Business Modeling for Service Descriptions: A Meta Model and a UML Profile

Gregor Scheithauer[1,2] **Guido Wirtz**[2]

[1] Siemens Corporate Technology,
Information & Communication,
Otto-Hahn-Ring 6, 81379 Munich, Germany,
Email: gregor.scheithauer.ext@siemens.com

[2] University of Bamberg
Distributed Systems Group,
Feldkirchenstraße 21, 96052 Bamberg, Germany,
Email: guido.wirtz@uni-bamberg.de

Abstract

The evolution of service-oriented architectures toward market places for business services in the Internet, raises the need for rich service descriptions with respect to service proposition and service discovery. Service providers face the challenge of business-oriented development of service descriptions for there is no conceptual formalism, a wide range and overlapping IT standards, and low alignment between business and IT. This paper reports from a research project which develops a service description method that allows documenting, communicating, and reasoning about service descriptions on various levels depending on intention and abstraction. It introduces the concepts of service market places, offers a business service meta model, and shows a valid UML Profile for it. Furthermore, a case study in the IT outsourcing domain demonstrates the strengths and weaknesses of this approach.

Keywords: Service Description, Business Model, Method, UML Profile

1 Introduction

Globalization, technological change, and an increasing demand for services (Peneder et al. 2003) transform countries from industry economies toward service economies. Regarding this trend, it becomes clear that services and their development play an important role in today's and tomorrow's business. In line with this trend, service ecosystems emerge, as an evolution of service orientation (Papazoglou 2003) that takes services from merely integration purposes to the next level by making them available as tradable products on service market places (Barros & Dumas 2006), such as *StrikeIron* and *SalesForce.com.* These providers aim at trading services over the Internet between different legal bodies, compose complex services from existing ones, and build platforms for IT-supported service provisioning (Janiesch et al. 2008). This development raises the need for rich service descriptions to enable service trade.

Figure 1 depicts steps where service descriptions contribute to service trade (cf. (Kuropka et al. 2008)). By means of service proposition, service

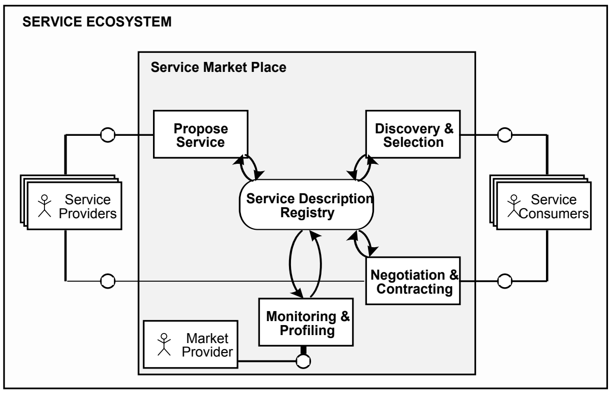

Figure 1: Trade in Service Ecosystems

providers advertise their services toward potential consumers, whereas during discovery & selection, service consumers specify their service preferences toward providers. If a service consumer selects an appropriate service, providers and consumers negotiate and finally agree on service levels (SLA) which are monitored throughout value exchange. In the event that service levels are not met, compensations have to be triggered. During service profiling, valuable information on services' performance is stored, which is gathered through value exchange and monitoring.

From the perspective of service providers, a business-oriented development of service descriptions becomes a crucial part of the service development process, which is impeded for the following reasons. Firstly, there exists no formalism for defining service descriptions on the conceptual level (Kuropka et al. 2008). Secondly, service descriptions embody divergent information and need the involvement of different subject-matter-experts. Thirdly, there do exist ample technical specifications how to describe web services with overlapping domains, which employ first-order logic, predicates, and XML, such as WSDL, WSMO, and SA-WSDL. Fourthly, there is no real alignment between business and IT. These reasons indicate that the service description development process is prone to errors, slow, and irreproducible. While recent work concentrates on *business process modeling* with a focus on how to formalize the relationship between conceptual business requirements and how to implement them with service-oriented architectures (cf. (Ouyang et al. 2006)), no attempt has been made for enhancing (the process of providing) service descriptions.

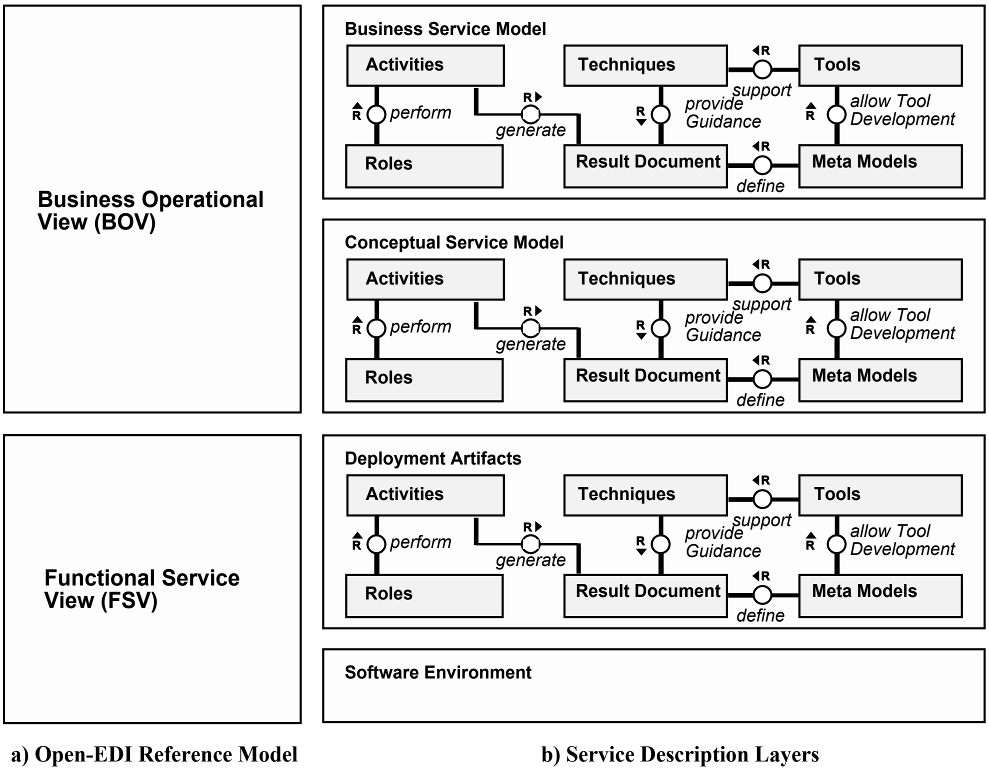

Figure 2: Open-EDI Reference Model & Service Description Layers.

Service providers would benefit from a method that allows for documenting, communicating, and reasoning about service descriptions on different levels of abstraction and, hence, support a much smoother development process. This paper tackles these problems by proposing a service description method that helps to overcome the gap between business and IT (cf. section 2). Section 3 details the Business Service Model, which comprises a knowledge structure, a notation, activities, and tools for service modeling. The approach is tested against a case study in the IT outsourcing domain in section 4. Section 5 discusses related work. Finally, section 6 concludes this work as well as offers prospects about future work.

2 Toward a Service Description Method

This section outlines the Service Description Method for Service Ecosystems (SDM4SE) (Scheithauer 2009). SDM4SE is a method to support defining service descriptions in a business-oriented fashion and to transform them into technical specifications and, hence, to ease and fasten the service description development process. Figure 2 depicts the method's cornerstones, which comprise the open-EDI reference model and method engineering, which both will be briefly explained.

2.1 Reference Model

The reference model differentiates several service description modeling phases. It is based on the open-EDI reference model (International Organization for Standardization (ISO) 2004) and work of (Dorn et al. 2007). The open-EDI reference model distinguishes between the Business Operation View (BOV) and the Functional Service View (FSV). BOV comprises business data semantics as well as business transaction rules, such as agreements and obligations between business partners. FSV, on the other hand, focuses on information technology which includes interfaces, functional capabilities, and protocols.

Dorn et al. add subtle refinements to the open-EDI reference model. They refine BOV into a business model and a process model. Business models express value exchange between different actors and business analysis. Process models represent how each actor realizes value exchanges. Likewise, they refine FSV into deployment artifacts and software environments. Deployment artifacts address implementations of business processes with technical specifications, e.g., BPEL (Alves et al. 2007). Software environments describe runtimes to execute technical artifacts. This refined model serves as a classification system for concepts and modeling notations as well as to define means to bridge gaps between different layers.

Figure 2b shows an adapted version of this reference model. Whereas Dorn et al. focuses mainly on process descriptions, this work proposes a service reference model in that the *process model* changes to *Conceptual Service Model*.

2.2 Method Engineering

Method engineering is a theory about the development of methods in the IT domain. Such methods comprise existing experience and knowledge in a domain and offer a structured approach in terms of guidance as well as documentation. Method engineering supports the formalization of this knowledge and to share it among practitioners.

According to (Gutzwiller 1994), a method embodies (1) meta models for result document specification, (2) activities to guide the modeling process, (3) role definitions, (4) tools specification, and (5) techniques (cf. figure 2b).

Result Documents embody necessary knowledge gathered throughout the engineering process. This includes, e.g., a requirement document or an architecture document. Result documents can be decomposed into sub-documents. *Meta models* define result documents by specifying a knowledge structure by means of concepts and their relationships. *Activities* comprise knowledge about which steps are to

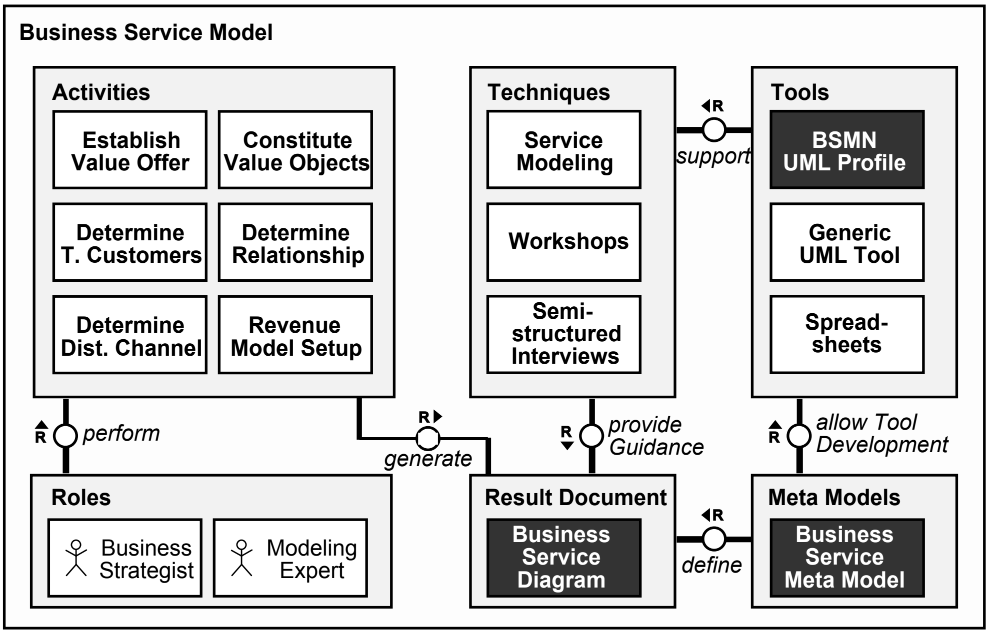

Figure 3: SDM4SE Overview for Business Service Model.

be performed in order to generate result documents. During the performance of activities semi-final result documents may be used as input. Activities may be disaggregated into sub-activities. Furthermore, sequences keep activities linked to each other. *Roles* acknowledge the fact that people with different skills are needed at certain stages in a method. A role defines a specific set of human skills which are needed for an activity. *Techniques* describe theories which are helpful to complete result documents, which include, e.g., data modeling, workflow modeling, and interviews, just to name a few. *Tools* lastly, provide support for techniques.

2.3 SDM4SE: Service Description Method for Service Ecosystems

This subsection outlines the Service Description Method for Service Ecosystems that intends to remedy the issues involved with the development of service descriptions. Figure 2b shows the combination of the reference model and method engineering. (Scheithauer et al. 2009*b*) argue that service properties in the *Business Service Model* layer own a strategic semantics and take into account services' final purpose and context. The next layer, the *Conceptual Service Model*, represents the actual modeling purpose of service descriptions. Service properties on this layer reflect a firm establishment with concrete values. The result is a value proposition toward potential customers. *Deployment Artifacts* describe technical-related specifications to implement service properties. Each layer features the artifacts from method engineering: activities, roles, techniques, result documents, tools, and meta models.

The service description layers offer an appropriate work-break-down structure in order to reduce complexity and to establish a bridge between business and IT. The definition of method engineering artifacts provides a conceptual formalism for service descriptions. Figure 2b shows that method engineering artifacts need to be defined for each layer. This is due to the fact that each layer presents a discreet phase in the service description development process. By defining the method engineering artifacts for each layer, it is possible to acknowledge different subject-matter-experts involved in describing services by codifying best-practices, to manage and generate IT specifications, and to offer cohesion between business and IT, which in turn results in less errors, fasten the development process, and makes it comprehensible. The following paragraphs briefly describe each layer.

Business Service Model: Figure 3 depicts an overview of this layer. Its purpose is to grasp services' core idea. Its meta model (BSMM), which is applied in section 3.1, holds information about target customers, distribution channels, value objects, and revenue models. The corresponding modeling notation (BSMN) is a semi-formal graphical notation based on a specific UML Profile (cf. section 3.2) which is used to document business service models. Business strategists with the capability to elicit and judging opportunities in the service market are the main actors for this layer. Typical abstract activities for business service modeling are outlined in section 4; for details see also (Scheithauer et al. 2009*b*)). This layer is discussed in more depth in section 3.

Conceptual Service Model: This layer's purpose is to transform service ideas into concrete service offers. The layer's meta model is described in previous work (Scheithauer et al. 2008, 2009*b*); it holds information about functionality, QoS, marketing, legal, as well as financial aspects. A modeling notation for this layer does not yet exists, but is planned for future work. Business analysts with knowledge about service markets and products take service ideas from the Business Service Model and use them in order to model service offers. Guiding activities for this model have already been developed (cf. (Scheithauer et al. 2009*b*)).

Deployment Artifact: This layer implements service offerings with a deployable technical language. IT architects are responsible for this layer. No specific meta model is needed for the conceptual service's meta model applies here as well. Possible technical languages (result documents) include: (1) WSDL (Chinnici et al. 2007), (2) OWL-S (Martin et al. 2004), (3) WSMO (Roman et al. 2005), (4) SA-WSDL (Farrell & Lausen 2007), and (5) WSLA (Keller & Ludwig 2003). Guiding activities are yet to be developed.

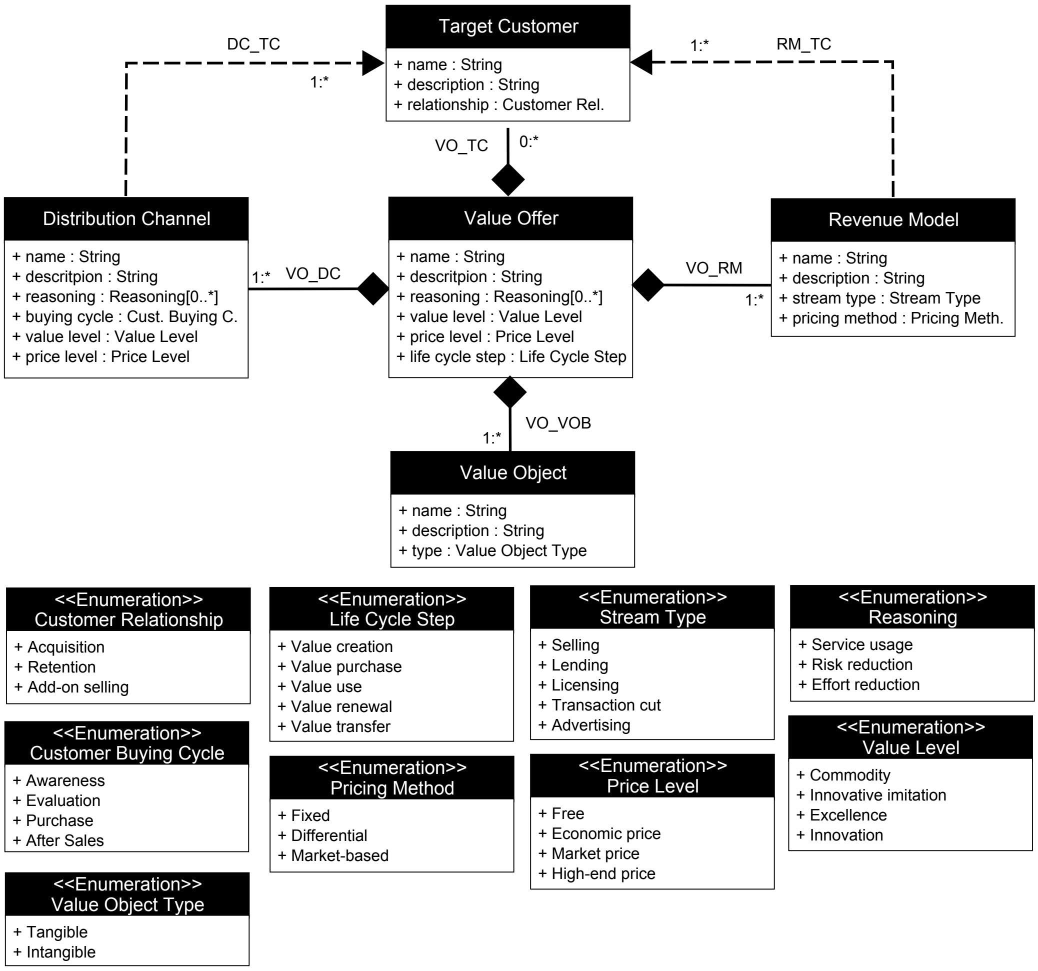

Figure 4: Business Service Meta Model (BSMM). Serves as input for UML Profile generation.

Software Environment: The most technical layer serves as a runtime for service descriptions and for deploying, discovering, and reasoning about services. Possible registries are UDDI (Organization for the Advancement of Structured Information Standards (OASIS) 2004) or WSMX (Roman et al. 2006) for semantic web services.

3 BSM: Business Service Model

Whereas the previous section provided a complete method overview, this section elaborates on the Business Service Model with its artifacts.

Figure 3 depicts a detailing view of the Business Service Model (cf. figure 2b) as well as the corresponding method engineering artifacts. In order to complete the first layer, six *activities* need to be performed: (1) establish value offer, (2) constitute value objects, (3) determine target customers, (4) determine relationship for each target customer, (5) determine distribution channel, and (6) setup appropriate revenue models. The two *roles* business strategist and modeling expert perform these six activities in collaboration. Business strategists are subject-matter-experts in a service domain and posses valuable knowledge of service markets, marketing in general, and service trends. Modeling experts, on the other hand, have the ability to elicit and to document this knowledge of business strategists. For doing so, modeling experts rely on a set of *techniques*: service modeling, workshops, and semi-structured interviews. *Tools* such as UML Profiles, UML in general, or spreadsheets support these techniques.

The black-shaded method engineering artifacts indicate the focus of this paper. (Scheithauer et al. 2009*b*) elaborate on the other artifacts. The first subsection introduces the BSMM, a *meta model* for defining service descriptions with a business model, used to grasp services' core ideas. The following subsection shows how to develop the BSMN, a modeling notation for the meta model based on a UML Profile as a *tool*. The last subsection addresses possible applications for both, the meta model and the modeling notation in the domains of service-oriented modeling and service engineering. The business service diagram is the *result document* and is part of the case study that is illustrated in figure 10.

3.1 BSMM: Business Service Meta Model

BSMM is a knowledge structure to define service descriptions on an abstract level. (Scheithauer et al. 2009*a*) discuss how this model has been developed using the work of the Business Model Ontology (BMO) (Osterwalder 2004) as well as the e^3 Value ontology (Gordijn 2002). Whereas BMO's focus lies on the internal value generation processes, the e^3 Value ontology highlights the value exchange between different actors. The resulting model selects only specific concepts that contribute to a service description, which includes: (1) value offer, (2) value object, (3) revenue model (4) distribution channel, and (5) target customer. Figure 4 shows the resulting meta model that is explained in the following paragraphs in more detail.

Value Offer is the root element and bundles the following properties: reasoning, value level, price level as well as life cycle step. Reasoning describes in which way a service is valuable for targeted customers. (Osterwalder 2004) distinguishes three elementary characteristics: value is either created by *using* a service, reducing any kind of *risk* for targeted customers, or reducing customers' *efforts*. The value level states to what extent services distinguish themselves from other companies' offers. Osterwalder provides four possible classifications: either a value offer is a *commodity*, an *innovative imitation*, an *excellence*, or an *innovation*. The price level expresses a services' qualitative pricing strategy. Services are either offered for *free*, for an *economic* (low) price, for an appropriate *market* price, or for a *high-end* price. The life cycle step formalizes when value is created during the service life cycle. Osterwalder explains the life cycle with five steps: *value creation*, *value purchase*, *value use*, *value renewal*, and *value transfer*.

Value Object is the actual value which is exchanged by companies offering services and companies consuming services. Evidence for this element is found by Osterwalder (called 'Resource') as well as by Gordijn ('Value Object'). Its properties include the value object itself and the value object type. The type attribute tells whether the value object is *tangible* or *intangible*.

Revenue Model describes the transformation of value offerings into income. It comprises the following properties: stream type and pricing method as well as a link to the customer property bundle. The stream type property formalizes how income is generated. Possible stream types include: *selling*, *lending*, *licensing*, *transaction cut*, and *advertising*. The pricing method describes in which way a price is determined. According to Osterwalder, a price is either *fixed* and is agnostic to the environment and customer characteristics, is *differential* and depends on product as well as customer characteristics, or is *market-based* in that the price is determined dynamically between provider and customer.

Distribution Channel tells how companies deliver value to targeted customers. The element bundles the properties: reasoning, value level, price level, and customer buying cycle. The properties reasoning, value level, and price level have the same semantic as in the value offer bundle, and hence, these can be setup for each channel. The customer buying cycle tells which step the channel addresses. Osterwalder proposes four steps for the buying cycle: *awareness*, *evaluation*, *purchase*, and *after sales*.

Target Customer specifies customer segments. Segments base, for example, on geographical criteria. The relationship property depicts in detail the type of connection between companies and their target customers. The relationship element classifies target customers according to their equity goals. Osterwalder offers three classes, namely *acquisition*, *retention*, and *add-on selling*.

3.2 BSMN: Business Service Modeling Notation (a UML Profile)

Following the business service meta model introduction in the previous subsection, this subsection elaborates on a corresponding notation. BSMN intends to support business strategists and modeling experts while *documenting* and *discussing* business service models, and hence to apply the Business Service Meta Model.

The Unified Modeling Language (UML) (Object Management Group (OMG) 2007) is an accepted and well-known semi-formal graphical language. Originality it aims at object-oriented design, but is not limited to it. UML Profile is part of the UML specification and offers a standard way to customize UML diagrams to cover domain-specific semantics. Standard UML and these *profiles* form the basis for a domain-specific modeling notation. This enables practitioners, who are already familiar with UML, to model specific domains. The UML Profiles were developed and used with the Eclipse UML 2 Toolset (*Eclipse Model Development Tools (MDT)* n.d.). (Giachetti et al. 2009) provide a UML Profile generation process to transform domain-specific languages into UML Profiles, which consists of three main steps:

1. Definition of Integration Meta Model – *Transformation of DSL into a meta model with its elements mapped to UML's meta model.*
2. Meta Model Comparison – *Identification of differences between meta model and UML superstructure.*
3. Integration Meta Model Transformation – *Setup of transformation rules and generation of a valid UML profile.*

3.2.1 Step 1: Definition of Integration Meta Model

The first step is to establish a meta model, namely the *Integration Meta Model (IMM)*, from the BSMM (cf. figure 4). Meta model elements need to be mapped to UML meta model elements. This step clarifies how to represent domain-specific elements with UML elements. Three main areas for mapping exist: (1) Classes & Properties, (2) Enumerations & Literals, and (3) Associations. Figure 5 exemplifies this. For example, it shows that the element Target Customer corresponds to UML Class, the element Customer Relationship is a UML Enumeration, and that RM_TC relates to a UML Association. This mapping serves as input for step 2.

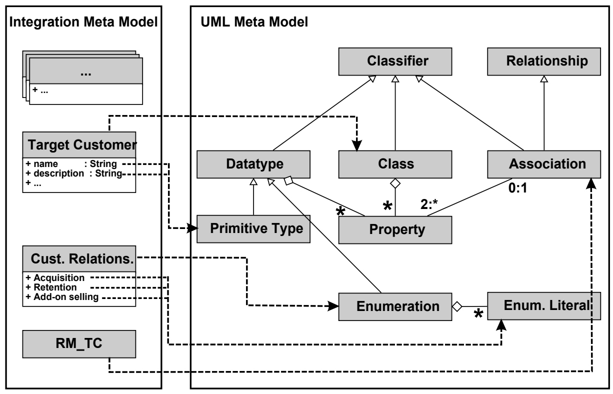

Figure 5: Excerpt of mapping between Integration Meta Model with UML Meta Model.

3.2.2 Step 2: Meta Model Comparison

With the availability of the IMM and the mapping, this step outlines differences between the IMM and the UML meta model. Each discovered discrepancy needs to be considered for the UML Profile generation. Exemplarily, table 1 shows deviances for some IMM elements that the following paragraphs explain in detail.

IMM	Differences
Target Customer	UML Class (No Diff.)
name	UML Class :: name (No Diff.)
description	New Property
relationship	New Property
...	...
Cust. Rel.	UML Enumeration (No Diff.)
Acquisition	New Enumeration Literal
Retention	New Enumeration Literal
Add-on selling	New Enumeration Literal
...	...
RM_TC	UML Association (No Diff.)
endType	Different endType: IMM = TargetCustomer; UML = relatedElement
...	...

Table 1: Extract of discovered differences between IMM & UML meta model.

Classes & Properties: In step 1 identified classes and their properties were mapped to the UML meta model. For example, the IMM's Target Customer element is mapped to UML Class. Likewise, the Target Customer's name property is mapped to the existing UML Class property *name*. However, the Target Customer's properties *description* and *relationship* may not be directly mapped and are marked with *New Property*. This Class/Property mapping is done in the same manner for the IMM's elements: Value Offer, Distribution Channel, Revenue Model, and Value Object.

Enumerations & Literals: Likewise, enumerations and their literals are mapped to the UML meta model. E.g., the IMM's Customer Relationship element is mapped to the UML Enumeration. The differences here are that the literals, *Acquisition*, *Retention*, and *Add-on selling*, are non-existent in UML's meta model, and in consequence, marked as *New Enumeration Literals*. This mapping is similar to the other enumerations, such as, Customer Buying Cycle, Value Object Type, Life Cycle Step, Pricing Method, Stream Type, Price Level, Reasoning, and Value Level.

Associations: Lastly, associations need to be mapped to the UML meta model. The IMM outlines six associations for interconnecting classes. For example, the association RM_TC tells that a Revenue Model is valid for at least one Target Customer and is mapped with the UML element Association. However, the difference between the IMM and UML is that in case of the IMM, the Revenue Models may be only associated with Target Customer, whereas the UML Association defines its endType with any related element, and thus, the endType is marked with *Different endType*. This endType difference is similar for the remaining five associations DC_TC, VO_TC, VO_VOB, VO_RM, and VO_DC.

3.2.3 Step 3: Integration Meta Model Transformation

The last step aims at codifying the discovered differences in step 2 with *transformation rules*. Eleven rules (cf. (Giachetti et al. 2009)) are the basis for the UML Profile. This subsection goes through the rules 1, 2, 6 one by one for classes, attributes & associations, and enumerations. Other rules are skipped for they are not necessary for the BSMN.

Rule 1: *(one Stereotype for each equivalent class)* As aforementioned, the IMM shows five domain-specific classes. In coherence with rule one, each class is represented with a new Stereotype. Figure 6 exemplifies that Target Customer and Value Object are UML Classes and are represented with a Stereotype in the UML Profile definition.

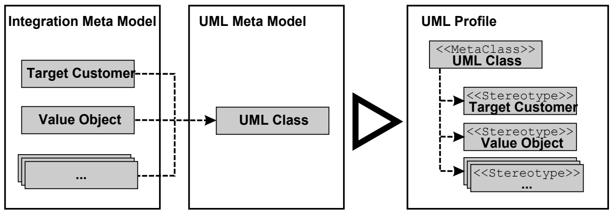

Figure 6: Application of Rule 1.

Rule 2: *(one Tagged Value for each new property)* Properties comprise *attributes* and *associations*. Tagged values consist of a name and a type. In step 2 discovered new attributes will be represented with a tagged value. For example, the class Target Customer embodies the new property *description* that is presented as a tagged value: description: String. Figure 7 shows rule 2's output. It is important to note that the *name* attribute is not represented with a tagged value for this attribute already exists in the UML Class element.

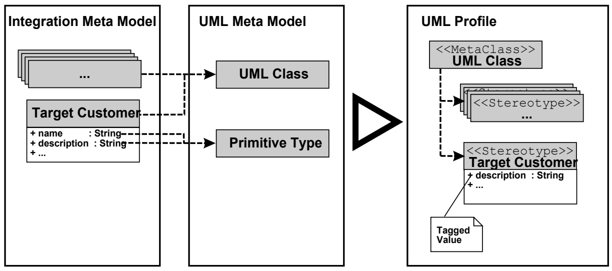

Figure 7: Application of Rule 2.

Rule 6: *(one Enumeration for each new enumeration with new literals)* Each of the IMM's enumerations with their literals are acknowledged with a UML Enumeration. Figure 8 shows that the element Customer Relationship is an Enumeration, and that its attributes Acquisition, Retention, and Add-on selling are Enumeration Literals.

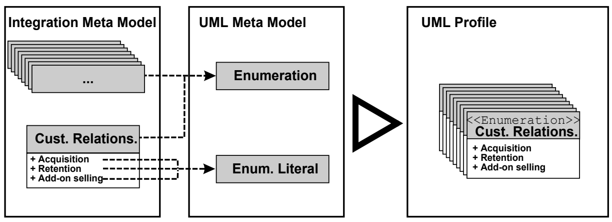

Figure 8: Application of Rule 6.

3.3 Possible Applications

Next to use BSMM and BSMN with the outlined method in section 2 and to refine service descriptions toward deployment artifacts, there exist two other promising applications for BSM: (1) modeling service-oriented architectures and (2) service engineering.

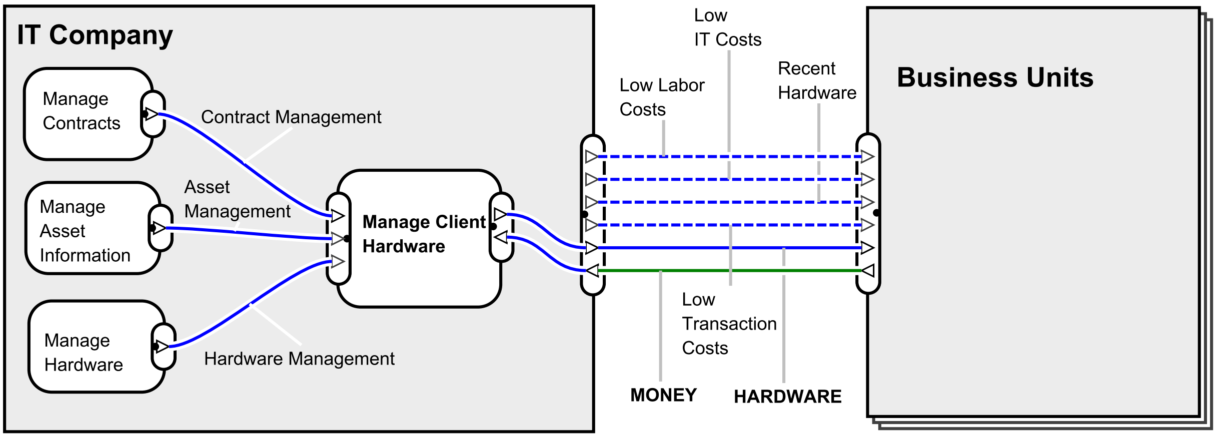

Figure 9: Manage Client Hardware Scenario (e^3 Value Diagram).

(Arsanjani et al. 2008) define Service Oriented Modeling & Architecture (SOMA) as *"... an end-to-end software development method for building SOA-based solutions"*. This method applies to establish a design and implementation for service-oriented architectures. It specifies a life-cycle comprising 21 steps, which are grouped into seven phases: (1) business modeling & transformation, (2) solution management, (3) identification, (4) specification, (5) realization, (6) implementation, (7) deployment, monitoring, and management. The authors recognize the first phase *business modeling and transformation* as an important first step that serves as the entry point for the phase *identification* & *specification*. However, Arsanjani et al. do not further describe this phase. As a suggestion, the business service meta model and the business service modeling notation may be applied to this phase.

Contrary to SOMA, (Kett et al. 2008) specify the Integrated Service Engineering (ISE) Framework for developing single business services. The ISE Framework is an orthogonal matrix and similar to the Zachman framework. The vertical axis shows four perspectives of the engineering process and is named *service perspectives*. Each perspective relates to a specific role with appropriate skills and offers different sets of tools and methods. It also implies the chronology of the framework. The horizontal axis shows five different *descriptions of a service*. Each description is valid for each perspective. Each intersection in the matrix is placeholder for a meta model, a notation, and activities, which are appropriate for the respective perspective and the modeling aspect. The Business Service Meta Model as well as the Notation fits the ISE framework's *strategic perspective* for the *service* description.

4 Case Study

After introducing BSMM and BSMN, this section outlines a case study in the IT outsourcing domain, where a real-world business service forms the basis for evaluating the Business Service Model. The following subsections depict the case study's scenario, the implementation of the scenario, and finally conclude with a discussion of the findings. It was necessary to modify the scenario and to disguise the company name for publication. The scenario's scope and complexity remain the same, nevertheless.

4.1 Scenario

IT Company is a multi-national firm that offers the business service *Manage Client Hardware*. The service's business model is to allow outsourcing of purchasing and the maintaining of computer hardware e.g., a desktop PC. Figure 9 depicts the business model with the e^3 Value Ontology. The business model comprises one actor with four value activities, a market segment, and nine value exchanges. The main actor is the IT Company itself. The company possesses three internal value activities: *manage contracts*, *manage asset information*, and *manage hardware*, with value exchanges toward the main value activity manage client hardware, which defines the external offered service. The market segment on the figure's right hand side pictures the company's target customers, i.e., its own business units. Between the actor and the market segment, the figure shows six value exchanges, and their corresponding value objects. The lowest one shows the value object *Money* that goes from the business units toward the IT Company. In this case, money is exchanged for the value object *Hardware*, which is directed from the company toward the business units. Next to these tangible values which are exchanged, four other values flow from the IT Company toward the Business Units: Low Transaction Costs, Low Labor Costs, Low IT costs, and Recent Hardware. These values are so-called second-order-values that are intangible and not actually transferred between the actors (cf. (Weigand et al. 2009)). However, business units gain these values additionally to the main value objects.

4.2 Scenario Modeling

The intention of this case study is to find out BSMN's suitability for business service modeling. It shows how to apply the UML Profile developed in section 3.2. The task includes eliciting and documenting knowledge about the *Manage Client Hardware* service for the following reasons: formalizing and communicating business ideas as well as to form a basis for service conceptualization and implementation. The scenario modeling follows the activities shown in figure 3: (1) establish value offer, (2) constitute value objects, (3) determine target customers, (4) determine relationship for each target customer, (5) determine distribution channel, and (6) setup appropriate revenue models.

Manage Client Hardware is the Value Offer. The reasoning is that it will reduce customers' effort in that the company will provide and maintain computer hardware. The value level is set to *commodity*, for the value offer is easy to imitate by competitors. The price level is situated as *economic*. The value for customers are created while *value use* during the life cycle step. The one tangible Value Object of the service is the *hardware object*. However, next to the hardware, there exist intangible value objects

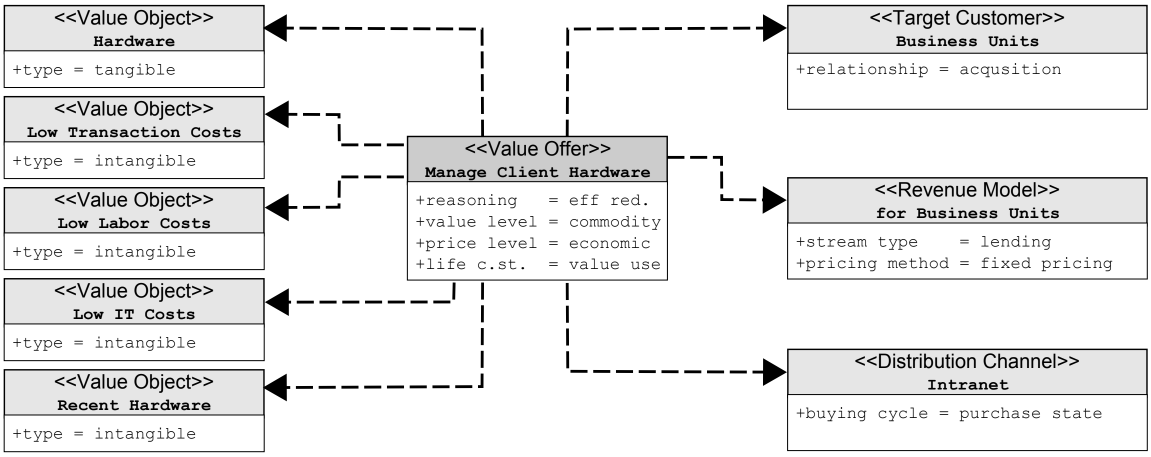

Figure 10: Business Service Diagram: Manage Client Hardware.

which also contribute to the service offering. Outsourcing hardware management to IT Company results in *lower transaction costs* for purchasing and contracting, *lower labor costs* for hardware maintenance, *lower IT costs*, and *state-of-the-art hardware*. *Business units* are the service's Target Customers. The Relationship to these customers is not yet established. Hence the relationship is marked as *acquisition*. The *Manage Client Hardware* service's Distribution Channel relies solely on the company's web online portal, which supports *purchase* state of customers' buying cycle. Likewise, the company follows one Revenue Model, which settles for a *fixed price* as pricing method and *lending* for the stream type.

Figure 10 depicts he final UML diagram (*resulting document*, cf. section 3) with the aforementioned Business Service Meta Model elements, which figure 4 prescribes. Additionally, listing 1 shows the corresponding XML fragment for the UML diagram that can be used for persistence and further processing such as model transformation.

Listing 1: Corresponding XML Code

```
<?xml version="1.0" encoding="UTF-8"?>
<bm:BusinessServiceModel xmlns:bm="http:
    //www.itcompany.com">
  <ValueOffer Name="Manage Client
      Hardware" Reasoning="Effort
      Reduction" ValueLevel="Commodity"
      PriceLevel="Economic"
      LifeCycleStep="Value Use" VOf_DC="
      DC1" VOf_RM="RM1" VOf_TC="TC1"
      VOf_VOb="VO1_VO2_VO3_VO4_VO5"/>
  <TargetCustomer TargetCustomerID="TC1"
      Name="Business Units">
    <consistsOf CustomerEquity="
        Acquisition"/>
  </TargetCustomer>
  <RevenueModel RevenueModelID="RM1"
      Name="for Business Units"
      StreamType="Lending" PricingMethod
      ="Fixed Price"/>
  <DistributionChannel
      DistributionChannelID="DC1" Name="
      Intranet" Reasoning="Effort
      Reduction" CustomerBuyingCycle="
      Purchase" ValueLevel="Commodity"
      PriceLevel="Economic"/>
  <ValueObject ValueObjectID="VO1" Name=
      "Hardware" Type="Tangible"/>
  <ValueObject ValueObjectID="VO2" Name=
      "Low Transaction Costs" Type="
      Intangible"/>
  <ValueObject ValueObjectID="VO3" Name=
      "Low Labor Costs" Type="Intangible
      "/>
  <ValueObject ValueObjectID="VO4" Name=
      "Low IT Costs" Type="Intangible"/>
  <ValueObject ValueObjectID="VO5" Name=
      "Recent Hardware" Type="Intangible
      "/>
</bm:BusinessServiceModel>
```

4.3 Findings

The case study's intention was to figure out whether the Business Service Meta Model and Modeling Notation supports the *documentation*, the *communication*, and the *reasoning* of service descriptions on a strategic level.

The case study shows that the proposed approach is appropriate for *documenting* business service models. In particular, the developed UML Profile (BSMN) guarantees a full documentation of services' core ideas. However, business strategists were not familiar with UML or UML Profiles. This experience made it necessary to involve modeling experts who are familiar with UML in order to document business service models. One idea is to hide the notations from business strategists and rather use semi-structured interviews to elicit necessary information and use the answers to these questions to build business service diagram.

Furthermore, the case study proves that business strategists were able to *communicate* the service's main idea with involved business strategists on the basis of the business service diagram (cf. figure 10). Moreover, business strategists were in the position to *discuss* and rethink the business service model and hence to improve it.

The fact that the resulting diagrams use XML as a serialization (cf. listing 1) allows further processing of services' information, such as model transformations.

Further case studies need to detect whether the business service diagram is an appropriate starting position for the Conceptual Service Model.

5 Related Work

(Baida et al. 2003) argue that eCommerce is still mainly characterized by the relatively straightforward trading of commodity goods. Current challenges are advanced business scenarios, such as collaborative design over the Internet of sophisticated goods and services. Their work elaborates on further challenges in order to achieve collaborative eCommerce concerned with real-world services. Similar to the Business Service Model, Baida et al. focus on service trade and propose a knowledge structure in form of a service ontology. The differences lie in that Baida et al.'s service ontology rather targets the Conceptual Service Model than the Business Service Model, and that they neither propose a modeling notation nor a procedure model.

(Weigand et al. 2009) introduce a unified view on services by means of a service model and a modeling method targeted to the design and analysis of services. They argue that there exists a business view on services, such as in the approaches from Gordijn and Osterwalder. Additionally, they propose a 'software view' on services, namely the Service-Oriented Modeling & Architecture (SOMA). The authors find that a gap exists between these two views and that a service model closes this gap. Following that, Weigand et al. discuss business modeling with the REA and e^3 Value Ontology, and Spohrer's service systems theory. The service model comprises of a service ontology, a service classification, and a service layer architecture. Likewise to the Business Service Model, Weigand et al. propose a knowledge structure in form of an ontology and utilize the e^3 Value ontology as a notation. Their approach differs from the Business Service Model in that it aims at service identification and classifies in the Conceptual Service Model.

(Terlouw 2008) finds the UDDI specification too technology-driven for specifying services and hence believes that it contradicts SOA promises of increased flexibility of service reuse and business-IT alignment. She finds the business component framework and particularly the task specification more suitable for doing so. Terlouw claims that for business process execution, suitable services need to be identified as well as to specified. Service registries store these specifications for identification. In consequence, she proposes the Enterprise Ontology and the business component specification for business task specification. Terlouw offers a knowledge structure on the basis of the Enterprise Ontology. Her solution focuses on services as business tasks and relates to the Conceptual Service Model and offers neither a modeling notation nor a procedure model.

(Dumas et al. 2001) identify the need for a semantic service description framework because of the Internet's global and inexpensive connectivity. Such a description aims at advertising, locating, analyzing, and comparison of services. The authors' intention is to define requirements for future service descriptions. They propose the following service characteristics: provider, availability (time & spatial), channel, pricing, payment, security, quality of service, and reputation. The authors find the UDDI's TModel appropriate as an underlying model for these service characters. Likewise to the Business Service Model, Dumas et al. aim at service proposition. However, they mainly propose requirements for service propositions that relate to the Conceptual Service Model. They neither give information about a modeling notation nor a procedure model.

6 Conclusion and Future Work

With the evolution of service-oriented architectures toward service market places in the Internet, services and their description become even more important. Service descriptions are an elementary part of service trade for they contribute to service proposition, service discovery & selection, negotiation & contracting as well as monitoring.

From the perspective of service providers, the business-oriented development of such descriptions is a crucial part of the service development process. However, until recently no conceptual formalisms do exist for doing so (Kuropka et al. 2008). There are, however, many technical specifications out there, e.g. WSDL, SA-WSDL, and WSLA, using sometimes completely different notations but describing partly overlapping aspects. Even more important, it is necessary to involve *distinct* subject-matter-experts in the service development process. Using description formalisms primarily tailored to technical aspects will hardly be successful under these circumstances. Last, but by no means least, it is essential to align business with information technology.

Hence, service providers would benefit from a method that allows for the crucial tasks of *documenting*, *communicating*, and *reasoning about* service descriptions on the different levels of abstraction and domains of expertise that are needed during the process. An understandable description of all levels would avoid mistakes and fasten the service description process materially.

Against this background, this paper outlines a service description method that combines the Open-EDI Reference Model with method engineering, which offers a work-break-down structure in order to reduce complexity and to align business and IT. Particularly, the Business Service Model, as one part of the method, is further detailed. The paper proposes a meta model for business-oriented service descriptions and develops a corresponding modeling notation on the basis of UML Profile, which supports all three steps, i.e., documenting, communication, and reasoning about descriptions on a strategic level. The result document is a valuable input for service descriptions on a conceptual level. Two possible fields of applications have been outlined: service-oriented modeling and service engineering. The proposed approach was tested in a case study in the IT outsourcing domain to show the applicability of the Business Service Model.

The case study shows that the proposed solution is appropriate for all three tasks on a business-oriented level. The usage of UML made it necessary to involve modeling experts. Nevertheless, the modeling notation turns out to be practical for communicating and reasoning about service descriptions.

Future work includes integrating the meta model and the modeling notation with related approaches in the service description domain. Furthermore, the Conceptual Service Model needs to be detailed and integrated with the Business Service Model. This will be addressed in the next steps of the Theseus/TEXO research project (Janiesch et al. 2008).

Acknowledgements

This project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The responsibility for the content of this publication lies with the authors.

References

Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guzar, A., Kartha, N. & Liu, C. K. (2007), 'Specification: Business Process Execution Language for Web Services version 2.0'.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Gariapathy, S. & Holley, K. (2008), 'Soma: a method for developing service-oriented solutions', *IBM Syst. J.* *47*(3), 377–396.

Baida, Z., Akkermans, H. & Gordijn, J. (2003), Serviguration: towards online configurability of real-world services, *in* N. M. Sadeh, M. J. Dively, R. J. Kauffman, Y. Labrou, O. Shehory, R. Telang & L. F. Cranor, eds, 'ICEC', Vol. 50 of *ACM International Conference Proceeding Series*, ACM, pp. 111–118.

Barros, A. P. & Dumas, M. (2006), 'The Rise of Web Service Ecosystems', *IT Professional* *8*(5), 31–37.

Chinnici, R., Moreau, J.-J., Ryman, A. & Weerawarana, S. (2007), 'Specification: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language', W3C Recommendation.

Dorn, J., Grun, C., Werthner, H. & Zapletal, M. (2007), A Survey of B2B Methodologies and Technologies: From Business Models towards Deployment Artifacts, *in* 'HICSS', IEEE Computer Society, p. 143.

Dumas, M., O'Sullivan, J., Heravizadeh, M., Edmond, D. & ter Hofstede, A. H. M. (2001), Towards A Semantic Framework for Service Description, *in* 'DS-9', pp. 277–291.

Eclipse Model Development Tools (MDT) (n.d.), http://www.eclipse.org/modeling/mdt/.

Farrell, J. & Lausen, H. (2007), 'Specification: Semantic Annotations for WSDL and XML Schema (SA-WSDL)', http://www.w3.org/TR/sawsdl/.

Giachetti, G., Marín, B. & Pastor, O. (2009), Using UML as a Domain-Specific Modeling Language: A Proposal for Automatic Generation of UML Profiles, *in* (van Eck et al. 2009), pp. 110–124.

Gordijn, J. (2002), E^3-value in a Nutshell, Technical report, HEC University Lausanne, Lausanne.

Gutzwiller, T. (1994), *Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen*, Physica-Verlag, Heidelberg.

International Organization for Standardization (ISO) (2004), 'Open-edi reference model, iso standard 14662, second edition'.

Janiesch, C., Ruggaber, R. & Sure, Y. (2008), 'Eine Infrastruktur für das Internet der Dienste', HMD - Praxis der Wirtschaftsinformatik (45:261), 2008, pp. 71-79.

Keller, A. & Ludwig, H. (2003), 'The WSLA framework: Specifying and monitoring service level agreements for web services', *J. Network Syst. Manage 11*(1).

Kett, H., Voigt, K., Scheithauer, G. & Cardoso, J. (2008), Service Engineering in Business Ecosystems, *in* 'Proceedings of the XVIII. International RESER Conference', Stuttgart, Germany.

Kuropka, D., Troeger, P., Staab, S. & Weske, M., eds (2008), *Semantic Service Provisioning*, Springer Berlin Heidelberg. ISBN 978-3-540-78616-0.

Martin, D. L., Paolucci, M., McIlraith, S. A., Burstein, M. H., McDermott, D. V., McGuinness, D. L., Parsia, B., Payne, T. R., Sabou, M., Solanki, M., Srinivasan, N. & Sycara, K. P. (2004), Bringing Semantics to Web Services: The OWL-S Approach, *in* J. Cardoso & A. P. Sheth, eds, 'SWSWPC', Vol. 3387 of *Lecture Notes in Computer Science*, Springer, pp. 26–42.

Object Management Group (OMG) (2007), 'Specification: Unified Modeling Lanugage (UML) version 2.0', http://www.omg.org/docs/formal/05-07-04.pdf.

Organization for the Advancement of Structured Information Standards (OASIS) (2004), 'Specification: Universal Description Discovery and Integration (UDDI) Version 3.0'

Osterwalder, A. (2004), The Business Model Ontology: A Proposition in a Design Science Approach, PhD thesis, Universite de Lausanne Ecole des Hautes Etudes Commerciales.

Ouyang, C., van der Aalst, Dumas, M., ter Hofstede & M., A. H. (2006), 'From business process models to process-oriented software systems: The BPMN to BPEL way', ePrint: http://eprints.qut.edu.au/archive/00005266/.

Papazoglou, M. P. (2003), Service-Oriented Computing: Concepts, Characteristics and Directions, *in* 'WISE', IEEE Computer Society, pp. 3–12.

Peneder, M., Kaniovski, S. & Dachs, B. (2003), 'What Follows Tertiarisation? Structural Change and the Role of Knowledge-based Services', *The Service Industries Journal 23 Issue 2*(146), 47–66.

Roman, D., de Bruijn, J., Mocan, A., Lausen, H., Domingue, J., Bussler, C. & Fensel, D. (2006), WWW: WSMO, WSML, and WSMX in a Nutshell, *in* 'ASWC', pp. 516–522.

Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C. & Fensel, D. (2005), 'Web Service Modeling Ontology', *Applied Ontology 1*(1), 77–106.

Scheithauer, G. (2009), Business Service Description Methodology for Service Ecosystems, *in* 'Proceedings of the 16th CAiSE-DC'09 Amsterdam, The Netherlands, June 9-10, 2009', Vol. 479 of *ceur-ws*.

Scheithauer, G., Augustin, S. & Wirtz, G. (2008), Describing Services for Service Ecosystems, *in* G. Feuerlicht & W. Lamersdorf, eds, 'ICSOC Workshops', Vol. 5472 of *Lecture Notes in Computer Science*, Springer, Sidney, Australia, pp. 242–255.

Scheithauer, G., Augustin, S. & Wirtz, G. (2009*a*), Business Modeling for Service Engineering: Toward an Integrated Procedure Model, *in* 'Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), Boston, Massachusetts, USA, July 1-3, 2009', Boston, MA, USA, pp. 322–327.

Scheithauer, G., Augustin, S. & Wirtz, G. (2009*b*), Service Value Properties for Service Ecosystems: A Reference Model and a Modeling Guideline, *in* J. Barjis, J. Kinghorn, S. Ramaswamy, E. Dubois & P. Johannesson, eds, 'Proceedings of the EOMAS Workshop', Vol. Vol-458 of *CEUR-WS*, Amsterdam, Netherlands.

Terlouw, L. (2008), Towards a Business-Oriented Specification for Services, *in* 'Advances in Enterprise Engineering I, 4th International Workshop CIAO! and 4th International Workshop EOMAS', Vol. 10 of *Lecture Notes in Business Information Processing*, Springer, Montpellier, France, pp. 122–136.

van Eck, P., Gordijn, J. & Wieringa, R., eds (2009), *Advanced Information Systems Engineering, 21st International Conference, CAiSE 2009, Amsterdam, The Netherlands, June 8-12, 2009. Proceedings*, Vol. 5565 of *Lecture Notes in Computer Science*, Springer.

Weigand, H., Johannesson, P., Andersson, B. & Bergholtz, M. (2009), Value-based service modeling and design: Toward a unified view of services, *in* (van Eck et al. 2009), pp. 410–424.