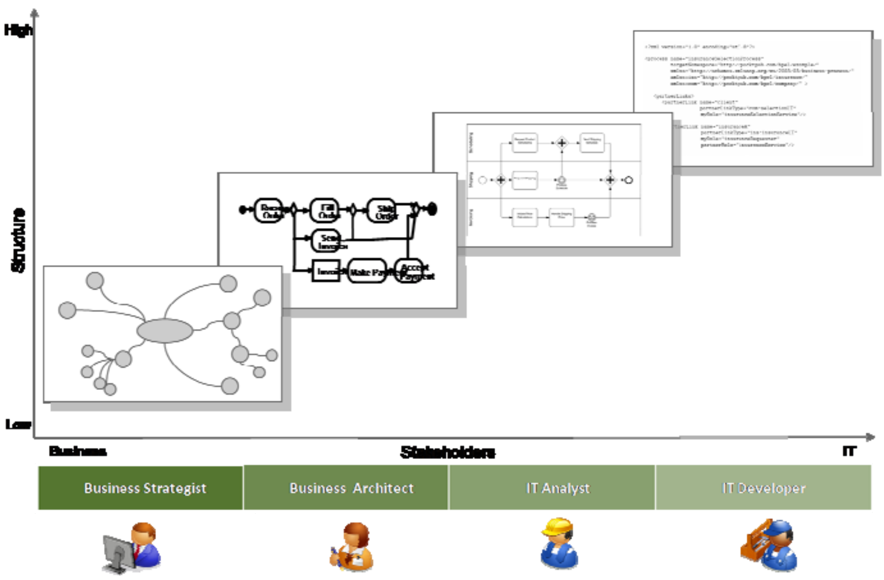

Fig. 1: Levels of structured information

Distinct Contexts and Views

Business and technical professionals have invariably different backgrounds, skills and mindsets. This fact makes the alignment of the business and IT perspectives on e-services a challenging task. On one hand, the concerns of business and IT perspectives are distinct. Business professionals are concern with aspects related to financial assets, marketing missions, ROI, competition, value chains, and SWOT analysis. On the other hand, technical people are concerned with functions, data, components, integration, compatibility, programming, scalability, API, concepts and their relationships, formal models, UI, and rules. As a result, it is often common that different people use the same word to describe distinct entities and distinct words to refer to the same entity (Fensel et al., 02) (see also section 3 which illustrates the different types of service definitions from stakeholders with different backgrounds).

(Fensel et al., 02) D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko and R. Siebes, Semantic Web Application Areas, Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Retrieved from http://www.cs.vu.nl/~mcaklein/papers/NLDB02.pdf, Accessed date 15 August 2008, 2002.

To structure heterogeneous and distributed information available inside an organization appropriate meta-level descriptions are needed to represent a higher-level layer of information. In order to develop an effective methodology for e-services, it is necessary to identify, represent, and reuse existing organizational data assets. The representation of business and IT information can be captured using formal models. When a set of models is identified and their concepts are mapped and aligned, the resulting model can be classified as an ontology. An ontology is a high level formal and explicit shared conceptualization of a domain (Gruber, 1992). It is a conceptualization since it is an abstract view of a domain about real entities and their relationships. It is formal because it has a formal software specification and it is machine-readable. It is explicit since all concepts and relationships used in the ontology are explicitly defined. Finally, it is shared since entities and relationships form a consensual knowledge, that is, not related to an individual, but accepted by a group.

A suitable methodology for e-service development can use the notion of ontology to aggregate distinct models (e.g. SWOT, Porter's Five Forces, balanced scorecards, UML activity diagrams, BPMN, etc.) from distinct communities to represent and manage both organizational information containers and contents. This option allows the representation of information in a way that it facilitates knowledge sharing and reuse between the stakeholders involved in the e-service development process**.**

Model Integration

Real world entities such as services come with a great variety of properties which characterizes them. By abstracting these properties, the level of detail is dramatically reduced and only a fraction of properties – the most important ones -- is represented in one model. Depending on the perspective that is taken on a service, specific characteristics are selected and represented in a suitable form. This abstraction can be achieved using models which contain an abstract representation of entities (information). As soon as multiple models contain representations of the same entity, the need for integration arises. For example, consider a business process modeled with BPMN which contains branching gateways with a conditional expression specified with data parameters. The definition of data parameters can be done using an ontology which is external to the business process. As a result, two different models describe or reference to the same entity and integration becomes a challenge.

The business and technical perspectives of an e-service can be separated by the level of abstraction. The more abstract level – the business level – contains informal models capturing financial information, risk analysis, goal models, etc. of e-services. The more concrete level – the technical level – describes e-service functions and technical implementations. Nevertheless, both perspectives focus on the same e-service. Consequently, they contain representations of an e-service's properties, whereas the properties of the technical world extend the ones defined in the business level. This means there is an intersection between the properties represented and, thus, the need to transform the meaning of one representation to the other one. This requires a propagation mechanism to support changes in both perspectives. The translation between both perspectives can be captured relying on the notion of mappings between formal abstract models.

Tool support

One of the challenges in the area of e-service development is to make available to engineers a suitable tool support. In order to provide a common approach, one common set of tools should be used across stakeholders. Since these tools are used by various stakeholders with different perspectives (i.e. business and technical perspectives), they should provide a global view on the models that are used to describe e-services. In order to support collaborative development of a service a common repository is needed. This includes a shared access to multiple models in the repository, i.e. every participant involved in service engineering should see the same version of a service models. This enables an instant synchronization (and visualization) of changes.

Such set of tools should not only provide a shared model access, but should also simplify the development process by providing assistance and guidance. Therefore, the construction of roadmaps is another challenge. Roadmaps guide users informing what are the tasks that need to be carry out, when and how. This way, they provide guidance on the sequence of models that should be modeled and which suitable tools to use. In this context, the most challenging task is to create an association between the development, innovation, design, and implementation phases and the integration of different representation formalisms such as unstructured information (e.g. natural language) and structured ones (e.g. formal MOF-models).

4.2. The ISE Methodology

The idea behind the ISE methodology is to take the strengths of the business- and technical oriented methodologies and combine them to provide a new approach for service engineering. Therefore, the Service Engineering from Fraunhofer IAO (business perspective) and the Zachman Framework (IT perspective) is applied in the ISE methodology. Both approaches are introduced in the next sections.

4.2.1. Service Engineering based on Fraunhofer IAO

For the integration of business-oriented aspects into the ISE methodology the Service Engineering approach from Fraunhofer IAO (Bullinger et al., 2003;Meiren, 2001) has been chosen (see Fig. 1). This approach provides a phased model based on six phases and a set of methods to model each phase in terms of product (outcome dimension), process (process dimension), resources (structure dimension), and marketing. The approach is iterative and previous phases can be refined prior to continue to the next phases.

We have chosen the SE from Fraunhofer IAO since it has a strong focus on the business-orientation in the service development process. Especially, the definition phase and the requirements analysis provide methods to design a service from the idea management over feasibility studies, market and business requirements to the service, process, resource, and marketing design. In order to support the design process, a service model has been developed which illustrates the main elements of a service concept. The activities in each phase lead to the development and the gradual refinement of the service model elements.

Especially for the definition phase and requirements analysis a variety of different business-oriented methods are used to model products, processes, resources, and marketing aspects of the service model. The methods can be, for example, Porter's Five Forces model, Resource-Based View on strategy, SWOT-analysis, Cost-Profit-Analysis, BCG-Matrix (market growth versus Market share), Competitive Strategies of Porter, the analysis of the value chain from Porter, the Four Cs – Customer, Competition, Cost, and Capabilities, and the Four Ps – Price, Product, Place, and Promotion.

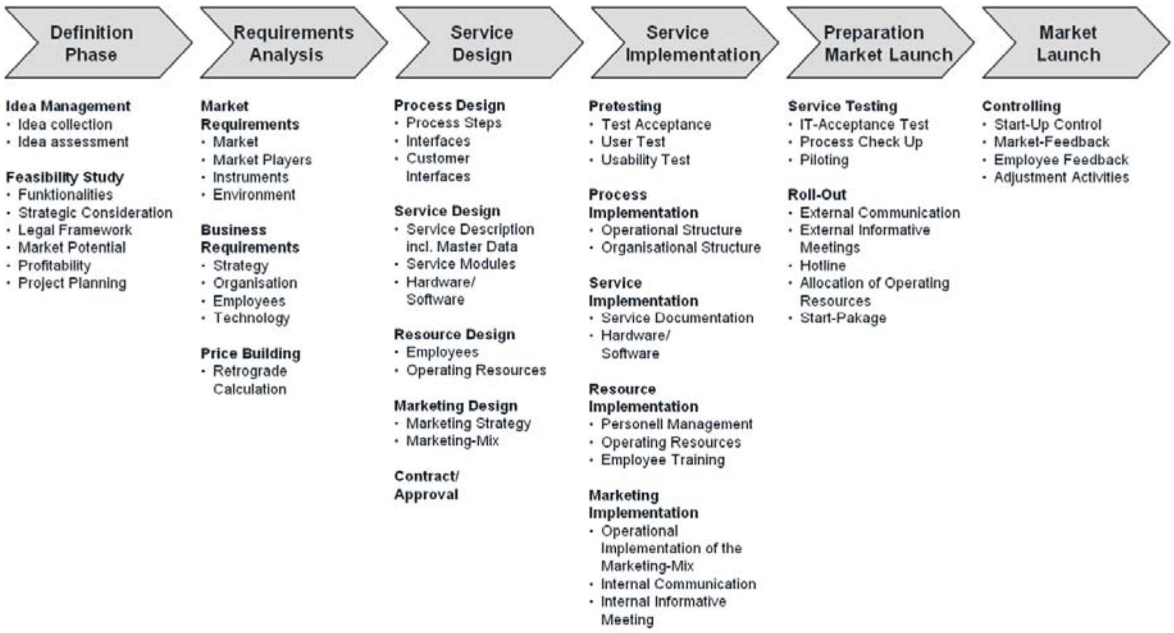

Fig. 2: Service Engineering from Fraunhofer IAO

However, the service design, service implementation, market launch and its preparation mainly focus on traditional services. Since traditional services and their development concentrate on the direct customer contact and interactions (see 3.1 Business-oriented Service Development Approaches), SE's models and methods can be further optimized and improved to develop e-services. Appropriate technical oriented models and methods are missing from SE and should also be considered.

4.2.2. Enterprise Architecture Framework based on Zachman

The Information System Architecture (ISA) framework (Zachman, 1987) provides a taxonomy to relate real world concepts to Enterprise Architecture (Sowa & Zachman, 1992). Zachman describes Enterprise Architecture as means to flexible react o business changes and to manage the varied resources of an enterprise. Zachman concluded that an architecture framework for information systems is required to integrate different perspectives of stakeholders on the architecture and, thus, to interface the different enterprise artifacts.

Therefore, Zachman breaks down an enterprise architecture into six different stakeholder perspectives and each perspective into six different descriptions as shown in Table 4 (Sowa & Zachman, 1992). The outcome of each cell is an artifact and refers to the respective description from a particular perspective of the enterprise architecture.

Zachman identifies four usages for the framework. Firstly, the framework advances the communication within the information system discipline. Secondly, by visualizing the perspectives and the different descriptions helps to understand the reasons and risks in case one or more artifacts are not developed. Thirdly, existing tools or methodologies can be set into relation, by assigning them to a specific cell. Fourthly, it helps to develop new tools and methodologies for artifacts.

The advantages of the Zachman framework for applying in the context of e-service development are its model-oriented approach on breaking down a complex enterprise architecture into small models and its differentiation of specific stakeholders' perspectives.

	Data *What*	Function *How*	Network *Where*	People *Who*	Time *When*	Motivation *Why*
Objective/ Scope (contextual) → *Role: Planner*	List of Things important in the Business	List of Core Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goals/ Strategies
Enterprise Model (*Conceptual*) → *Role: Owner*	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (*Logical*) → *Role: Designer*	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (*Physical*) → *Role: Builder*	Physical Data/ Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representations (*Out of Context*) → *Role: Programmer*	Data Definitions	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification
Functioning Enterprise → *Role: User*	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

Table 4: Zachman Framework

4.2.3. The Integration of both Methodologies in the ISE Framework

The ISE Framework supports three phases of SE: requirements analysis, service design, and service implementation. Those phases are assigned to Zachman's different abstraction layers (perspectives), i.e. strategic, conceptual, logical, and technical (see Fig. 3).

Therefore, the ISE Framework provides selected models and methods for the specific players of each layer. In order to reduce the complexity, each layer is not only based on one complex model, a layer is broken down in five dimensions: service description, workflow, people, data, and rules. A dimension is a part of an abstraction layer and consists of its own model(s) (artifact). The artifacts of each dimension can be transformed from one abstraction layer to another.

The ISE framework is built concerning the following rules:

- The dimensions do not need to be analyzed and modeled in a specific order.
- All dimensions of one layer are combined into a complete model for the layer-specific stakeholders.
- Each abstraction layer features a simple, basic, and unique model.
- The layers are separated and clearly differentiated (no overlapping).
- In order to achieve an integrated approach, information of the artifacts is transformed from one layer model to another and back (iterative approach).

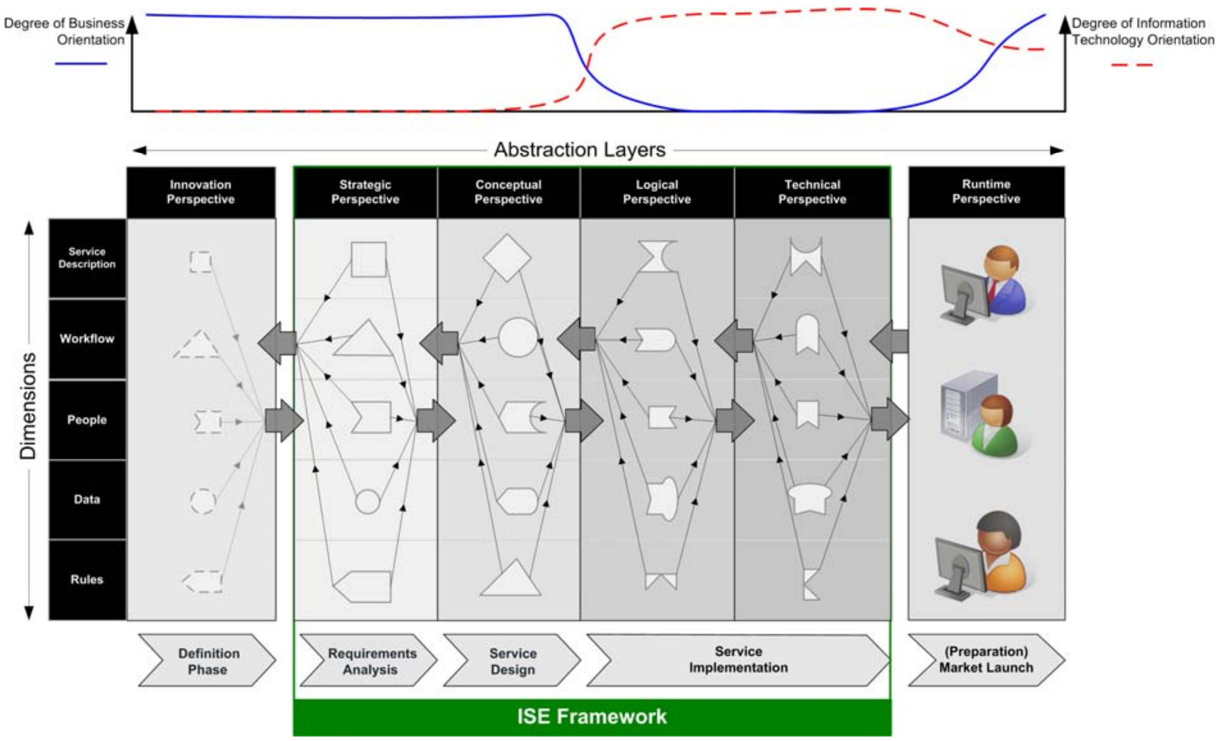

Fig. 3: ISE Framework and the degree of business/technical orientation concerning the abstraction level

Fig. 3 illustrates that the degree of business-orientation in the innovation, strategic and conceptual perspective is high and decreases in the next two perspectives when the e-services are technically designed and implemented. Whereas, in the runtime perspective the e-service is launched and offered on the market and, thus, business and technical aspects become important.

The four abstraction layers of the ISE methodology are described below:

Strategic Perspective

ISE starts with the development of the strategic perspective on an e-service. The main aim of this perspective is the identification of market and business requirements for an e-service, which are derived from the market, market players, competitors, environment, company's strategy, organization, and capabilities. The market and business requirements are identified with established business methods (see 4.2.1 Service Engineering based on Fraunhofer IAO) which are well-known among business developers. The semi-formal results of these methods are then transferred into the formal business model ontology of Osterwalder (Osterwalder, 2004). The main parts of the business model are: the service (e.g. VALUE PROPOSITION), the customer interface (e.g. TARGET CUSTOMER, DISTRIBUTION CHANNEL, AND RELATIONSHIP), the infrastructure management (e.g. VALUE CONFIGURATION, CAPABILITY, and PARTNERSHIP), and the financial aspects (e.g. COST STRUCTURE and REVENUE MODEL).

Prior to develop the strategic perspective, innovative service ideas are identified and evaluated with the target of filtering service ideas with a high business potential. In the strategic perspective, only the remaining service ideas are then in the focus of its activities.

We propose the following models forming the strategic layer:

- Service Description – The e-service itself (VALUE PROPOSITION), the way how the e-service is marketed and sold (DISTRIBUTION CHANNELS), the RELATIONSHIPS to customer groups, and the REVENUE MODEL are parts of the business model ontology which are in the focus of this dimension.
- Workflow – This dimension stresses the infrastructure management of a developing e-service and, thus, the issues of how an e-service is delivered (VALUE CONFIGURATION) and what know-how and resources are needed for the delivery (CAPABILITIES).
- Data – A list of relevant data (IMMATERIAL RESOURCES) which are required for the e-service delivering.
- People – The people model lists the relevant stakeholders who are involved in the e-service delivery and consumption (TARGET CUSTOMERS, ACTORS, AND PARTNERS).
- Rules – In this dimension the main POLICIES of a developing e-service are listed which needs to be considered and refined during the e-service development and delivery process.

In this abstraction layer, a strategic business model including a coarse-grained service concept is developed. This perspective mostly includes a decision-making process of an organization's top management. Thus, the abstraction layer is built for CEOs/CIOs/CFOs who decide for their organizations' investments in new e-service offerings. They usually do not develop the business models themselves but rather assign business developers for fulfilling the task.

Conceptual Perspective

The conceptual perspective follows the strategic perspective and refines its results. Whereas business models describe the exchange of value between business partners, process models show operational characteristics of how this is done (Dorn et al., 2007).

Thus, during the conceptual perspective business architect have the liability to operationalize and implement the strategic artifacts, which were developed during the business perspective. Business analysts refer to the architect in the Zachman Framework. Business analysts have knowledge about the market the e-service will be offered in, important processes, organizational structures, crucial assets, and domain constraints. They have the responsibility to analyze and express requirements for the e-service from this perspective. Additionally, business analysts, have knowledge about models, modeling, architecture, and transformation. Their expertise and responsibility is to transform the domain expert's perceptual requirements into diagrams. These diagrams serve as a communication basis for domain experts to agree or disagree with the e-service design analysis (Zachman, 1987).

The models for the conceptual layer must comply with the following requirements:

- Service Description – Illustration of the e-service (Baida et al., 2004) offering in terms of functionality (Oaks et al., 2003), and monetary, quality, legal and security aspects (O'Sullivan, 2006). On this basis, potential e-service consumers search, rank, compare, select, and substitute e-services. E-service can be described by the e-service property ontology introduced by O'Sullivan (O'Sullivan, 2006) and e-service classification approaches such as UNSPSC[5].
- Workflow – The aim of the workflow dimension is to show the e-service behavior. It presents tasks in sequence or in parallel which need to be carried out in order to fulfill e-service functionality. Tasks represent a company's capability. Additionally, workflow models make use of data, people, and rule concepts (Bhattacharya et al., 2007;Muehlen et al., 2007). Appropriate model notations for the workflow dimension are Event-driven Process Chains (EPC) (Scheer & Nuettgens, 2000), and UML Activity Diagram[6].
- People – The people dimension embodies organizations (actors), organizations' hierarchies, and roles within organizations. People own capabilities and carry out or are responsible for tasks within a workflow. Model notations for the people dimension are UML Use Case Diagram[6] and organizational charts.
- Data – Data conceptualize information, which is necessary to perform the different tasks described in the workflow model. Information comes from within organizations (intangible resources) and from outside organizations. Data can be de-

[5] http://www.unspsc.org/
[6] http://www.omg.org

scribed using UML Class Diagram[6] and Entity-Relationship Diagrams (Chen, 1977).

- Rule – The aim of rule descriptions (Muehlen et al., 2007) is to implement rules and policies which an e-service must comply to. Model notations for the rule dimension include Semantic Rule Model Notation (SRML) and Semantic Business Vocabulary (SBVR) (Muehlen et al., 2007).

Logical Perspective

The logical layer follows the conceptual layer and, therefore, contains the models concerned with the abstract technical implementation of an e-service. The logical layer serves as a bridge for the gap between design and implementation. This means the models of the logical layer are not executable (e.g. by a workflow engine), however the focus lies on definition of technical details, which serve as a base for a transformation into executable models. The logical layer consists of separated models with common elements, which have multiple representations in the models.

We propose the following requirements and models forming the logical layer:

- Service Description – Definition of technical non-functional Requirements like Quality of Service (QoS) in terms of time-based latency, availability etc., an e-service abstract interface and structure. An e-service interface can be described using UMPS.

- Workflow – A notation and formalism for a graph-oriented definition of an e-service underlying workflow is required. For example BPMN is used to model the workflow.

- Data – The data model should contain a data model capturing all associated information as well as data artifacts required by other models (e.g. the workflow or people model). An OWL UML-Profile (Brockmanns, 2007) is applied to the data model. It is combining the visual notation of UML and the expressiveness of OWL, so a semantic data model can be defined.

- People – The model contains an abstract representation of graphical interaction with an e-service connected to the workflow. It also captures a flow of interaction regarding the interface to the users (actors) involved with an execution of an e-service. The people column is covered by Diamodl (Trætteberg, 2006), which supports modeling of user interaction and data flow between user interface components

- Rules – We propose to select a visual modeling of rules using a common notation which defines constrains for the semantic data model and conditions for workflow and interface definition. In order to define constraints for the data model we propose the usage of an F-Logic UML-Profile, which supports the modeling of logical rules using the UML notation.

This layer involves the IT-analyst which refines and creates the final version of the

workflow in form of a process by adding constructs for error handling, fault-tolerance, transactions, etc. Furthermore he is responsible for interaction modeling and the enrichment of the models by technical details.

Technical Perspective

The technical layer is targeted at the definition of technical specifications using established standards. It serves as an aggregator of the models of abstract technical information provided by the logical layer. This formal information is transformed into executable or interpretable models. These models provide the full information in order to generate technical specifications using a model-to-text transformation (code generation). The goal is to prepare a full technical specification of an e-service, which enables automatic deployment and execution.

We propose the following requirements and models forming the technical layer:

- Service Description – Definition of the interface of an e-service including provided operations, data types, transactional behavior, security aspects etc. WSDL and related WS-standards, like WS-Security or WS-Transaction. We chose them since they are a well established and accepted way of using XML to represent an e-service interface and related information. Furthermore this integrates the Web-Service world within the ISE-Framework.

- Workflow – An executable specification of an e-services underlying workflow which can be run by a process engine. Using BPEL allows expressing the underlying workflow in an executable language with a variety of supporting engines.

- Data – A formal representation of e-services data artifacts, including structures used by the process, data types exposed via an interface and aspects describing an e-service, which should be represented in a semantic data model. Here OWL is used as a representation of semantic information.

- People – A specification of an user interface allowing interaction with the e-service. This specification fully describes a user interface; such it is interpretable by an engine. The CAP (Constantine, 2003) notation is used for describing an abstract user interface, i.e. interaction elements like buttons, text fields, output fields, etc. and their containers describing their relative alignment. Information like color, size and style is up to the generated code.

- Rules – The rules are derived from the graphical representation of the logical layer. Therefore this model represents the rules, such as they are interpretable by a rule engine. We propose the use of F-Logic, since it allows the definition of constraints over an ontology, which connects to the data dimension.

The role concerned with the technical layer is named IT Developer. His responsibility covers the implementation of the technology that will perform the processes, as well as the refinement of all other models.

Validation – ISE Workbench

Concepts and methodologies of the last years have shown that a theoretical foundation without tool support does not usually achieve acceptance. The reason is twofold. On the one hand, tool support lowers the entry barrier for users by providing them guidance and assistance. In case of very complex problems tool support may be crucial, since a user may not be able to cope with complex information and structures. On the other hand, tool support is a first validation of concepts and a methodology by proving its feasibility and revealing potential problems.

Therefore, we decided to implement the ISE-Framework in the ISE-Workbench. Due to the model-driven approach of ISE, the ISE workbench offers a set of model-based tools according to the models which are defined for each ISE-layer. All of these tools are based on a common formalism (for model representation). This allows applying automatic model transformations by using a model transformation engine. Furthermore, concepts like integrated validation, common persistency, model federation, etc. can be applied. The frame of our workbench is provided by Eclipse[7]. The Eclipse Modeling Framework (EMF) serves as the base for model definition and also provides a repository implementation supporting collaborative modeling, satisfying the requirements of service engineering in business ecosystems.

Using these constraints (Eclipse and EMF) we selected a number of existing tools like a graphical BPEL-Toolset, a graphical BPMN-Editor, graphical UML-Tools, etc. Furthermore the ISE-Workbench will provide an entry point to the ISE-Framework by visualizing the model selection and path through the model of the ISE-layers. The ISE-workbench guides a user through the development process and visualizes model dependencies. Besides providing user guidance, tool aggregation and model management, it also implements automatic support for model dependencies by using model transformation. This work should be based on a standard language for model transformation rule specification like Query, View and Transformation (QVT) provided by the OMG.

4.3. Benefits of the ISE Methodology

Since the concept of e-services is still in a ramp up phase, no customized methodologies exist for the engineering of e-services. As a result, it is inevitable that front-runners will carry out e-service development planning in a manual, ad hoc, subjective, time consuming, and error-prone fashion. This will lead to e-service solutions which are disorganized, behind schedule, over budget, or cancelled. Therefore, the development of the ISE framework provides key design practices and artifacts to the planning of e-services. The nature of ISE brings a set of benefits to the creation of e-services:

- ***Active engagement:*** All stakeholders (e.g. business strategists, business architects, IT analysts, and IT developers) that can influence the engineering of e-services participate in the development to guarantee that business objectives are accomplished. ISE supports those stakeholders by providing specific perspectives on the e-service development.
- ***Model-based approach:*** The ISE framework supports the engineering of e-services by a set of models that describe or specify e-services structure, functionality and behavior. Each model is assigned to specific shareholders and to specific dimensions. Shareholders can visualize dependencies between models

[7] http://www.eclipse.org

which are synchronized. Thus, an evolutionary change in one model triggers changes in adjacent models.

- ***Compliance with regulations****:* Since ISE strongly relies on formal models to design e-services, it provides a paradigm which may guarantee that architected e-services comply with relevant laws, policies, and regulations. For example, ISE provides information on billing, payment, security, and confidentiality. Therefore, financial information and security policies can be easily checked.

- ***Technology standardization****:* E-services must be conform to existing standards in order to be compatible to data, applications, services, communications, integration, and security. Since the models devised within ISE and the main output of ISE is a set of technical standards, formal models of e-services are independent of specific technological choices and thus, platforms.

- ***Clear semantics****:* ISE relies on two constructions to provide semantics for e-services to an architecture: using models and using a business ontology. The set of models which is made available by ISE provides strong semantics to stakeholders due to their high level of familiarity and standardization. On the other hand, the business ontology allows constructing a common vocabulary of concepts (data) consistently throughout the organization which is shared across perspectives and dimensions and is accessible for all models to perform their functions.

5. Conclusion and Future Activities

Different types of stakeholders, such as business developers, business analysts, IT analysts, and IT developers, are involved in the development of e-services. Here, the business developers tend to work with semi-formal information on strategic issues, whereas, the IT developers base their work on formal models on implementational issues of the developing e-service. Therefore, the ISE methodology is characterized by three main features: (1) According to the different stakeholders, different perspectives on the developing e-service need to be supported by the new methodology. (2) In order to have a methodical approach for the e-service development over all perspectives, formal models are taken and assigned to each perspective. (3) The consistency of the e-service development over all perspectives is achieved by transforming the content of a model in one dimension from one perspective to the next one and vise versa.

In order to further improve the ISE methodology, the underlying model-driven framework needs to be evaluated in different contexts, e.g. entrepreneurs versus large organizations, simple versus complex (networked) e-services, and heterarchical versus hierarchical networked partner organizations. One result of the evaluation may be an extension of the ISE methodology in the future by including the innovation and runtime phases to support the full e-service lifecycle.

6. Literature

(Arsanjani, 2004) Arsanjani, A. (2004). Service-oriented modeling and architecture - How to identify, specify, and realize services for your SOA. http://www.ibm.com/developerworks/webservices/library/ws-soa-design1.

(Baida et al., 2004) Baida, Z., Gordijn, J., & Omelayenko, B. (2004). A shared Service Terminology for Online Service Provisioning. In M. Janssen, H. G. Sol, & R. W. Wagenaar (Eds.), *ICEC*, volume 60 of *ACM International Conference Proceeding Series* (pp. 1–10).: ACM.

(Bhattacharya et al., 2007) Bhattacharya, K., Gerede, C. E., Hull, R., Liu, R., & Su, J. (2007). Towards formal analysis of artifact-centric business process models. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *BPM*, volume 4714 of *Lecture Notes in Computer Science* (pp. 288–304).: Springer.

(Bieger & Rüegg-Stürm, 2002) Bieger, T. & Rüegg-Stürm, J. (2002). Net economy - die bedeutung der gestaltung von beziehungskonfigurationen. In T. Bieger, N. Bickhoff, R. Caspers, D. Knyphausen-Aufseß zu, & K. Reding (Eds.), *Zukünftige Geschäftsmodelle - Konzept und Anwendung in der Netzökonomie*. Springer.

(Buhl et al., 2008) Buhl, H. U., Heinrich, B., Henneberger, M., & Krammer, A. (2008). Service sience. *Wirtschaftsinformatik*, 1, 60–65.

(Bullinger et al., 2003) Bullinger, H.-J., Fähnrich, K.-P., & Meiren, T. (2003). Service engineering—methodical development of new service products. *International Journal of Production Economics*, 85, 275–287.

(Chen, 1977) Chen, P. P. (1977). The Entity-Relationship Model - A basis for the Enterprise View of Data. In *AFIPS National Computer Conference* (pp. 77–84).

(Chesbrough & Spohrer, 2006) Chesbrough, H. & Spohrer, J. (2006). A research manifesto for service science. *Communications of ACM*, 49, 35–40.

(Constantine, 2003) Constantine, L. L. (2003). Canonical abstract prototypes for abstract visual and interaction. In *Proceedings of DSV-IS 2003, 10th International Conference on Design, Specification and Verification of Interactive Systems. Lecture Notes in Computer Science, Vol. 2844*.

(Corsten, 1997) Corsten, H. (1997). *Dienstleistungsmanagement*. Oldenburg-Verlag.

(Daun & Klein, 2004) Daun, C. & Klein, R. (2004). *Computer Aided Service Engineering*, chapter Vorgehensweisen zur systematischen Entwicklung von Dienstleistungen im Überblick, (pp. 43–67). Springer-Verlag.

(Dorn et al., 2007) Dorn, J., Grun, C., Werthner, H., & Zapletal, M. (2007). A survey of B2B methodologies and technologies: From business models towards deployment artifacts. In *HICSS* (pp. 143).: IEEE Computer Society.

(Gruber, 1992) Gruber, T. R. (1992). *Ontolingua: A Mechanism to Support Portable Ontologies*. Technical report.

(Janiesch et al., 2008) Janiesch, C., Ruggaber, R., & Sure, Y. (2008). Eine infrastruktur für das internet der dienste. *HMD*, 261, 71–79.

(Kagermann & Österle, 2006) Kagermann, H. & Österle, H. (2006). *Geschäftsmodelle 2010*. F.A.Z.-Institut für Management-, Markt- und Medieninformationen.

(Lewis et al., 2005) Lewis, G. A., Morris, E. J., Smith, D. B., & O'Brien, L. (2005). Service-Oriented Migration and Reuse Technique (SMART). In *STEP* (pp. 222–229).: IEEE Computer Society. 0-7695-2639-X.

(Maglio et al., 2006) Maglio, P. P., Srinivasan, S., Kreulen, J., & Spohrer, J. (2006). Service systems, service scientists, ssme, and innovation. *Communications of the ACM*, 49, 81–85.

(Meiren, 2001) Meiren, T. (2001). Entwicklung von dienstleistungen unter besonderer berücksichtigung von human ressources. In H.-J. Bullinger (Ed.), *Entwicklung und Gestaltung innovativer Dienstleistungen*.

(Muehlen et al., 2007) Muehlen, M. z., Indulska, M., & Kamp, G. (2007). Business Process and Business Rule Modeling: A Representational Analysis. In K. Taveter & D. Gasevic (Eds.), *The 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2007)*, number 3: IEEE Publishers.

(Oaks et al., 2003) Oaks, P., ter Hofstede, A. H. M., & Edmond, D. (2003). Capabilities: Describing What Services Can Do. In M. E. Orlowska, S. Weerawarana, M. P. Papazoglou, & J. Yang (Eds.), *ICSOC*, volume 2910 of *Lecture Notes in Computer Science* (pp. 1–16).: Springer.

(Osterwalder, 2004) Osterwalder, A. (2004). *The Business Model Ontology - A Proposition in a Design Science Approach*. PhD thesis, Université de Lausanne.

(O'Sullivan, 2006) O'Sullivan, J. (2006). *Towards a Precise Understanding of Service Properties*. PhD thesis, Queensland University of Technology.

(Papazoglou, 2003) Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth international Conference on Web information Systems Engineering IEEE Computer Society*.

(Papazoglou et al., 2006) Papazoglou, M. P., Travero, P., Dustaar, S., Leymann, F., & Krämer, B. J. (2006). Service-oriented computing: A research roadmap. *Dagstuhl Seminar Proceedings 2006 Internationales Begegnungs- und Forschungszentrum für Informatik (IFBI)*.

(Rai & Sambamurthy, 2006) Rai, A. & Sambamurthy, V. (2006). Editorial notes - the growth of interest in service management: Opportunities for information systems scholars. *Information Systems Research 14*, 4, 327–331.

(Scheer et al., 2003) Scheer, A.-W., Grieble, O., & Klein, R. (2003). *Service Engineering*, chapter Modellbasiertes Dienstleistungsmanagement, (pp. 19–49). Springer.

(Scheer & Nuettgens, 2000) Scheer, A.-W. & Nuettgens, M. (2000). Architecture and Reference Models for Business Process Management. *Lecture Notes in Computer Science*, 1806 / 2000, 376–389.

(Sehmi & Schwegler, 2006) Sehmi, A. & Schwegler, B. (2006). Service-Oriented Modeling for Connected Systems. The Architecture Journal. Last accessed: 2008-07-21.

(Sowa & Zachman, 1992) Sowa, J. F. & Zachman, J. A. (1992). Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, 31(3), 590–616.

(Trætteberg, 2006) Trætteberg, H. (2006). A hybrid tool for user interface modelling and prototyping. In *Computer-Aided Design of User Interfaces*: Springer Science Business Media.

(Vasconcelos et al., 2003) Vasconcelos, J. B. d., Kimble, C., & Rocha, I. (2003). Organisational memory information systems an example of a group memory system for the management of group competencies. *Journal of Universal Computer Science*, 9, 1410–1427.

(Zachman, 1987) Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3), 276–292.

(Zimmermann et al., 2004) Zimmermann, O., Krogdahl, P., & Gee, C. (2004). Elements of Service-Oriented Analysis and Design - An interdisciplinary modeling approach for SOA projects. http://www-128.ibm.com/developerworks/webservices/library/ws-soad1/. Last accessed: 2008-07-21.

Authors:

Holger Kett, MBA
Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO
Competence Center Electronic Business
Nobelstraße 12, 70569 Stuttgart, Germany
holger.kett@iao.fraunhofer.de

Konrad Voigt, Dipl.-Inf.
SAP AG
Research CEC Dresden
Chemnitzer Strasse 48, 01187 Dresden, Germany
konrad.voigt@sap.com

Gregor Scheithauer, Dipl. Wirtsch.Inf. Univ.
University of Bamberg
Distributed and Mobile Systems Group
Feldkirchenstraße 21, 96045 Bamberg, Germany
gregor.scheithauer@stud.uni-bamberg.de

Dr. Jorge Cardoso
SAP AG
Research CEC Dresden
Chemnitzer Strasse 48, 01187 Dresden, Germany
jorge.cardoso@sap.com