BAMBERGER BEITRÄGE

ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK

ISSN 0937-3349

Nr. 78

A Service Description Framework for Service Ecosystems

Gregor Scheithauer and Matthias Winkler

October 2008

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

Distributed and Mobile Systems Group

Otto-Friedrich Universität Bamberg

Feldkirchenstr. 21, 96052 Bamberg, GERMANY

Prof. Dr. rer. nat. Guido Wirtz

http://www.uni-bamberg.de/pi/

Due to hardware developments, strong application needs and the overwhelming influence of the net in almost all areas, distributed and mobile systems, especially software systems, have become one of the most important topics for nowadays software industry. Unfortunately, distribution adds its share to the problems of developing complex software systems. Heterogeneity in both, hardware and software, concurrency, distribution of components and the need for interoperability between different systems complicate matters. Moreover, new technical aspects like resource management, load balancing and deadlock handling put an additional burden onto the developer. Although subject to permanent changes, distributed systems have high requirements w.r.t. dependability, robustness and performance.

The long-term common goal of our research efforts is the development, implementation and evaluation of methods helpful for the development of robust and easy-to-use software for complex systems in general while putting a focus on the problems and issues regarding the software development for distributed as well as mobile systems on all levels. Our current research activities are focussed on different aspects centered around that theme:

- *Robust and adaptive Service-oriented Architectures*: Development of design methods, languages and middleware to ease the development of SOAs with an emphasis on provable correct systems that allow for early design-evaluation due to rigorous development methods and tools. Additionally, we work on approaches to autonomic components and container-support for such components in order to ensure robustness also at runtime.
- *Agent and Multi-Agent (MAS) Technology:* Development of new approaches to use Multi-Agent-Systems and negotiation techniques, for designing, organizing and optimizing complex distributed systems, esp. service-based architectures.
- *Context-Models and Context-Support for small mobile devices*: Investigation of techniques for providing, representing and exchanging context information in networks of small mobile devices like, e.g. PDAs or smart phones. The focus is on the development of a truly distributed context model taking care of information reliability as well as privacy issues.
- *Peer-to-Peer Systems*: Development of algorithms, techniques and middleware suitable for building applications based on unstructured as well as structured P2P systems. A specific focus is put on privacy as well as anonymity issues.
- *Visual Programming- and Design-Languages*: The goal of this long-term effort is the utilitization of visual metaphores and languages as well as visualization techniques to make design- and programming languages more understandable and, hence, easy-to-use.

More information about our work, i.e., projects, papers and software, is available at our homepage. If you have any questions or suggestions regarding this report or our work in general, don't hesitate to contact me at `guido.wirtz@uni-bamberg.de`

Guido Wirtz

Bamberg, April 2008

A Service Description Framework for Service Ecosystems

Gregor Scheithauer and Matthias Winkler

Abstract Recently, service orientation strongly influenced the way enterprise applications are build. Service ecosystems are an evaluation of service orientation which provide means to trade services between companies like goods. To allow service offering, discovering, selection, and consumption a common way to describe services is a necessity. This paper discusses existing approaches to describe certain service aspects. Finally, a Service Description Framework for service ecosystems is proposed and exemplifed.

Keywords Service Ecosystems, Service Description

Contact:

Gregor Scheithauer
Siemens AG
Corporate Technology
Knowledge Management
Otto-Hahn-Ring 6
81739 Munich, Germany
gregor.scheithauer.ext@siemens.com

Matthias Winkler
SAP Research CEC Dresden
Chemnitzer Str. 42
01187 Dresden, Germany
Matthias.Winkler@sap.com

Contents

1 Introduction **1**

2 Existing Approaches **1**

3 Running Example **3**

4 The Service Description Framework **3**

4.1 General Information 3

4.1.1 Title 4

4.1.2 Identifier 4

4.1.3 Creator 5

4.1.4 Publisher 5

4.1.5 Date 5

4.1.6 Version 5

4.2 Functionality 5

4.2.1 Function 6

4.2.2 Classification 6

4.2.3 Benefit 6

4.3 Service Interaction 7

4.3.1 Message Exchange Pattern 7

4.3.2 Protocol 7

4.3.3 User Interface 7

4.4 Data 8

4.4.1 Input and Output Data 8

4.4.2 Business Objects 8

4.4.3 Documentation 8

4.5 Business 8

4.5.1 Price . . . 9
4.5.2 Discount . . . 9
4.5.3 Payment . . . 10
4.6 Legal . . . 10
4.6.1 Rights . . . 10
4.6.2 Obligations . . . 11
4.6.3 Penalties . . . 11
4.6.4 Terms of Use . . . 11
4.7 Quality of Service . . . 11
4.7.1 Performance . . . 12
4.7.2 Dependability . . . 12
4.8 Security and Trust . . . 13
4.8.1 Security . . . 13
4.8.2 Trust . . . 13
4.9 Rating . . . 14
4.9.1 Community Feedback . . . 14
4.9.2 Expert Test Rating . . . 15
4.9.3 Certification . . . 15

5 Conclusion and Future Work **15**

References **17**

A List of previous University of Bamberg reports **19**

1 Introduction

Enterprise application vendors are currently experiencing a strong shift towards service-oriented architectures (SOA). Several analysts emphasize the importance of the SOA topic [5, 23]. The vision of service ecosystems is an evolution of service orientation and takes services from merely integration purposes to the next level by making them available as tradable goods on service delivery platforms [3].

Nowadays web services in service-oriented architectures are described with the Web Service Description Language (WSDL) [18] and Universal Description Discovery and Integration (UDDI) [7]. These specifications address mainly technical information about service functionality and usage. This is suitable for company-internal enterprise applications with a focus on integration, where legal and financial aspects do not apply.

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

Thus, there is a strong need for a comprehensive service description framework which addresses service ecosystem requirements and supports all steps of service trade.

There are first approaches for describing different aspects of services. A major work in this area is J.O'Sullivan's PhD thesis [16]. He created a taxonomy for the non-functional description of services. Moerschel and Hoeck [15] tackled the service description topic from the perspective of service procurement. Other service description approaches are presented in section 2.

The remainder of this paper is structured as follows: section 2 discusses identified existing approaches. Following this, a running example is presented. Section 4 proposes a service description framework and section 5 concludes this work.

2 Existing Approaches

This section introduces several existing description approaches which cover non-functional service properties, business services, software requirements, resources in general, and web services. Origins of these approaches include academics, industry, professionals, and standardization institutes. The different purposes and the heterogeneous backgrounds offer a solid and rich first basis for the *Service Description Framework*.

In his doctoral thesis, O'Sullivan [16] developed a taxonomy for describing non-functional properties for technical as well as business service properties. He defined non-functional properties as constraints of functionality. The strength of this approach is the wide range of attributes, such as pricing and payment, security and trust, and obligations, to only mention a few. However, functional attributes were not considered. Nevertheless, the taxonomy serves as a stable basis for the service description framework.

Moerschel and Hoeck described in the Public Available Specification (PAS) 1018:2002 [15] an *essential structure for the description of services in the procurement stage*. This specification aims to advance the industrialization of the service sector, to boost service trade, and to improve transparency within the service sector. Their work is based on a study about electronical market places, and industry work shops. They depict a procurement process with 14 steps which covers the phases before, during, and after service supply. In addition, they introduce 16 attributes (e.g. service classification, location of provisioning, and delivery terms) to describe services for different steps in the procurement process before service supply. Most of these attributes are quite unique and complement O'Sullivan's work. These attributes advance the service description framework with business and functional related aspects.

The IEEE 830-1998 is a recommendation for writing specifications for software requirements [21]. While the problem tackled by the recommendation has a very different background as compared to service description, there are a number of interesting requirements aspects that are of interest for our work. Important attributes are the description of relevant functions and interfaces, input and output, availability, performance and reliability. These attributes improve the service description with functional and quality related aspects.

The Dublin Core Metadate Element Set (DCMES) was developed by the Dublin Core Metadata Initiative (DCMI) [1]. DCMI aims to develop standards for metadata interoperability. Its members are from libraries, academia, and museum communities. DCMES (ISO Standard 15836) offers 15 attributes to domain-independently describe resources. These attributes are very helpful to describe apparent aspects of a service, such as the service name and the service publisher (e.g. Creator and Subject). DCMES is used to describe HTML [24] web page meta data. This allows software agents to automatically interpret and classify the web page content. Moreover, DCMES is also used by the Web Service Modeling Language (WSML) [19] to express non-functional properties of a service within an ontology. These attributes describe mere meta information, and do not hold valuable information which could match service consumers' preferences. Nevertheless, most of these attributes are unique and ameliorate the service description framework with general aspects.

The W3C standard Web Service Description Language (WSDL) for describing web services is currently available in version 2.0 [18]. It covers technical aspects such as the interface of the service, the input and output parameters, the messages and message exchange pattern for interacting with the service. Also, the protocol for interacting as well as the location of the service are specified. WSDL is widely accepted in the industry and is a target platform for the service description framework, though it does not define any means to define non-functional attributes. Nevertheless, the concepts of are incorporated into the service description framework to improve functional and data aspects.

The Universal Description, Discovery and Integration (UDDI) [7] OASIS standard provides the means for describing service meta information. White Pages include name and contact information for each service. Yellow Pages provide a schema to classify services. Lastly, Green Pages cover technical information, such as interfaces. UDDI provides few attributes to describe non-functional attributes. Still, the concepts are incorporated into the service description framework to improve general and functional aspects.

3 Running Example

This section presents the Eco Calculator Service. This example is used throughout this paper to illustrate how to apply the service description framework. Moreover, the Eco Calculator Service is the use case for the BMWi-aided research project TEXO [17] and described in more detail in [13].

Within this use case, the Eco Calculator Service is created by UGS, a product life-cycle management company, and provided by SAP on a service delivery platform. The service calculates eco values for different products (e.g. the eco value of a car seat). The eco value is a rating taking into consideration information such as energy consumption and pollution during the manufacturing process, used materials, and recycling information of the product being analyzed. The eco value of a product is calculated in a recursive way by combining the eco values of all of the products subparts. A service consumer can use the service to analyze its own product by providing a bill of material containing information on the used material. The service will then analyze the subparts and retrieve the necessary information from different sources (e.g. third party service offering information on eco values of different registered materials). As a result of the process the eco value of the product is provided. In the case that specific requirements are met, a certificate may be issued for the product.

4 The Service Description Framework

Figure 1 depicts the service description framework. It consists of nine facets and corresponding attributes. Facets group attributes to reduce complexity. The nine service facets will be described along with the corresponding attributes in the following subsections.

4.1 General Information

General Information covers the self-evident attributes of a service. These attributes provide service consumers with a basic understanding and an appropriate perception of services.

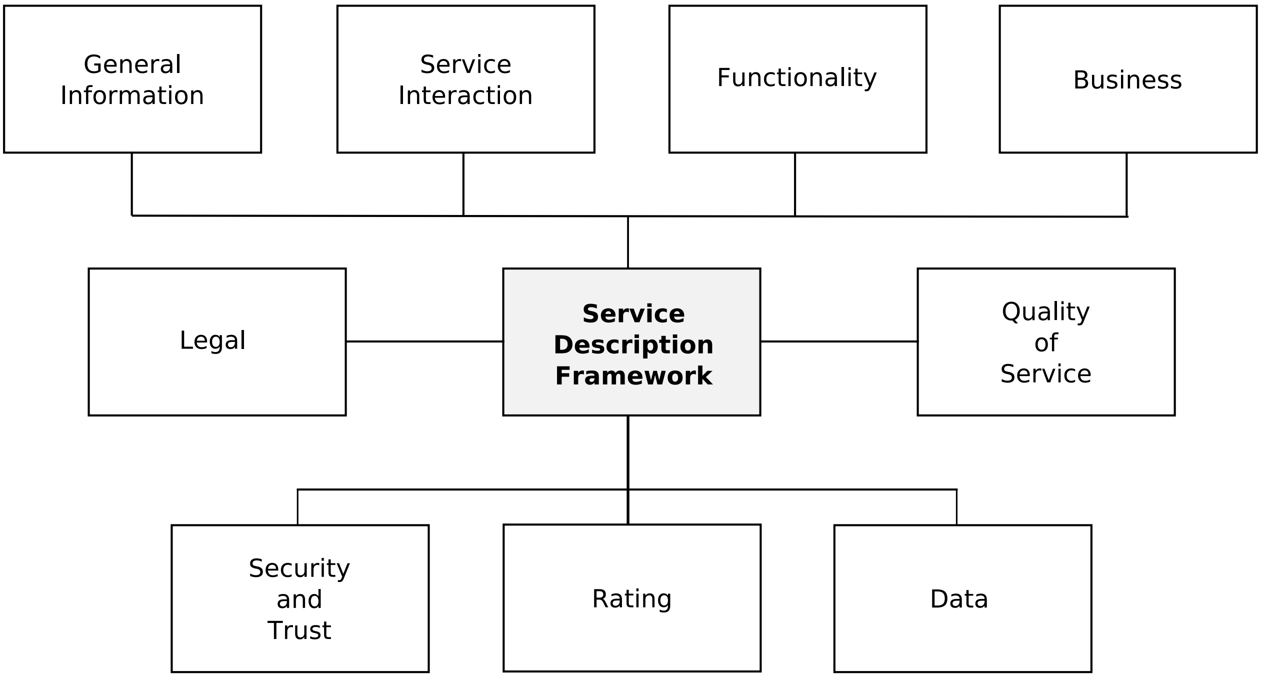

Figure 1: Nine Service Facets.

4.1.1 Title

This attribute represents the *name* of a service. For the service description framework, Title will be a name by which services are formally known and is mandatory. A service has exactly one Title. The Title is represented with text. For example the Eco Calculator service's title would be *Eco Calculator*. The Title allows service providers to give a first idea what the service does. Service consumers use the Title as a reference to the service itself.

4.1.2 Identifier

This attribute exemplifies tokens to uniquely *name* services. For the service description framework, the Identifier allows to reference services unambiguously. Whereas the service Title names services also, the Identifier is more explicit because it is guaranteed to be unique, which is not necessarily the case for the Title. Additionally, the attribute is very helpful for information systems and database systems. This is analogous to the concept of European Article Number (EAN), and the Global Trade Item Number (GTIN). A service has exactly one Identifier and this attribute is mandatory. The Identifier is represented with a string or a number conforming to a formal system such as EAN or GTIN. For example the identifier of the Eco Calculator service could be *9783125171541*.

4.1.3 Creator

This attribute represents the person or the organization who created the service. For the service description framework, Creator is associated with persons or organizations who created the service. This attribute is mandatory. A service has one or more Creators. The attribute Creator is represented with persons' or organizations' names. For example the creator of the Eco Calculator service could be *UGS*. This attribute provides service consumers with the information on who built the service in the first place. This knowledge is important for marketing, legal issues, and trust.

4.1.4 Publisher

This attribute represents the person or organizations who published and offers services to service consumers. The Publisher is synonymous with the service provider. This attribute is mandatory. A service has one Publisher. The attribute Publisher is represented with persons' or organizations' unique names. For example, the publisher of the Eco Calculator service could be *SAP*. This attribute reveals service consumers the identity of the responsible person or organization who provide a specific service. This information is crucial for marketing, negotiation, contracting, legal issues, and trust.

4.1.5 Date

This attribute represents important dates associated with services. For the service description framework, a Date represents an event in the service life-cycle, such as *creation date*, *published date*, and *last update*. This attribute is optional. A service has none or more Dates. The version attribute is represented with ISO 8601 [12]. For example, the published date for the EcoCalculator service could be *2008-06-11*.

4.1.6 Version

This attribute represents services' actual version. This attribute is mandatory. A service has one Version. The Version attribute is represented with a normal name system. For example, the version for the EcoCalculator service could be *1.0*.

4.2 Functionality

Functionality provides the service consumer with an understanding of what the service is actually providing and thus, what the consumer can expect from the service. Attributes include functions, a service classification, and a benefits statement. For example, a service might be classified as a computation service in the eco domain with the functions calculate eco value for car parts and issue eco certificate.

4.2.1 Function

This attribute represents the major functions that services provide. The term *operation* may be used synonymously to the term *function*. For the service description framework, a function represents partially or completely a service's functionality. A service has one or more functions. A function allows a service consumer to access a services' functionality and correspondence with the data section. Often, services' functionality is divided into several functions. This allows service consumers to access particular subsets of services' functionality. Additionally, a service's outcome might be different, depending which functions to perform in what order (cf. service interaction section). E.g. a flight booking service offers different functions, such as to browse different flights, to plan a flight route, to book a flight, and to pay for it. In some cases just some of these functions are necessary for the service consumer to achieve its goals. However, to book a flight, one of more specific functions must be invoked in a predefined way. This attribute is mandatory. This attribute is represented by a formal naming system. This would include a function's name, involved parties, data which is processed by the service, and the outcome. For example, a function for the Eco Calculator services might be *calculateEcoValue*. Another function could be *createCertificate*. This attribute is important, since it allows service consumers to access services' functionality, thus it is the functional interface which glues services' functionality, involved parties, and processed data together.

4.2.2 Classification

This attribute allows to apply the service into one or more classification systems. A classification is a system of interrelated terms which generally form a hierarchically structure. The terms allow to specify the kind of service, an unique identifier, and a reference to a classification standard. For the service description framework, the classification attribute serves as reference to a classification standard, such as eCl@ss and UNSPSC [8, 9]. While the classification attribute is optional, it may be the case that a service is classified according to multiple classification standards. For that reason it is necessary to model service classification as a tuple of a reference to a classification standard and a unique identifier.

For example, a classification for a medical monitoring service would be *UNSPSC:14111539*. Classifying services supports potential service consumers to discover services more easily and service consumers are enabled to find all suppliers of a specific service kind.

4.2.3 Benefit

The benefit of a service is the gained outcome of the service for the service user. This information is needed for a potential service consumer to determine whether this particular service has the potential to suit its needs. While it is difficult to measure the benefit of a service there is a great value in providing the user with helpful information. This information will be much more accurate than a functional classification or the description of methods of a service because it is possible to describe the specialties of the service. The *benefit* attribute will be represented by a natural language description to be understandable by human beings. An example for this attribute is the following: The Eco Calculator service calculates the eco value of a product according to the norm AUS2008.

4.3 Service Interaction

Service interaction covers a number of different attributes that describe the interaction of an entity with a service. This topic can be divided into the subtopics UI, message exchange patter, and protocol.

4.3.1 Message Exchange Pattern

The message exchange pattern (MEP) covers the aspect of interactions between two entities (service-service, service-human) on the level of messages being exchanged between them. The combination of several messages (request-response, message, request-response-confirm) forms a message exchange pattern necessary for the provisioning of a service. A number of different MEPs are defined for WSDL [6]. To model a MEP it is necessary to specify the *mepURI* identifying the specific MEP (e.g. mepURI="http://www.w3.org/2006/01/wsdl/in-opt-out") as well as describing the concrete messages of the interaction for each operation. An operation is a single function that is provided to a service consumer by the service. Functions will be described in a later section.

4.3.2 Protocol

A number of different protocols may be used for the interaction with a web service. One example is the usage of HTTP to transmit SOAP messages. This is defined using the *protocol* attribute. It is also necessary to state which style of SOAP messages (RPC or document) is to be used for the interaction between two entities. This is done via the *messageType* attribute. The Eco-Calculator would be modeled as follows: *protocol=SOAP/HTTP* and *messageType=document*.

4.3.3 User Interface

Another possible type of interaction is the interaction of a human user with a service. In cases users need to interact with services a user interface (UI) needs to be provided. Therefore a service needs to provide some information regarding its user interface. The UI will be a representation of the single UI elements needed as well as their relationship (e.g. layout information). Momentary the UI requirements are not yet clear. This will be determined throughout the further course of this work. The user interface description will be represented through a separate document to achieve a good separation of concerns. The service description provides the attributes *hasUI* stating whether a UI description exists for this service and *uiRef* which contains a link to the UI description. The EcoCalculator has the following settings: *hasUI=true* and *uiRef=anyURI*.

4.4 Data

The data facet describes all kinds of data handled by a service. This includes the input and output data of each service, business objects which are affected by the service execution, and documentation that is available about the service. Most of these concepts can be directly mapped to WSDL [18].

4.4.1 Input and Output Data

The input and output data of a service is the data passed to the service for execution and returned to the user after service execution finished. This data may be used to manipulate other data objects (business objects) during the course of service execution. The modeling of data will be realized through the attribute *ioData* which should use an identifier to reference descriptions of data objects modeled in a representation such as XMLSchema. Example: *iodata=anyUri*.

4.4.2 Business Objects

In the process of executing a service business object (BO) might be manipulated (e.g. the bill of material business object is needed by the EcoCalculator service). The BOs affected by the service execution will be described using the *businessObjects* attribute. This provides the service user with additional understanding of what the service does by setting it into a context. Example: *businessObjects=BOM*.

4.4.3 Documentation

The description of a service may have various forms. Besides textual or semantic description of certain service attributes, there might also be a complex documentation describing the details of a service in a human readable form. Documentation will be provided in the form of full documents. It is attached to the service via a link. The attribute *documentationType* will describe what kind of documentation is available. The attribute *documentationRef* will provide a link to where the information can be found. Example: *documentationType=marketing*, *documentationRef=anyUri*.

4.5 Business

This section comprises monetary and marketing related attributes. These include price, payment, discounts, and delivery unit. Price depicts the amount the service consumer must pay for service usage. Payment describes accepted payment instruments, e.g. credit cards, cash, etc. Discount addresses price reductions for specific service consumers, e.g. for regular customers. Delivery unit holds information about how service outcomes are packaged and provided to service consumers.

4.5.1 Price

This attribute represents an economical numerical value for services. PAS 1018:2002 depict two price attributes. The first price attribute describes a service providers' price conception. The second price attribute specifies the service consumers' price idea. O'Sullivan, however, offers a more holistic approach. His work includes four different types of price. It is possible to relate all price types to entities such as *time*, *area*, etc. This allows to specify different prices for different time or areas of service usage. Additionally, tax information can be included as well. For the service description framework, the approach from O'Sullivan is mostly adapted. The four price types are explained briefly. An *absolute price* specifies a specific amount of money and a currency. E.g. Booking a flight costs EUR 10. A *proportional price* depicts a percentage with respect to a given value. E.g. a life insurance monthly rate is 1% of one's yearly income. A *ranged price* allows to specify a price range with a minimum and maximum *absolute price* or *proportional price*. Service providers may use this price type in case it is impossible to set an absolute price. To fix the final price is part of the negotiation phase between service provider and service consumer. E.g. a rental car's price per day ranges from EUR 50 to EUR 70. The final price depends on the final car configuration. A *dynamic price* covers auctions, where the price matching is based on natural supply and demand. E.g. a service provider offers train tickets and potential service consumers bet an amount of money they perceive as their value. A service has one or more prices. A price is a mandatory attribute. The metric for currencies is the ISO 4217:2001 [11]. The price amount is represented by a numerical data type. The granularity taxonomy is taken from O'Sullivan [16]. For example, the Eco Calculator service's absolute price would be *amount: 5.35; currency: EUR; exludedTAX: 19%;*.

4.5.2 Discount

This attribute specifies possible price reductions and is complements the Price attribute. A service has no or more discounts for a *price*. For the service description framework, most of O'Sullivan's work is adapted. In general, *discount* attributes can be offered within a specified time segment (*temporal*), for a specific location (*locative*), or a given *condition*. Additionally, the Discount attribute is differentiated between payment related discounts and payee related discounts. *Payment related discounts* group types of discounts that refer to how payment is done. This includes *early payment*, *type of payment instrument*, *coupons*, *location of payment*, and *volume invocation*. *Payee related discounts* relates to the service consumer, who pays for a service. This includes *age group*, *student*, *membership*, and *shareholder*. Dates are represented with ISO 8601 [12], and regions with ISO 3166 [10]. For example, a discount for the Eco Calculator service could be *Type of discount: volume invocation (payment); 10% off after using the service more than 100 times a month.* Discounts offer service providers a flexible way to attract different potential service consumers with a single price.

4.5.3 Payment

This attribute specifies feasible options to fulfill service consumer's payment liability. Where PAS 1018:2002 depicts only a placeholder for payment, O'Sullivan offers a more thorough approach. However, they do not contradict each other. For the service description framework, the more formal approach from O'sullivan is followed. According to him, payment is complemantary to the *price* attribute. He subdivided this attribute into four models: payment options, payment schedules, payment instruments, and payment instrument types. A *payment option* constitutes, whether a particular payment option is the preferred one, whether there is a charge connected to the payment option, where a payment option is available, and specific conditions for a payment option. A *payment schedule* depicts when a payment is due. This attribute has two dimensions. Firstly, it is possible to specify a percentage of the whole price with respect to services' provisioning moment (before, during, and after). Secondly, percentages together with concrete dates can be specified. A *payment instrument* is issued by a service provider. It supports one or more currencies and relates to specific regions. Four *payment instrument types* are available: *card based instruments*, *cheques*, *cash*, and *vouchers*. A service has one or more payment options, schedules, instruments, and instrument types. Payment is a mandatory attribute. Dates are represented with ISO 8601 [12], currencies with ISO 4217 [11], and regions with ISO 3166 [10]. For example, a payment for the Eco Calculator service would be *Cash is the preferred payment; No charge for cash payment; Only available in Germany; Complete payment is due before service provisioning; Accepted currency is: EUR;* This quite complex model allows service providers within an Internet of Services to specify payment in very flexible ways. Numerous payments are conceivable and do not restrict unforeseen business models.

4.6 Legal

When providing and consuming services a number of legal aspects come up which need consideration. The representation of legal issues in the service description is supposed to facilitate the process of finding suitable services for a service consumer by formalizing those issues and thus allowing for their inclusion into the search procedure. Also, we envision to do further research regarding the support of automatic negotiation and monitoring of some of the legal aspects (where possible). The following sections introduce the attributes rights, obligations, penalties, and terms of use. The descriptions in this section are mainly based on the work presented in [16].

4.6.1 Rights

Rights can apply to the service consumer as well as the service provider. They may refer to service usage (e.g. the consumer has the right to offer the service as part of a service composition), service provisioning (e.g. the provider has the right to refuse service provisioning), or be of general nature (e.g. the provider may store data regarding the service provisioning process for internal use). The rights may be expressed in a natural language style or be formalized. Several instances of the attributes *rightConsumer* and *rightProvider* may be listed. Each of them also needs an identifier to be referencable later on. In our Eco Calculator example the following is stated: *rightConsumer='Service may not be offered for resale.'*

4.6.2 Obligations

Each party involved in the service interaction may have certain obligations regarding the service interaction. Examples could be to have the service consumer provide certain information to the service provider in a timely fashion for her to be able to provide the service. Another example is the obligation to treat certain consumer information confidential. Also, the obligation of payment could be modeled. Obligations are expressed using the attributes *obligationConsumer* and *obligationProvider*. The values can be modeled in natural language or be formalized. Eco Calculator example: *obligationConsumer='Provide complete bill-of-material'*.

4.6.3 Penalties

Penalties might be imposed on any party in the case of violating obligations or rights. Penalties might for example have the form of a fine to be paied to the other party. Legal steps would also be a possible option. Penalties are described in a natural language fashion. Currently, the execution of penalties will be driven by human beings. The automation of this process would be rather complex. In order to describe a penalty it is necessary to model the condition under which a penalty applies using the attribute *penaltyReason* which references rights or obligations. Also, the type of a penalty is modeled using the attribute *penaltyType*. While the penaltyReason attribute contains a formal reference, the penaltyType attribute is modeled using natural language. Example: *penaltyReason=anyRef* and *penaltyType='Payment of fine'*.

4.6.4 Terms of Use

The rights, obligations, and penalties may be described in a formal way using expressions from the legal domain. This information may be represented via electronic Terms of Use (ToU). The ToU is a complex document which may be referenced through a URI using the attribute *touURI*. This approach may be useful in the case that a service provider has a fairly complex number of rights and obligations related to a service and it may thus be tedious to list all of them in the service description.

4.7 Quality of Service

The term quality of service is often used in the domain of computer networks. It describes a number of different quality attributes. Software quality is described as "...the degree to which software possesses a desired combination of attributes..." by [22]. Many of these attributes are relevant for describing the overall quality of a service. It is important to note that it is not suitable to judge the overall quality of a service using a single quality attribute. It is rather the combination of different attributes that provides a holistic picture of the service.

In this section a number of these attributes are outlined and their relevance is explained to service description. Our selection of quality of service attributes is based on [2] which presents a taxonomy for service quality attributes.

4.7.1 Performance

The performance of a system is defined as "...the timeliness of the service delivered by the system..." by [2]. Important aspects of performance which are relevant for this work are response time of a service (which replaces the latency attribute found in [2]) and its capacity.

Response time describes the service's ability to respond to a service request within a specified time frame. A minimum and maximum response time are provided. We distinguish between the initial response time of a service which describes the amount of time that it takes a service to react to a request and the execution response time stating how long it will take the service to fully complete service execution. The following attributes may be described: *responsetimeInitialMin*, *responsetimeInitialMax*, *responsetimeProvisioningMin*, and *responsetimeProvisioningMax*. An additional attribute that can be described is the *jitter*, which states the variation in response time between single service calls. An example for response time would be the following: *responsetimeProvisioningMin=600000* - the minimum response time for service provisioning of the Eco Calculator service is 10 minutes.

The capacity of a service describes how many requests the service can execute during a certain interval without degradation of the response time. The attribute *capacity* may be used to model this information. An example for the Eco Calculator would be: *capacity=100 calls per minute.*

4.7.2 Dependability

The dependability of a system can be described by a number of different attributes such as availability and reliability among others [2]. Dependability focuses on these two attributes. Other attributes such as confidentiality and integrity, which are described in [2] are covered by the security facet of the SDF.

The availability of a service describes when a service is available for provisioning and how often it might be expected to be unavailable (e.g. due to maintenance work). In many situations it is not realistic for a service to be up and running 24 hours 7 days a week. The attribute *availability* describes the percental time value of a service being available for service provisioning (e.g. *availability=99,2*).

The reliability of a service states in how far a service provides its work in the expected way over time. No service can be guaranteed to fully run without problems, but of course problems should occur very rarely. The smaller the probability of failure the better the service's reliability. The attribute *reliability* expresses a percental value of the service providing its benefit properly (e.g. *reliability=97,3*).

4.8 Security and Trust

The areas of security and trust are crucial in the context of service ecosystems. Users will only make use of a service if security and trust are guaranteed, assuring that they are able to reach their business goals with a limited risk of damage. The goal of security measures is to ensure the confidentiality and integrity of information and processes [2]. Trust is concerned with a service's overall reputation but also has a strong emphasis on the payment procedure.

4.8.1 Security

The security goals mentioned above can be reached by taking measures concerning the interaction with a service consumer as well as taking special care with regard to the service implementation during development. With regard to service interaction there are two main aspects to be considered: authentication and encryption. Service internal implementation aspects are not considered here.

Different approaches can be taken to authenticate a service consumer. They include the usage of passwords or authentication through a third party (e.g. the TEXO platform). Authentication is necessary for limiting access to resources and to track the usage of services. It is represented through the attribute *authenticationMechanism*.

The second aspect is the encryption of messages. The channel for communicating with the service in some cases needs to be secured through a suitable encryption mechanism (e.g. Secure Socket Layer). This is modeled using the attribute *encryption*.

The Eco Calculator service has the following settings: *authenticationMechanism=platform*, and *encryption=SSL*.

4.8.2 Trust

Next to security, trust needs to be established between different actors. Trust between the service provider and the consumer can be achieved through a variety of factors such as endorsement, escrow, and insurance of payment [16]. The endorsement of services is a very complex and important topic and thus decided to capture this aspect in a separate facet called rating.

Trust can be achieved through an escrow service during payment. It enables the payment of the due amount to a trusted third party prior to service delivery. Once the payment was made the provider can provide the service, knowing that the payment will be finalized by the escrow service after service provisioning. The advantage for both parties involved in the service interaction is that there is no need to incur in a financial transaction with a possibly unknown party. Using a service identifier, the *escrowService* attribute points to the escrow service that is to be used to support the interaction with the current service.

Another approach for achieving trust is the insurance of payment. Using this approach it is possible to state that due amounts are insured, meaning that failure to provide the service will result in a refund of the paid amount. To model the insurance of payment it is necessary to provide a reference to the *iopService*.

The specific conditions of using the escrow or payment insurance services will be stated in their respective service descriptions. In the Eco Calculator example both services are available: *escrowService=anyUri*, *iopService=anyUri*.

4.9 Rating

A rating enables a potential service consumer to get a view on how the service is seen from a community perspective. User rating is a representation of the overall impression the service made on a number of users. Each user rating is a subjective view on the service. Expert test ratings provide a subjective view on the service from an expert perspective. On the other hand, a certification would provide a rather neutral view on a service provided by a third party. Certifications could be issued from TEXO as well as from standardization institutes.

4.9.1 Community Feedback

This attribute represents aggregated values from existing user opinions about services. Two types of feedback have to be considered. Firstly, *explicit community feedback* and, secondly, *implicit feedback*. The former one indicates that users of a specific service disclose the relation between their service expectation and the perceived outcome of this service. This relation can be expressed in two different ways. *Unstructured Feedback* is found in form of natural language in web forums, community portals, and on the service platform itself. This feedback needs to be collected, and to be computed such that a single value shows the community's opinion about services. *Structured feedback*, on the other hand, is much more easy to collect. Conceivable is a scale with a range from one to ten, where ten is the best rate, to rate a service in whole, or to rate specific parts of the service, such as the way security is handled, or how good the service's availability is. This explicit feedback allows a fine granular filtering of services. For example, a potential service consumer looks for a service which has a strong point on security, but is less interested in the service response time. It is possible to use this preference profile for service discovery. *Implicit rating*, however, tells how often a service was used, and when a service was used the last time, etc. This information must be provided by the service platform. For the service description framework, this attribute reflects users' opinions about the service. However, this attribute is not intended to be provided by service providers. Another party must be involved to collect, calculate, and aggregate the community feedback. The calculated feedback serves as a decisional base for potential service consumers. Important to note is that this attribute of the service description framework is still under development and research. Thus no final comment about the metric can be made, yet. For example, an explicit structured feedback regarding the security for the Eco Calculator service could be *8/10 (5 votings from 100 usages in total)*. Community feedback is important for a service delivery platform to establish trust between service providers and service consumers. Poor services are exposed rather quickly.

Additionally, a rating can support potential service consumers in their decision on which service to select from different alternatives.

4.9.2 Expert Test Rating

This attribute represents a rating from autonomous parties which are experts on the service domain. For the service description framework, potential service consumers might consult the Expert Test Rating to decide whether to use the service or not. The expert test rating is determined by thorough tests, where domain-specific criterias are applied to services and then, depending on the performance, rated. This attribute may be represented via a scale of values ranging from a minimum to a maximum value (e.g. scale from 1 to 10 as described before). For example, an expert test rating for the Eco Calculator service could be *8/10*. The Expert Test Rating is of importance for potential service consumers who do not want to base their decision whether to use a specific service on non-experts, such as the community feedback.

4.9.3 Certification

This attribute represents a certificate issued by trusted institutes or by the platform itself. For the service description framework, this attribute tells whether a service is certified by a known and trusted party. This party issues a certificate in case one or more requirements regarding services are met. An analogous concept is the certification for secure websites. The certificate is represented with a formal system or a common standard, such as the *X.509* [4]. A certificate might establish trust between service providers and service consumers.

5 Conclusion and Future Work

A service description framework for service ecosystems was motivated and existing approaches were discussed. Finally, a service description framework was presented, which was exemplified with the Eco Calculator Service. The framework aims at the full service lifecycle. When a service is created the attributes of the framework will help to describe the service. This will provide a sound base for service discovery, enabling service consumers to more easily find a suitable service according to their needs [20]. The framework will also serve as a basis for monitoring single services [25] and processes (service aggregations).

Future work includes the evaluation of the proposed framework and refine it based on further requirements for service ecosystems. The evaluation will be twofold: on the one hand a survey regarding strength and weaknesses of the framework with experts from the business and web service domains will be conducted and on the other hand the Eco Calculator service will be implemented as a more practical evaluation. In addition, the framework is already used within the Inter-enterprise Service Engineering (ISE) Methodology [14]. Furthermore, the service description framework will be formalized as a meta model in order to serialize and exchange service descriptions. Tools will be created to specify requirements toward services from a service consumer perspective as well as to describe existing services from a service provider perspective [20].

Acknowledgements

Parts of this project were 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

[1] The Dublin Core Metadata Initiative. http://purl.oclc.org/dc/, 1998.

[2] Mario Barbacci, Mark H. Klein, Thomas A. Longstaff, and Charles B. Weinstock. Quality attributes. Technical Report ESC-TR-95-021, CMU, 1995.

[3] Alistair Barros, Marlon Dumas, and Peter Bruza. The move to web service ecosystems. *BPTrends*, 2005.

[4] CCITT (Consultative Committee in International Telegraphy and Telephony). *Recommendation X.509: The Directory—Authentication Framework*, 1988.

[5] David W. Cearley, Jackie Fenn, and Daryl C. Plummer. Gartner's positions on the five hottest it topics and trends in 2005. Gartner Research Report, 5 2005.

[6] Roberto Chinnici, Hugo Haas, Amelia A. Lewis, Jean-Jacques Moreau, David Orchard, and Sanjiva Weerawarana. Web services description language (wsdl) version 2.0 part 2: Adjuncts. W3C Recommendation, 6 2007.

[7] Luc Clement, Andrew Hately, Claus von Riegen, and Tony Rogers. Uddi version 3.0.2. UDDI Spec Technical Committee Draft, 10 2004.

[8] Friedhelm Hausmann and Thomas Einsporn. ecl@ss - the leading classification system. In Armin B. Cremers, Rainer Manthey, Peter Martini, and Volker Steinhage, editors, *GI Jahrestagung (2)*, volume 68 of *LNI*, pages 387–389. GI, 2005.

[9] Martin Hepp, Jörg Leukel, and Volker Schmitz. A quantitative analysis of product categorization standards: content, coverage, and maintenance of eCl@ss, UNSPSC, eOTD, and the rosettanet technical dictionary. *Knowl. Inf. Syst*, 13(1):77–114, 2007.

[10] International Organization for Standardization. Codes for the representation of names of countries, 1981.

[11] International Organization for Standardization. Codes for the representation of currencies and funds. ISO 4217, August 2001.

[12] International Organization for Standardization. Data elements and interchange formats — information interchange — representation of dates and times. ISO 8601, December 2004.

[13] Christian Janiesch, Rainer Ruggaber, and York Sure. Eine Infrastruktur für das Internet der Dienste. HMD - Praxis der Wirtschaftsinformatik (45:261), 2008, pp. 71-79, June 2008.

[14] Holger Kett, Konrad Voigt, Gregor Scheithauer, and Jorge Cardoso. Service Engineering for Business Service Ecosystems. In *Proceedings of the XVIII. International RESER Conference*, 2008.

[15] Inka C. Moerschel and Hendrik Hoeck. Grundstruktur für die Beschreibung von Dienstleistungen in der Ausschreibungsphase. Beuth Verlag GmbH, 2001. Ref.Nr. PAS1018:2002-12.

[16] Justin O'Sullivan. *Towards a Precise Understanding of Service Properties*. PhD thesis, Queensland University of Technology, 2006.

[17] THESEUS Pressebüro. TEXO - Business Webs im Internet der Dienste. http://theseus-programm.de/scenarios/de/texo, March 2008.

[18] Arthur Ryman Sanjiva Weerawarana Roberto Chinnici, Jean-Jacques Moreau. Web services description language (wsdl) version 2.0 part 1: Core language. W3C Recommendation, 6 2007.

[19] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Christina Feier, Christoph Bussler, and Dieter Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005.

[20] Gregor Scheithauer. Process-oriented Requirement Modeling for the Internet of Services. In *Proceedings of the 1st Internet of Services Doctoral Symposium 2008 (I-ESA)*, volume Vol-374, Berlin, Germany, March, 25 2008.

[21] Software Engineering Standards Committee of the IEEE Computer Society USA. IEEE Guide for Software Requirements Specifications 830-1998, 1998.

[22] Software Engineering Standards Committee of the IEEE Computer Society USA. Ieee standard for a software quality metrics methodology. IEEE Standard 1061-1998, 12 1998.

[23] Laurie Sullivan. Analysts: Prepare for soa shift. ChannelWeb article, 7 2007.

[24] W3C HTML Working Group. XHTML The Extensible HyperText Markup Language. W3C Recommendation 1.0, W3C, August 2002. Second Edition.

[25] Matthias Winkler. Service Description in Business Value Networks. In *Proceedings of the 1st Internet of Services Doctoral Symposium 2008 (I-ESA)*, volume Vol-374, Berlin, Germany, March, 25 2008.

Bamberger Beiträge zur Wirtschaftsinformatik

Stand November 26, 2008

Nr. 1 (1989) Augsburger W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)

Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG

Nr. 3 (1990) Augsburger W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)

Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)

Nr. 6 (1991) Augsburger W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Nr. 7 (1991) Augsburger W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender

Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung

Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell

Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)

Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffsystem des Semantischen Objektmodells

Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell

Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organsiation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen

Nr. 15 (1992) Schwab H. J.: EISREVU-Modellierungssystem. Benutzerhandbuch

Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip

Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Bürovorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells

Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach

Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994

Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994

Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen

Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems

Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle

Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen

Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme

Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes

Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995

Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach

Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit

Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse

Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger

Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?

Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -

Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme

Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996

Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996

Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze

Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997

Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik ’97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997

Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunkheft ComponentWare, 1997

Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997

Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2nd Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998

Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung

Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998

Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement“. Aachen, September 1997

Nr. 47 (1998) Sinz, E.J.:, Wismans B.: Das „Elektronische Prüfungsamt“. Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998

Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Respresentation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering

Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems

Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460

Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS'99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)

Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS'99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)

Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule" im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999

Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999

Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburger W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.

Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur

Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000

Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.

Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.

Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001

Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001" im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Note: The title of our technical report series has been changed from *Bamberger Beiträge zur Wirtschaftsinformatik* to *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* starting with TR No. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

Nr. 61 (2002)	Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.
Nr. 62 (2002)	Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002
Nr. 63 (2005)	Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions
Nr. 64 (2005)	Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005
Nr. 65 (2006)	Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet
Nr. 66 (2006)	Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006
Nr. 67 (2006)	Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006
Nr. 68 (2006)	Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation
Nr. 69 (2007)	Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework . February 2007
Nr. 70 (2007)	Thomas Meins: Integration eines allgemeinen Service-Centers für PC-und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien. Februar 2007
Nr. 71 (2007)	Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises. März 2007
Nr. 72 (2007)	Michael Mendler, Gerald Lüttgen: Is Observational Congruence on μ-Expressions Axiomatisable in Equational Horn Logic?
Nr. 73 (2007)	Martin Schissler: to be announced

Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 74, Bamberg University, October 2007. ISSN 0937-3349.

Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second Privacy Enhancing Technologies Convention (PET-CON 2008.1). Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 75, Bamberg University, April 2008. ISSN 0937-3349.

Nr. 76 (2008) G. Scheithauer and G. Wirtz: Applying Business Process Management Systems? A Case Study. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 76, Bamberg University, May 2008. ISSN 0937-3349.

Nr. 77 (2008) Michael Mendler, Stephan Scheele: Towards Constructive Description Logics for Abstraction and Refinement. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 77, Bamberg University, September 2008. ISSN 0937-3349.

Nr. 78 (2008) Gregor Scheithauer and Matthias Winkler: A Service Description Framework for Service Ecosystems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78, Bamberg University, October 2008. ISSN 0937-3349.