Integrated Service Engineering workbench: service engineering for digital ecosystems

Gregor Scheithauer*

Business Engineering,
OPITZ CONSULTING München GmbH,
Weltenburger Str. 4,
81677 Munich, Germany
E-mail: scheithauer@acm.org
*Corresponding author

Konrad Voigt and Matthias Winkler

CEC Dresden, SAP Research,
Chemnitzer Str. 48,
01187 Dresden, Germany
E-mail: konrad.voigt@sap.com
E-mail: matthias.winkler@sap.com

Veli Bicer

FZI Forschungszentrum Informatik,
Haid-und-Neu-Straße 10-14,
76131 Karlsruhe, Germany
E-mail: bicer@fzi.de

Anja Strunk

Computer Networks, TU Dresden,
Nöthnitzer Str. 46,
01187 Dresden, Germany
E-mail: anja.strunk@tu-dresden.de

Abstract: The evolution of service-oriented architectures towards digital ecosystems comprehends a number of challenges. One challenge is to develop services in a dynamic environment with high uncertainties and in collaboration with other companies. Another challenge lies in how to describe services sufficiently for trading them in digital ecosystems. This paper explores requirements of service engineering and description, and presents the ISE Framework and Workbench with its architecture and its functionality. Furthermore, this paper introduces a novel language for describing business services along with SLA management. This work concludes with a discussion about lessons learned during the development of the ISE Workbench.

Keywords: service engineering; business modelling; BPM; MDA; model-driven architecture; digital ecosystems.

Reference to this paper should be made as follows: Scheithauer, G., Voigt, K., Winkler, M., Bicer, V. and Strunk, A. (2011) 'Integrated Service Engineering workbench: service engineering for digital ecosystems', *Int. J. Electronic Business*, Vol. 9, Nos. 5/6, pp.392–413.

Biographical notes: Gregor Scheithauer works as a Business Engineering Consultant at OPITZ CONSULTING München GmbH in Germany. He holds a Diploma in Business Information Systems from University of Bamberg, Germany, where he also pursues his PhD thesis about conceptual service modelling. His research interests comprise service descriptions, business process modelling and automation as well as service engineering.

Konrad Voigt works at SAP Research as PhD candidate. He works in the public German project Texo concerned with a marketplace for services. In his PhD Thesis he is investigating structural graph-based matching techniques for data integration. Prior to joining SAP, he received his Diploma Degree from the HU Berlin, where he focused on metamodelling.

Matthias Winkler received his PhD in Computer Science at the Technical University of Dresden in collaboration with SAP Research. He works as a researcher at SAP Research Center Dresden. His research focuses on service description and SLA management as well as collaboration of business partners in the Internet of Services.

Veli Bicer is a Research Scientist and PhD candidate at Research Center for Computer Science (FZI – Forschungszentrum Informatik) at Karlsruhe Institute of Technology. He obtained his BSc and MSc Degrees in Computer Engineering from the Middle East Technical University in Ankara, Turkey. His research interests includes semantic search, and machine learning theory on information retrieval.

Anja Strunk is a Research Scientist at the Dresden University of Technology. She studied computer science at the Dresden University of Technology and received her PhD in the area of the internet of service in 2010. Her research interests include quality of service, adaptation and optimisation.

1 Introduction

The alignment between business and IT is crucial in today's business because organisations need to adapt quickly and frequently as a consequence of Globalisation and technological change (cf. Peneder et al., 2003), which provoke highly dynamic environments as well as high uncertainties according to Papazoglou et al. (2008). Furthermore, next to international production and trade of physical goods, digital ecosystems foster trading business services over the internet (cf. Kett et al., 2008).

These developments have an impact on how services, both business and technical, need to be developed (cf. Papazoglou et al., 2008) and described in order to be traded in digital ecosystems. Kett et al. (2008) show that existing software service development processes comprise shortcomings in the context of digital ecosystems. These shortcomings include the overall complexity of realising tradable business services, the multidimensional aspects of business services, the diverse parties involved in the engineering processes, and the ability to react to frequent changes.

The Integrated Service Engineering (ISE) Framework (cf. Kett et al., 2008; Scheithauer et al., 2009a, 2010; Kett et al., 2009) addresses these issues as it embraces the relationship between tradable business services and their realisation with information technology. The general idea is to define business logic in a technology-free fashion and to transform it into technical blueprints, neglecting traditional expensive and interminable software engineering projects by applying a firm method that comprises on the one hand business models (cf. Osterwalder, 2004), service marketing (cf. Lusch and Vargo, 2006) and enterprise architecture (cf. Zachman, 1987), and on the other hand, Model-Driven Architecture (MDA) (cf. MDA, 2003) and actual tooling (cf. Scheithauer et al., 2009b). Furthermore, this approach provides a novel language for describing services beyond technical matters in that it addresses business and operational aspects including pricing as well as provides Service Level Agreements (SLAs) for composed services from different providers.

This work's contribution is an enhancement of the ISE Framework in that it introduces an implementation of the framework and its concepts for engineering and trading business services. The remainder of this paper is structured as follows: section 2 discusses specific requirements for digital ecosystems in terms of service engineering and service trade and provides a running example. Section 3 introduces the ISE approach with its framework and workbench, and model transformations. Section 4 concentrates on how to describe and contract business services, engineered with ISE in digital ecosystems. While Section 5 discusses lessons learned during a three year research project, Section 6 offers work that is related to the ISE Workbench. Section 7 concludes this work.

2 Digital ecosystems' requirements

The service idea has evolved as a foundational concept for the exchange of value coproduced as a result of a coordinated set of actions. Lusch and Vargo (2006) define services as

> "*... the application of specialised competences (knowledge and skills), through deeds, processes, and performances for the benefit of another entity or the entity itself.*"

The coproduction of service value is the key for the success of a service as it takes place as interventions of different entities comprised of people, technology, organisations, or possibly other services.

Recently, the internet and other similar networks provide a new perspective by changing the way how the coproduction takes place. The term digital ecosystem describes such environments where a logical collection of services, service providers, consumers, and intermediaries come together in order to trade services (cf. Barros and Dumas, 2006). They are supported with an infrastructure that consists of architectural standards, technology, tools and applications to easily realise and to execute services by the interaction of several parties. Service Oriented Architecture (SOA) and related service technologies such as BPEL (cf. Alves et al., 2007) and WSDL (cf. Ryman et al., 2007) provide a basis of such an infrastructure in a digital ecosystem. However, according to Kett et al., 2008, digital ecosystems comprise a number of challenges. Two specific challenges lie in how to engineer business services and how to advertise these Services for service trade in digital ecosystems that the next two subsections elaborate on.

2.1 Service engineering for digital ecosystems

In addition to providing a technical infrastructure, engineering a business service for such environments is a complex undertaking. First of all, a service in such a setting is not merely a single computational entity, but rather a multi-dimensional asset that spans different concerns, such as service processes, data, descriptions, interfaces, agreements, or legal and organisational aspects. This requires the design of a service to deal with all those concerns whereas they are to be combined into one final product prior to deployment. Therefore, each of those concerns requires a separate modelling support for its own purposes, while the coordination among the models has utmost impact on the final result.

Furthermore, engineering a service can be considered as the transformation of a business idea into a service. In a digital ecosystem setting, such a transformation involves the participation of several parties, capturing the business requirements of each party and reflecting them into the design at various stages. Considering the dynamic nature of the ecosystem, such a design should also be adaptable, prone to the changes, and supporting fast development to easily convert the idea into a service. Unlike the classical development methodologies that employ low-level techniques, a service engineering method for digital ecosystems should support a simplified and layered development in order to deal with the inherent complexity of the design.

Section 3 introduces the ISE Framework and Workbench, which address these business service engineering challenges.

2.2 Service trade in digital ecosystems

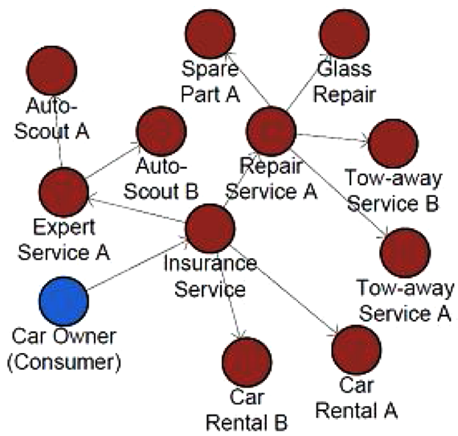
Next to engineering business services in terms of internal behaviour, involved roles, and information, it is crucial to market and to place these services in digital ecosystems for potential consumers. Current Web service technologies, e.g., WSDL (cf. Ryman et al., 2007) and UDDI (cf. OASIS, 2004), address mainly technical and functional service properties. While such descriptions are sufficient for technical service-oriented scenarios, service trade in digital ecosystems requires a holistic description from a business perspective, including details about legal aspects and financial terms (cf. Barros and Dumas, 2006; Scheithauer and Winkler, 2008; Scheithauer et al., 2008).

Furthermore, digital ecosystems allow service providers to compose novel services with existing services from different providers. Such service aggregators are faced with the challenge to derive a valid set of service levels from original services for composite services and to propose it to potential consumers. Section 4 presents the Universal Service Description Language (USDL) as a rich service description for digital ecosystems as well as SLAs for composite business services.

2.3 Motivating example

This subsection presents a car insurance scenario to illustrate collaboration within a digital ecosystem as depicted in Figure 1. The consumer (e.g., car owner) receives a car insurance service from an insurance company which is positioned in a network of many service providers in order to offer a complete car insurance (i.e., repair of the car, rental etc.). There exist a number of specialised service providers that are required to perform special activities for the service fulfilment. For example, a repair service supplies a car repair, which in turn, depends on a tow away service or other repair services (e.g., glass repair). There are also information services such as expert service, or auto-scout that provides an expertise report of possible damage requested by the insurance company.

Figure 1 An example service network for car insurance scenario (see online version for colours)



We consider that, for a car insurance service, the process starts with the design of the service (based on a service idea) that includes the specification of a business service model (cf. Scheithauer and Wirtz, 2010). From the actual service provider's perspective (insurance company in this case), the service requires the main steps of the service fulfilment to be provided to the consumer whenever a need to a service arises. The design step is followed by an agreement where particular Service-Level Agreements (SLAs) are reached either by manual contracting or automatic means. We assume that both design and agreement occur long before service initialisation and execution during which consumer and provider interact via a user interface (e.g., service mashup). In the remainder of the paper, we further elaborate on this scenario in terms of engineering and describing such a service in the context of the ISE Workbench.

3 Integrated Service Engineering

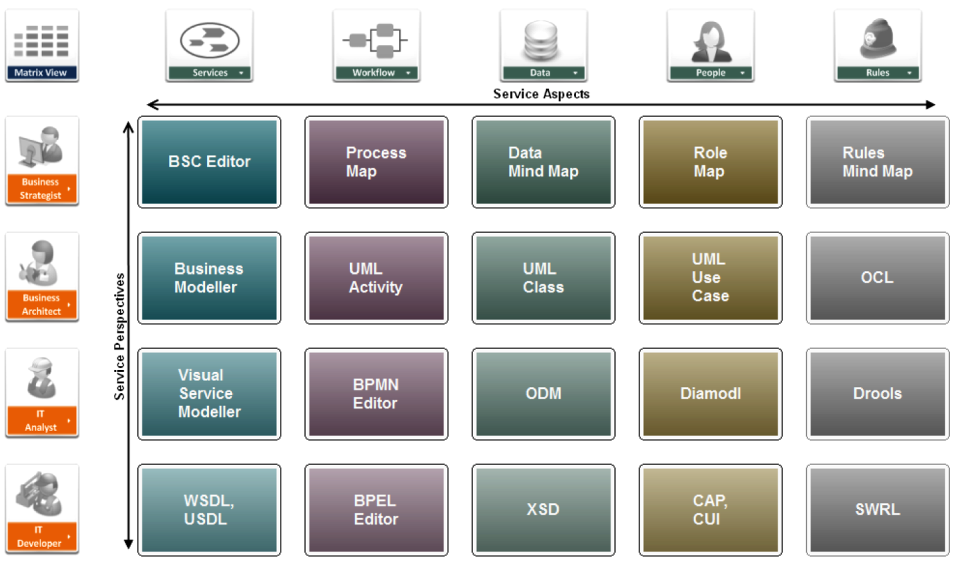
The ISE Framework was developed in order to cope with the complexity of service engineering in digital ecosystems. The ISE Workbench, on the other hand, is an implementation of the framework. It is a role-based tool consisting of 20 editors, a SLA Management component, and infrastructure components.

The following three subsections introduce briefly the ISE Framework, give an overview on the architecture of the ISE Workbench and present the technology and underlying dependencies. Furthermore the last subsection describes the synchronisation of the 20 editors models using model to model transformations.

3.1 ISE framework

The ISE Framework (cf. Kett et al., 2008, 2009; Scheithauer et al., 2009a, 2010) supports service engineering in terms of planning, designing and implementing services, which are traded over the internet, in addressing stakeholders from business and IT, and acknowledges different service aspects. Figure 2 shows that ISE relies on the Zachman framework and follows a divide and conquer approach. ISE is not only limited to computing services. Rather, it targets business services, such as insurance and financial services, civil services, marketing services, and telecommunication services.

Figure 2 ISE framework (see online version for colours)



The vertical axis in Figure 2 (service perspectives) represents four perspectives of the engineering process. 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. Additionally, the perspectives are linked to phases of the service engineering process.

The horizontal axis (aspects) in Figure 2 allows for five different descriptions of a service. Each description is valid for each perspective. Any intersection in the matrix is a placeholder for an appropriate model, a notation, and a modelling technique.

The ISE Framework addresses the involved complexity of service engineering by offering a work-break-down-structure of 20 different models, where each is concerned with a single aspect of the engineering process. It addresses five different aspects of business services from a business level toward a technical level. Furthermore, the framework acknowledges the different parties involved in service engineering, which include domain experts, business strategists and analysts, and IT architects and programmers. Also, ISE integrates the different models in order to cope with the overall complexity and employs transformation rules to address a fast development process as well as a means for reacting to frequent changes.

The development of ISE started in April 2007 as part of the Theseus/TEXO research project. Theseus/TEXO (2011) addresses necessary technology and concepts for trading services over the Internet, such as business models, legal aspects, governance, innovation, service discovery, and service runtimes as well as service engineering.

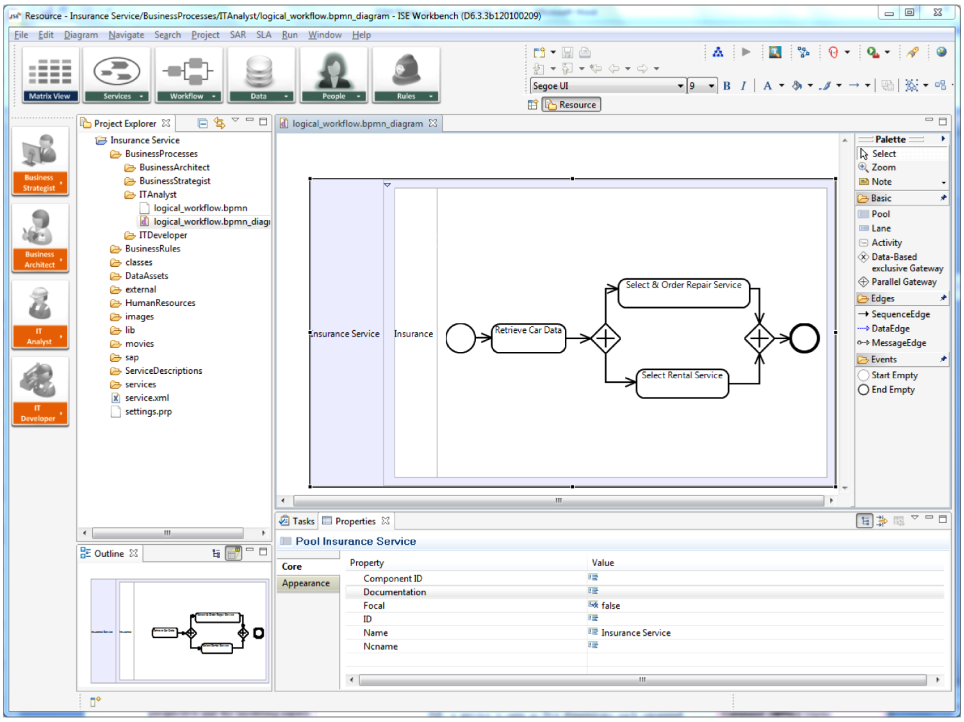
Kett et al. (2008) outline the scope as well as the intention of an integrated method for service engineering. The authors include an overview of available methods for technical service development. Additionally, they show ISE's relation to the Zachman (1987) framework as well as first ideas for the ISE Framework.

Scheithauer et al. (2009a) focus on the business requirement layers of ISE in terms of meta models for the strategic and the conceptual perspective of ISE. These meta models depict necessary knowledge in order to engineer a business service, and hence what information needs to be modelled during the first two perspectives. Furthermore, a procedure model with 11 abstract activities provides guidance for eliciting this information.

In Kett et al. (2009) describe the actual development of ISE. The authors show its relation to the business engineering discipline and its focus on service marketing aspects. In addition, the authors outline involved roles in the engineering process and introduce integration rules. Furthermore, Kett et al. specify key questions in order to support the information elicitation process for ISE's strategic and conceptual perspective, and hence, improve the framework's applicability.

In Scheithauer et al. (2010) perform a case study in the IT outsourcing domain, where a service for hardware outsourcing was developed. The case study concentrates on the strategic and the conceptual perspective of ISE. Although meta models existed for these two perspectives, no modelling notations were available. Scheithauer et al. use UML and UML profiles in order to build ISE specific modelling notations.

Figure 3 ISE workbench screenshot with BPMN editor showing the car insurance service process (see online version for colours)



3.2 ISE workbench

The ISE Workbench (cf. Scheithauer et al., 2009b) implements the ISE Framework and its 20 models, dedicated to a specific intersection of the ISE Framework. As imposed by ISE, a service is seen as five aspects each separated into four layers of abstraction. The resulting matrix consists of 20 intersections where an editor within the ISE Workbench represents each intersection.

This matrix serves at the same time as entry points to the ISE Workbench as depicted in Figure 3, where the matrix is in the upper left.

In order to fulfil the task of service engineering the workbench has to provide an underlying structure for service models as well as the integration of those. Since the context of digital ecosystems imposes constant change the workbench should be extensible as well as flexible:

- *Extensible*: The addition of new components, e.g., editors, should be possible.
- *Flexible*: The components should be exchangeable, i.e., removal and addition without effect on other components.
- *Model-based*: It should support a common formalism for model definition, thus enabling synchronisation.

Grounded on these requirements we chose the Eclipse Framework as a base for implementing the ISE Workbench. Eclipse is a Java implementation of an integrated development environment with support for graphical and textual editors. The framework for integrating these editors is the pluginbased approach by Eclipse which is defined in the OSGI (OSGi Alliance, 2007) standard. OSGI is a well-defined protocol on formulating dependencies between components as well as separating them. This choice fulfils the two requirements of extensibility and flexibility.

In order to support model-based model integration and synchronisation a framework within Eclipse can be reused. The Eclipse Modelling Framework (EMF) (cf. Steinberg et al., 2008) allows managing models within the common formalism Ecore. Ecore is closely related to MOF (cf. OMG, 2006), the formalism recommended within MDA, which makes it a perfect choice. Eventually, we use EMF to establish common ground for each editor to exchange information in form of models. Therefore, EMF provides a format and surrounding framework for integration and synchronisation.

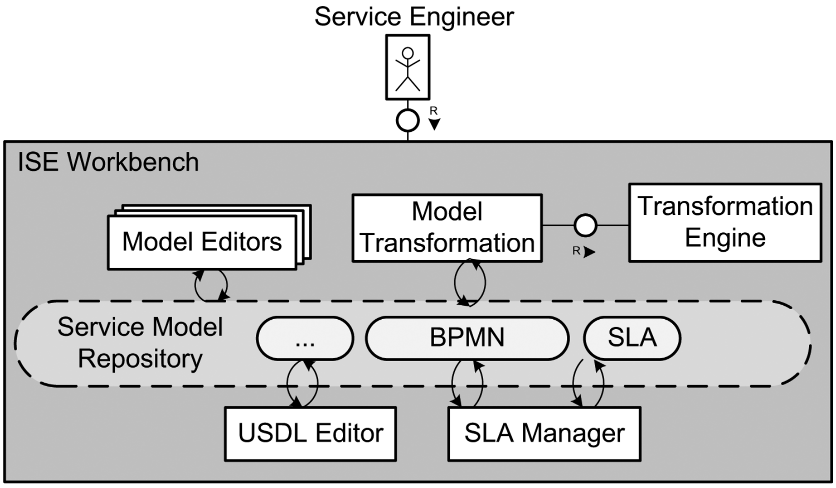
Based on the Eclipse technology, we investigated the choice of the 20 editors for each intersection of the matrix. Thereby, we focused on a reuse of existing technologies and editors in order to reduce the entry barrier for a service engineer. Consequently we integrated EMF-based editors such as the BPMN editor (cf. Figure 3), BPEL editor, UML (cf. OMG, 2007) editor, WSDL editor, XML schema editor, and the Ontoprise OWL editors. Whenever no existing tool could support an intersection of the ISE Framework, we implemented a new one. Especially the aspect of service description was not covered by an existing tool; therefore we developed an editor for the USDL that is outlined in Section 4.1. This editor presents a form-based view on the properties needed to model a services description. The details of USDL will be given in the following section. We also introduced an SLA manager and modelling approach which are detailed in Section 4.2.

Figure 4 depicts an architectural overview on the ISE Workbench. It shows the main concepts presented and the components realising the requirements of extensibility, flexibility, and to utilises models. The whole ISE Workbench serves as an entry point for a service engineer, i.e., each editor can be used separately. Each of these model editors stores the model created within a central service model repository. This repository is grounded on the formalism Ecore imposed by EMF. It is accessible by any component, e.g., the SLA manager, the USDL editor, the model editors, and model transformations. The models are synchronised through the model transformations. As the ISE Framework ascribes, different stakeholders can develop different models for defining a service. These models need to be integrated and synchronised, because of redundant or linked information. In the subsequent subsection we will detail this approach.

3.3 ISE model transformations

The problem of distributed and complex systems is well-known and has been tackled in several areas like software engineering. The most prominent approach in software engineering has been proposed by the OMG in 2002 (cf. MDA, 2003). The Model-Driven Architecture (MDA) has been developed by following two main principles: complexity reduction by abstraction and separation of concerns. The abstraction reduces a system level of detail, thus reducing the complexity.

Figure 4 Architectural view on the ISE workbench



The separation of concerns is followed by defining a domain specific approach, whereas each domain captures a concern.

The main pillars for realising MDA are: Computational Independent Model (CIM), Platform Independent Model (PIM), and Platform Specific Model (PSM). Beginning with the most abstract model, the CIM, an incremental refinement enriches a systems model, i.e., a stepwise creation of CIM, PIM, and PSM is recommended.

The approach fits the area of digital ecosystems, where a system is characterised by its services, which interact with each other. Following the MDA approach for the ISE Workbench, the CIM presents the business view (strategic and conceptual layer), where the PIM models the services interactions (logical layer) and finally the PSMs describe the technical implementation (technical layer).

The task of integrating and synchronisation of all ISE models are a major challenges because of the various people involved within the development process and the rising complexity of the models. To cope with these challenges we propose to integrate the models automatically.

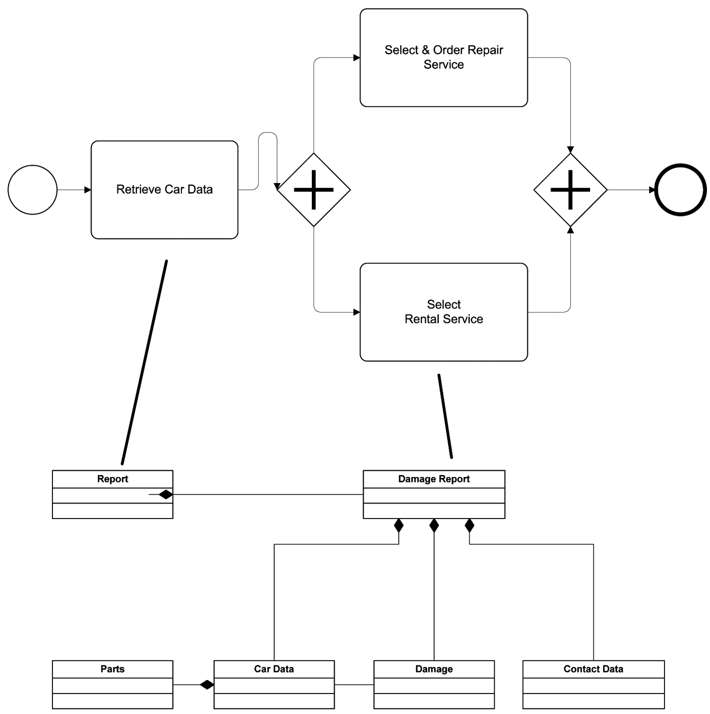
The 20 ISE models lead to multiple representations of information on different layers of abstraction in the corresponding service aspect. Each of these models has to be specified and maintained. Changes in one model have to be propagated into the affected models holding the overlapping information. This is a time-consuming and challenging task since each of the models must be adjusted to the changes.

We suggest implementing the integration of the models by an automatic support, i.e., a transformation. A transformation is defined by a mapping, which defines semantic correspondences between elements. That means different representations of information are assigned to each other. Figure 5 shows corresponding models for the car insurance example, which depicts the dependencies between models on the same layer but of different service aspects. Figure 5 shows the workflow and data models specified regarding the logical layer. While BPMN (cf. OMG, 2009) is used to represent process models in the logical layer, data is modelled using UML Class diagrams. The thick lines between the two models in Figure 5 depict artifacts that need to be synchronised. The example illustrates synchronisation of data specified in UML and used in the BPMN for processing. Therefore, the Report and Damage Report artifacts used in the workflow are defined in the UMLmodel. When one model changes (e.g., renaming or deletion), the depending models have to be updated. The user can perform these updates manually or through automatic support. One solution to enable an automatic approach is by using model transformations for implementing mappings.

The first step to enable the implementation of model transformations is to define one common formal representation of models. This is achieved by using EMF in context of the ISE Workbench. Based on this formalism, a domain specific language for model transformation can be used to define rules and apply them to the models. During the last years both academia and industry have proposed many model transformation languages. An overview is given in Czarnecki and Helsen (2006), which presents a classification of today'ss approaches. The two most prominent proposals in the context of MDA are Query, View and Transformation (QVT) (cf. QVT-Merge Group, 2004) and the ATLAS Transformation Language (ATL) (cf. Fabro et al., 2005).

We have chosen to rely on the rule-based language QVT to define model transformations executed by an engine implemented by ikv++. This allows for incremental and traceable transformations. It reduces the complexity, effort and errors in modelling a service using ISE, because of a support of automatic synchronisation by a rule interpreting engine.

Figure 5 Logical process and data description for car insurance service



4 Trading services in digital ecosystems

When services are traded via internet marketplaces two key challenges arise. Firstly, the offering of services on marketplace requires an adequate description of service functionality as well as the quality, legal, and financial terms under which the service can be provided. This enables consumers to find a suitable service to fulfil their business needs.

Secondly, services are typically offered under different terms for different prices. When engaging in business interaction, providers and consumers of services need to negotiate the precise terms of service provisioning and capture these terms in a formal contract called Service Level Agreement (SLA). For each service SLAs are negotiated based on the described service terms.

In this section we introduce a description language for services in digital ecosystems. We furthermore describe how SLAs are derived from the service description. Finally, we discuss the problem of managing SLAs for composite services, i.e., business processes created from services of different providers and offered to different customers.

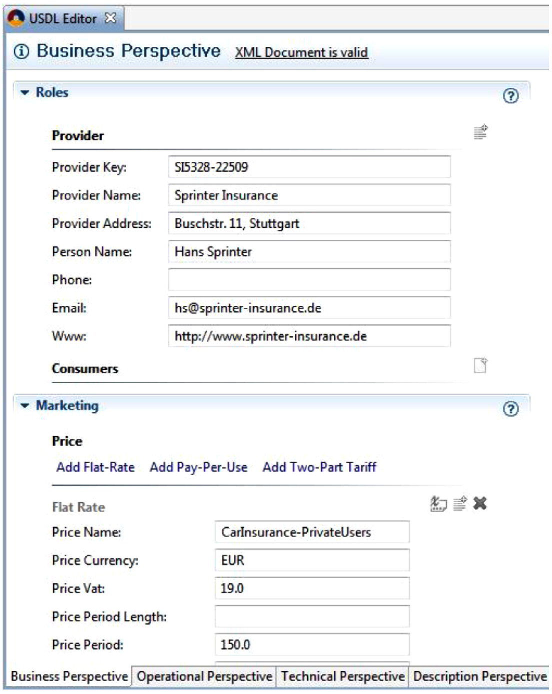
4.1 Universal service description language

In order to describe services in such a way that they can be offered on a service marketplace and found by interested consumers, the Unified Service Description Language (USDL) was developed. USDL allows the description of business, technical, and operational aspects of services (cf. Cardoso et al., 2009), which are relevant for most services independent of their nature (rather technical or more business oriented). The ISE Workbench supports the creation of USDL service descriptions (see Figure 6).

There are a number of existing approaches to describe services. Many of these approaches cover mainly technical service aspects. The WSDL specification allows the description of service interfaces including their input and output parameters, the communication protocol for accessing the service, as well as the endpoint where the service can be found. SAWSDL (cf. Farrell and Lausen, 2007) uses semantic annotations to WSDL elements to describe the meaning of input and output parameters as well as faults and thus provides an understanding of the functionality of the operations. These formal annotations allow machines to reason about the service. UDDI is a registry designed to support the registration and search of services. It allows the description of e.g., detailed provider information as well as a classification of the service functionality. However, important business information such as pricing, payment, legal or certification information about services is not handled.

USDL was designed with the objective to trade services via service marketplaces. Thus, it was important to describe also business and legal information of a service as well as the target consumers. To achieve that, USDL builds on existing work for describing business aspects of services such as PAS 1018 (cf. Moerschel and Hoeck, 2001) and the taxonomy for service description by O'Sullivan (2006). The following list summarises the three USDL perspectives and the descriptive aspects that are covered in Cardoso et al. (2010).

Figure 6 USDL editor: business perspective for car insurance service (see online version for colours)



Business perspective.

- *Provider, partner, and consumer information*: The USDL business perspective enables the description of different participants of service provisioning. Service provider information describes the organisation providing a service as well as a responsible contact person. Partners are stakeholders in service provisioning besides the provider. A profile describes target consumers of the service. This profile lists problems to be solved or goals to be achieved by the service and thus supports potential consumers to select the right service for their purposes.
- *Service level*: The service level includes QoS information (i.e., performance, dependability and security parameters), as well as a service rating.
- *Legal information*: This describes the rights and obligations of consumers and providers, as well as penalties that occur in the case of any party not respecting their obligations or the other party's rights.
- *Pricing and payment*: Information regarding different pricing options as well as payment methods for service usage are described.
- *Interaction*: This part comprises the means for invocation and execution of services. They may be realised in technical or nontechnical ways.

Operational perspective

- *Functional description*: The functional description allows describing what the service does, e.g., by using a classification scheme or operations.
- *Operations*: The operations of a service are described by interfaces with input and output parameters. They also describe where this service functionality is available.

Technical perspective: The technical perspective specifies different protocols to be used for this interaction, describing how a service is invoked and how service execution takes. Furthermore, it includes protocols regarding different security aspects. Here, USDL relies on existing standards. Listing 1 shows an excerpt of a USDL file for the car insurance example.

Listing 1 USDL file for car insurance service

```
<service>
 <business>
  <priceModels>
   <flatRate>
    <priceName> CarInsurance-PrivateUsers </priceName>
    <priceCurrency> EUR </priceCurrency>
    <priceVat> 19 </priceVat>
    <pricePeriod> 150 </pricePeriod>
    <paymentMethod> VISA </paymentMethod>
   </flatRate>
  </priceModels>
  <serviceLevel>
   <performance>
    <executionTime> 1h </executionTime>
   </performance>
  </serviceLevel>
 </business>
 <operational>
  <functionality>
   <udescription>
    <name> Car Insurance Service </name>
    <description> All-in-one car insurance service </
        description>
   </udescription>
  </functionality>
  <classifications>
   <udescription>
    <name> UN/SPSC: Car or truck insurance </name>
    <concept> 84131503 </concept>
   </udescription>
  </classifications>
  <operations> ... </operations>
 </operational>
 <technical> ... </technical>
</service>
```

4.2 SLA management for composite services

A formal contract called Service Level Agreement (SLA) typically regulates the provisioning of a service. An SLA describes the quality parameters under which a service is provided, the rights and obligations of the provider and the consumer as well as the price for service provisioning. Service provider and the consumer negotiate the SLA.

The management of SLAs including their negotiation, monitoring, and renegotiation is an important task of service providers for establishing trust among stakeholders of service trade in digital ecosystems. The ISE Workbench supports service providers with three important SLA management tasks: the creation of agreement templates based on USDL service descriptions, the negotiation of SLAs during composite service modelling, and the management of dependencies between different SLAs in a composition.

Automatic agreement template generation: In general, SLAs are negotiated according to the WS-Agreement specification described in Andrieux et al. (2007). It defines a negotiation protocol as well as the structure of an SLA document. The negotiation process starts by reading an agreement template file, which describes general service information as well as negotiable properties. This template is refined according to the consumers' needs and finalised as an agreement if the service provider and consumer agree on the terms. According to the specification an SLA contains provider information, service description terms describing what the service does, and a specification of service properties and their negotiable values as guarantee terms. However, WS-Agreement does not support the description of specific service properties. This is due to the fact that service descriptions vary greatly depending on the application domain. In order to describe the specific properties of a service within an SLA the USDL can be applied that makes WS-Agreement also suitable for digital ecosystems and rich service descriptions. It has description capabilities for financial and legal aspects, as well as stakeholders within an SLA. During a transformation process USDL service description elements are integrated into a WS-Agreement document. Information regarding pricing, legal and operational aspects is integrated into the WS-Agreement service description terms by integrating USDL code segments into the respective section. Information regarding service level properties is used to generate the service properties section describing all measurable service attributes as well as the guarantee terms specifying the negotiable service attribute values. The transformation was implemented as an Eclipse plugin for the ISE Workbench using openArchitectureWare (cf. openArchitectureWare.org, 2011). The generated agreement templates are deployed to the service marketplace along with the service, where they serve as base for negotiation. An SLA template for the car insurance example is shown in Listing 2.

Support for SLA negotiation: Composite service providers need to negotiate SLAs with all atomic service providers, whose services they integrate into their composition. In order to ease their work SLA negotiation functionality was incorporated into the ISE Workbench. The SLA negotiation wizard supports the request of SLA templates from a marketplace, SLA offer creation, and submission for agreement approval.

The SLA Negotiation Wizard shown in Figure 7 was implemented as an Eclipse plugin and was integrated with the SLA management component of the service marketplace to support the negotiation procedure.

Dependency management for composite service SLAs: In service compositions multiple services collaborate to achieve the composite service goal. The composite service provider needs to ensure that the SLAs negotiated for the atomic services and the composition enable the successful execution of the composition.

This is challenging due to dependencies among these services. A composition creates different types of service dependencies, e.g., with respect to produced and consumed resources, timing, Quality of Service (QoS), location of execution, and pricing. A service can depend on data or resources provided by another service. The price of a composed service depends on the pricing of the services forming the composition.

Listing 2 Agreement for car insurance service

```

<wsag:Template>
 <wsag:ServiceDescriptionTerm wsag:Name="Car␣Insur-ance">
  <usdl:ServiceDescription>
   <usdl:ServiceName>Car Insurance </usdl:ServiceName>
   <priceModels>
    <flatRate>
     <priceName> CarInsurance-PrivateUsers </priceName>
     <priceCurrency> EUR </priceCurrency>
     <priceVat> 19 </priceVat>
     <pricePeriod> 150 </pricePeriod>
     <paymentMethod> VISA </paymentMethod>
    </flatRate>
   </priceModels>
  </usdl:ServiceDescription>
 </wsag:ServiceDescriptionTerm>
 <wsag:ServiceProperties wsag:ServiceName="Car␣Insur-ance">
  <wsag:VariableSet>
   <wsag:Variable  wsag:Name="executionTime/>
␣␣</wsag:VariableSet>
␣</wsag:ServiceProperties>
␣<wsag:GuaranteeTerm␣wsag:Name="ExecTime␣GUARANTEE">
␣␣<wsag:ServiceLevelObjective>
␣␣␣<wsag:KPITarget>
␣␣␣␣<wsag:KPIName>␣executionTime␣</wsag:KPIName>
␣␣␣␣<wsag:Target>␣1h␣</wsag:Target>
␣␣␣</wsag:KPITarget>
␣␣</wsag:ServiceLevelObjective>
␣</wsag:GuaranteeTerm>
</wsag:Template>
```

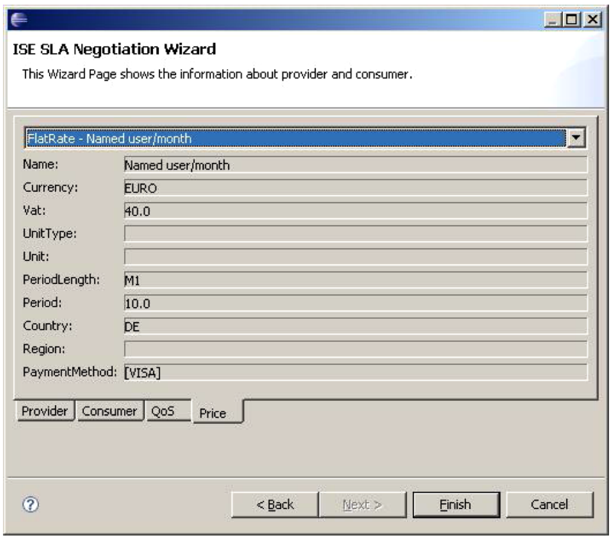
In order to consider these dependencies as part of SLA management, an approach for the handling of dependency information was developed by Winkler et al. (2010). Service dependency information is captured into a dependency model during the time of creating a composition and negotiating SLAs for the different services. This information is then used to support three different SLA management tasks. First, SLAs under negotiation are validated to check if all dependencies are fulfilled. Secondly, in the case of an SLA being violated during service provisioning the system checks whether this violation will affect any dependent service and result in further problems. Thirdly, requests for renegotiation of an SLA are evaluated regarding the effects of the anticipated changes on other services.

The functionality for dependency handling was implemented in the form of three plugins for the ISE Workbench. They cover the creation of dependency models, storage and validation of dependency information, and the evaluation of violation messages and renegotiation requests.

5 Discussion

This section discusses experiences and lessons learned, which were gained while developing and using the ISE Workbench (cf. Section 3) as well as during interviews and presentations at two conferences (cf. Scheithauer et al., 2009b, 2009c).

Figure 7 ISE SLA negotiation wizard showing price information (see online version for colours)



The ideas behind the framework and the workbench were overall very well received. The ISE Workbench handles the complexity well in that it offers a work-breakdown structure, which separates the engineering process and its result documents into 20 models. Furthermore, the incorporation of MDA and model transformation ideas provides consistency as well as integrity of service design and its realisation. Also, the distinct modelling editors, as tool support, decrease the complexity of the engineering process.

The workbench targets services' multidimensional aspects with five distinct descriptions, including a service's overall description, process, data, rules, and organisation. We reused available modelling languages and editors whenever possible, such in the case of BPMN and BPEL. On the other hand, this also involved the development of novel model types and realisation languages, such as USDL (cf. Section 4.1). Alas, this was not possible for all aspects, yet. For example, an appropriate solution for the rule aspect in the logical perspective is still amiss. Additionally, there was also critique on the introduction of novel models; we received suggestions to rather utilise existing approaches, such as UDDI.

The ISE Workbench targets different service engineering stakeholders and their modelling needs. The strategic perspective, for example, addresses a service's business model in terms of balanced score cards. The technical perspective, on the other hand, addresses Web service technology, such as BPEL and WSDL. However, even though the idea of having one integrated tool for service engineering was promising, it was suggested to have separate tools for different types of stakeholders in order to decrease the tool complexity as well as to improve the acceptance of the tool.

The usage of Eclipse technology as a basis for the development of the ISE Workbench went well. The Eclipse implementation of OSGI provides a means in order to integrate existing editors such as the BPMN process modelling tool. Additionally, EMF offers the technology for developing novel editors from the service engineering domain. Furthermore, the Eclipse ecosystem provides implementations for model transformation and means for distributed model persistency.

During the planning phase of the workbench, the choice for implementing model transformations was to use a declarative approach (cf. QVT-Merge Group, 2004), due to its recommendation by the OMG. However, it turned out that developers did to not appreciate the declarative QVT language, mainly because their experience lies with procedural or object-oriented languages, such as Java. In addition, during the development phase, the chosen tool support for declarative QVT hold potential for improvements. It is yet unclear, whether a switch to a procedural model transformation approach holds means for advancements in this matter.

The aforementioned lessons learned will be addressed in future work and further development of the ISE Workbench. In detail, we plan a two step approach:

The first step is an introduction of distributions of the ISE Workbench. Thereby, a distribution consists of a pre-selected set of editors and configured infrastructure components. The advantage is a reduced complexity and entry barrier for an introduction of ISE. However, the distributions should be extensible up to the complete ISE Workbench to support the complete functionality.

The second step is to look for possible cooperation in research and industry (cf. Scheithauer et al., 2010). Currently, the ISE Workbench is used in the public research project PROCESSUS for generating documentation using the model-based approach (cf. Heinrich et al., 2009), and in the EU research project SHAPE for supporting different variants of one service and their management.

6 Related work

There are a number of service engineering frameworks and integrated modelling environments to address the design and development of complex services. SoaML (cf. Sadovykh et al., 2009), MIDAS (cf. López-Sanz et al., 2008), and UML-S (cf. Dumez et al., 2008) follow an MDAbased approach for service modelling but target the development of SOAbased solutions and Web information systems. Unlike ISE, these approaches purely rely on UML models and focus on extending it for service modelling. The lack of a clear definition of software engineering process and the inexistence of organisational and information system perspectives makes it difficult to enable the participation of different stakeholders while designing a complex service. ISE decouples the overall framework and architecture from the underlying models and workbench in order to enable the use of different types of models while maintaining the coordination among different service aspects and engineering concerns.

Semantic Web services approaches such as WSMO, or OWLS provide frameworks, and tools (see Feier et al., 2005; Martin et al., 2004) to describe services with semantic annotations for better expressivity. Compared to ISE, these approaches concentrate their attention on the use of ontologies to enhance the expressiveness of descriptions of technical Web services and their interfaces (i.e., WSDL). While ISE also relies on ontologies, their use is not limited to the interfaces of services and can be also used to increase the expressiveness of the organisational and information system models that can be found, for example, in the business rule and human resource aspects.

There are also a number of commercial frameworks from several companies that target the use of multiple models to design services or SOA-based applications. For example, Select Architect, Business Architect, and Enterprise Architect, typically rely on business motivation modelling, business process modelling, componentbased models, and corporate data models to design information systems. Although these frameworks also rely on MDA approaches for code generation, they mainly target the design of enterprise applications for internal use within a company. Therefore, many important aspects of the services such as external service descriptions and SLA are not considered to enable the exchange of the services in the digital ecosystems environments.

7 Conclusions

Globalisation and novel technology change how physical goods and organisations produce and exchange services. In consequence, organisations concentrate on core competencies and rely on business partners for a complete supply chain. In line with this development, market places for business services emerge. These digital ecosystems foster service trade solely relying on internet technology.

Consequently, this process has an impact on how services need to be developed (cf. Papazoglou et al., 2008) and described in order to be traded in digital ecosystems. Kett et al. (2008) show that existing software service development processes comprise shortcomings in the context of digital ecosystems. These shortcomings include the overall complexity of realising tradable business services, the multidimensional aspects of business services, the diverse parties involved in the engineering processes, and how to manage service compositions from different service providers.

Considering this context, we introduced the ISE method, which allows developing and describing services for digital ecosystems. In particular, we presented the ISE Framework as a work-break-down-structure of 20 models in order to reduce the overall complexity of service engineering, to address distinct service aspects, and to support all involved parties. The ISE Workbench, on the other hand, is an implementation of the framework that implements MDA in order to speed up the development process as well as to provide consistency between the different service models. Following this, it was shown how to sufficiently describe services for digital ecosystems with a novel language that incorporates business and organisational aspects, such as pricing. Furthermore, it was presented how to deal with SLAs from composed services and their descriptions. Finally, lessons learned during the development of ISE in a three year research project are discussed along with possible improvements.

Future work includes addressing these improvements. Additionally, it is intended to conduct additional case studies and usability tests for further improving the overall performance of ISE and user acceptance.

Acknowledgements

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. The information in this document is proprietary to the following THESEUS consortium members funded by means of the German Federal Ministry of Economy and Technology: SAP Research, TU Dresden and, FZI. The information in this document is provided 'as is', and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

References

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

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. and Xu, M. (2007) *Web Services Agreement Specification (ws-agreement)*, Open Grid Forum Specification, http://www.ogf.org/documents/GFD.107.pdf, March 2007.

Barros, A.P. and Dumas, M. (2006) 'The rise of web service ecosystems', *IT Professional*, Vol. 8, No. 5, pp.31–37.

Cardoso, J., Winkler, M. and Voigt, K. (2009) 'A service description language for the internet of services', in Alt, R., Fhnrich, K-P. and Franczyk, B. (Eds.): *Proceedings First International Symposium on Services Science (ISSS'2009)*, Vol. 5 of *Leipziger Beitrge zur Wirtschaftsinformatik*, Logos Verlag Berlin, pp.229–240.

Cardoso, J., Barros, A., May, N. and Kylau, U. (2010) 'Towards a unified service description language for the internet of services: requirements and first developments', *IEEE International Conference on Services Computing*, IEEE Computer Society Press, Miami, USA, pp.602–609.

Czarnecki, K. and Helsen, S. (2006) 'Feature-based survey of model transformation approaches', *IBM Systems Journal, Special Issue on Model-Driven Software Development*, Vol. 45, No. 3, pp.621–645.

Dumez, C., Gaber, J. and Wack, M. (2008) 'Model-driven engineering of composite web services using uml-s', *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, ACM, Linz, Austria, pp.395–398.

Fabro, M.D.D., Bezivin, J., Jouault, F., Breton, E. and Gueltas, G. (2005) 'Amw: a generic model weaver', *Proceedings of the 1ère Journée sur l'Ingénierie Dirigée par les Modèles (IDM05)*, Paris, France, pp.105–114.

Farrell, J. and Lausen, H. (2007) *Semantic Annotations for wsdl and xml Schema*, W3C Recommendation.

Feier, C., Roman, D., Polleres, A., Domingue, J., Stollberg, M. and Fensel, D. (2005) 'Towards intelligent web services: the web service modeling ontology (WSMO)', *International Conference on Intelligent Computing (ICIC)*, Citeseer, pp.23–26.

Heinrich, M., Boehm-Peters, A. and Knechtel, M. (2009) 'A platform to automatically generate and incorporate documents into an ontology-based content repository', *DocEng '09, Proceedings of The9th ACM Symposium on Document Engineering*, Munich, Germany, pp.43–46.

Kett, H., Scheithauer, G., Weiner, N. and Weisbecker, A. (2009) 'Integrated Service Engineering (ISE) for service ecosystems: an interdisciplinary methodology for the internet of services', in Cunningham, P. and Cunningham, M. (Eds.): *Proceedings of eChallenges e-2009 Conference*, IIMC International Information Management Corporation, Istanbul, Turkey.

Kett, H., Voigt, K., Scheithauer, G. and Cardoso, J. (2008) 'Service engineering in business ecosystems', *Proceedings of the XVIII. International RESER Conference*, Stuttgart, Germany.

López-Sanz, M., Acuña, C.J., Cuesta, C.E. and Marcos, E. (2008), 'Modelling of service-oriented architectures with UML', *Electronic Notes in Theoretical Computer Science*, Vol. 194, No. 4, pp.23–37.

Lusch, R.F. and Vargo, S.L. (2006) 'Service-dominant logic: reactions, reflections and refinements', *Marketing Theory*, Vol. 6, No. 3, p.281.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T. R., Sirin, E., Srinivasan, N. and Sycara, K. (2004) *OWL-S: Semantic Markup for Web Services*, Technical Report , Member Submission, W3C.

MDA (2003) *MDA Guide Version 1.0.1*, OMG document omg/2003-06-01.

Moerschel, I.C. and Hoeck, H. (2001) *Grundstruktur fuer die Beschreibung von Dienstleistungen in der Ausschreibungsphase*, Beuth Verlag GmbH. Ref.Nr. PAS1018:2002-12.

Object Management Group (OMG) (2006) *Specification: Meta Object Facility (MOF), Version 2.0*, http://www.omg.org/spec/MOF/2.0/PDF

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

Object Management Group (OMG) (2009) *Specification: Business Process Modeling Notation (BPMN), Version 2.0*, http://www.omg.org/docs/dtc/09-08-14.pdf

openArchitectureWare.org (2011) *openarchitectureware – The Leading Platform for Professional Model-Driven Software Development*, Project page: http://www.openarchitectureware.org/index.php, last accessed: 2011-12-02.

Organization for the Advancement of Structured Information Standards (OASIS) (2004) *Specification: Universal Description Discovery and Integration (UDDI) Version 3.0*, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

OSGi Alliance (Ed.) (2007) *OSGi Service Platform: The OSGi Alliance*, IOS Press, USA.

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

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

Papazoglou, M.P., Traverso, P., Dustdar, S. and Leymann, F. (2008) 'Service-oriented computing: a research roadmap', *Int. J. Cooperative Inf. Syst.*, Vol. 17, No. 2, pp.223–255.

Peneder, M., Kaniovski, S. and Dachs, B. (2003) 'What follows tertiarisation? Structural change and the role of knowledge-based services', *The Service Industries Journal*, Vol. 23, No. 2(146), pp.47–66.

QVT-Merge Group (2004) *MOF 2.0 Query/Views/Transformations*, Object Management Group, ad/04-04-01, Revised Submission.

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

Sadovykh, A., Hahn, C., Panfilenko, D., Shafiq, O. and Limyr, A. (2009) 'SOA and SHA tools developed in SHAPE project', *Proceedings of the Tools and Consultancy Track. European Conference on Model-Driven Architecture (ECMDA-09), in Conjunction with Fifth European Conference on Model-Driven Architecture Foundations and Applications*, 23–26 June, Enschede, Netherlands, p.113, CTIT Proceedings Series (WP), Vol. 09-12, 6/2009.

Scheithauer, G., Augustin, S. and Wirtz, G. (2008) 'Describing services for service ecosystems', in Feuerlicht, G. and Lamersdorf, W. (Eds.): *ICSOC Workshops*, Vol. 5472 of *Lecture Notes in Computer Science*, Springer, Sydney, Australia, pp.242–255.

Scheithauer, G., Augustin, S. and Wirtz, G. (2009a) 'Business modeling for service engineering: toward an integrated procedure model', *Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), Boston, Massachusetts, USA, 1-3 July, 2009*, Boston, MA, USA, pp.322–327.

Scheithauer, G., Voigt, K., Bicer, V., Heinrich, M., Strunk, A. and Winkler, M. (2009b) 'Integrated service engineering workbench: service engineering for digital ecosystems', *Proceedings of the International ACM Conference on Management of Emergent Digital Ecosystems (MEDES)*, Lyon, France, pp.446–449.

Scheithauer, G., Voigt, K., Bicer, V., Heinrich, M., Strunk, A. and Winkler, M. (2009c) 'ISE workbench: integrated service engineering', *Business Process Management Conference Demo Track*, Ulm, Germany, p.1.

Scheithauer, G., Kett, H., Kaiser, J., Hackner, S., Hu, H. and Wirtz, G. (2010) 'Business modeling for service engineering: a case study in the IT outsourcing domain', *SAC 2010, 25th Symposium On Applied Computing, Enterprise Engineering Track*, Sierre, Switzerland, pp.118–123.

Scheithauer, G. and Winkler, M. (2008) *A Service Description Framework for Service Ecosystems*, Bamberger Beiträge zur Wirtschaftsinformatik 78, Bamberg University, ISSN 0937-3349.

Scheithauer, G. and Wirtz, G. (2010) 'Business modeling for service descriptions: a meta model and a UML profile', *APCCM 2010, 7th Asia-Pacific Conference on Conceptual Modelling*, Brisbane, Australia, pp.79–88.

Steinberg, D., Budinski, F., Paternostro, M. and Merks, E. (2008) *EMF Eclipse Modelling Framework*, 2nd ed., Addison Wesley, Amsterdam, Netherlands.

Theseus/TEXO (2011) *Theseus TEXO – Business Webs in the Internet of Services*, http://www.theseus-programm.de/en/914.php, last accessed: 2011-12-02.

Winkler, M., Springer, T., Trigos, E.D. and Schill, A. (2010) 'Semi-automatic dependency model creation based on process descriptions and slas', *ACT4SOC 2010 – Proceedings of the 5th International Conference on Software and Data Technologies*, Athens, Greece, pp.93–105.

Zachman, J.A. (1987) 'A framework for information systems architecture', *IBM Systems Journal*, Vol. 26, No. 3, pp.276–292.