# Knowledge Engineering for IT-based Services

**Sören Auer**
(University of Leipzig, Germany, auer@informatik.uni-leipzig.de)

**Patryk Burek**
(University of Leipzig, Germany, burek@informatik.uni-leipzig.de)

**Tonio Grawe**
(Advicio Ingenieurbüro Munich, Germany, tonio.grawe@advicio.com)

**Abstract:** A formal product model contains all the information (structured and formalized) to systematically reproduce a specific product (as economic asset). There are several approaches for formalizing product model information in the old economy (for example CAD/CAM-based product models in discrete parts manufacturing). The service sector evolved to the most important sector in all developed economies. Knowledge plays a crucial role for delivering many services. For complex, IT-based service products high in variants (such as insurances, IT outsourcing or public administration services) existing approaches are not suitable but formalization is desired (e.g. it allows easier rendering, export or trade of such products). This paper elicits a possible strategy for defining formal product models for knowledge-based services using knowledge representation and semantic web technologies.

**Key Words:** Service Engineering, Knowledge Representation, OWL, Product Model

**Category:** Information Storage (H.3.2), CAE (J.6), Model Development (I.6.5)

## 1 Introduction

Support of the service lifecycle by standards and applications is gaining increasing importance. Variant rich services (as characterized in [Fig. 1]) are highly modular, complex and sometimes even built-to-order thus requiring sophisticated service modelling and knowledge representation strategies. Product models for tangible products are well known (like EXPRESS/STEP [Anderl, Trippner 00]) but their representation languages are not sufficient for service products [Auer, Fähnrich 03a].

OWL [Dean, Schreiber 03] is a W3C knowledge representation standard which can be estimated to consolidate fragmented efforts to represent knowledge. OWL-S is a W3C interest group and an approach to base WSDL web service descriptions on OWL [OWL-S 03], which lacks (or does not attempt to) support for non-web technology related service description components and seems to be restricted to the description of technology related information only. OWL-S is not designed to integrate concepts concerning e.g. marketing or legal information.

The aim of the present paper is to propose an OWL-based service product description language (SPDL), whose purpose is to capture all knowledge required in the lifecycle of variant services, which also interoperates as canonically as possible with WSDL, seamlessly integrates into Tim Berners-Lee's semantic pyramid and which may sustain product lifecycle and extend product flexibility of services.

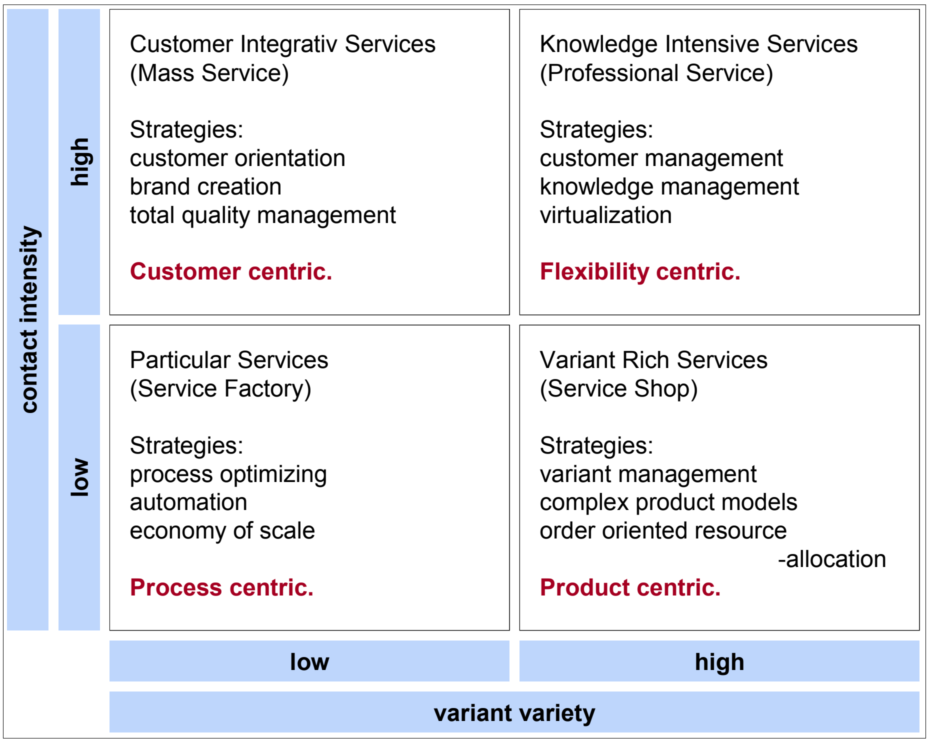


*Figure 1: Service Typology*

## 2 Ontology-based Service Modelling

Ontologies enable effective representation of domain knowledge. An ontology consists of concepts which are formally described by properties they possess, relations (like inheritance) between them and restrictions on these properties ([Guizzardi et al. 02]). Some crucial questions concerning the ontology underlying SPDL are answered in this chapter, a more formal specification of the ontology follows in [Section 3].

### 2.1 What is the domain that the ontology will cover?

SPDL is intended to support the life cycle of product centric variant rich IT-based services. In parallel to refining SPDL, two domain specific ontologies for services are developed and are on the way to proving the advantages of SPDL. These are:

- IT Infrastructure services, e.g. webhosting, internet access, data center operations
- Insurances services, e.g. life insurance, pension fund management, collective insurances

### 2.2 For what are we going to use SPDL?

SPDL represented knowledge about product centric services shall be used in the following phases of product life cycle:

### 2.2.1 Design / Modelling

Services are developed mostly ad hoc [Glanz, Meiren 02]. Determined strategies or technologies for a coordinated development of new services and their modelling (e.g. CAD/CAM for tangible products) do not exist. SPDL supplies a framework of important concepts common to product centric service products and thus, in conjunction with more domain specific service ontologies, supplies a crucial tool for service development. One of the main principles of SPDL is ensuring service products to be modelled in a recursive modular manner – which enables high reusability [Faehnrich, Grawe 03]. New products may thus be more easy constructed and simulated out of already existing components.

### 2.2.2 Simulation

SPDL already includes important concepts and strategies for simulation of services. These include:

- Resource planning
- Pricing
- Market analysis
- Contract validation
- Localization

### 2.2.3 Implementing, Rendering of Product Centric Service Products

At present, services at the IT level are implemented in a quite homogeneous way (different programming languages, databases, component-/middleware), which is probably utopian to change in near or even middle future. A problem, which, on the other hand, may and should be solved and is addressed by SPDL, is the separation of knowledge and functionality operating on this knowledge. Such a strategy avoids product information being widespread in software applications across a company.

Services are often rendered using many other services or service components sometimes provided by suppliers (companies, governmental organisations, and departments). Usage of common vocabularies improves interoperability dramatically. Further cooperation interactions (e.g. web-service integration) may be configured automatically.

### 2.2.4 Customer Interaction in Sales and Support

Due to the fact that we are considering most services where the interaction with customers is highly automatiseable - ways of a formal description of interactions with customers are needed (WSDL Grounding provides this for machine-machine interaction in web-services).

### 2.3 For what types of questions does the information in SPDL provide answers?

SPDL enables a formalized way to describe dependencies between services or service components, and thus may assist or automatically provide strategies for a choreography between them. This is especially important when service products should be available in a diversity of service variants, which may be managed mode easily (variant management).

Essential mechanisms of service quality assurance are service level agreements – an aggregation of service levels or usage parameters. The more complex a service product, the more expensive is the calculation of valid ranges for service levels. SPDL-based service products may thus propagate service levels.

One of the main visions for developing SPDL is an *automatic configuration of IT infrastructure and software applications* needed to render SPDL represented services. For this vision to become reality, knowledge repositories have to be developed which are as easily useable as existing middleware applications (e.g. databases).

### 2.4 Who will use and maintain SPDL?

The target audience for SPDL is the service industry (especially IT service providers and insurances for which SPDL-based domain ontologies are being developed), software developers and researchers. At the moment, SPDL is being maintained by the authors and a workgroup inside the ServCASE project (http://www.servcase.de) it is planned that SPDL will be maintained by a larger community including researchers, standardisation institutes and governmental organisations.

## 3 Service Product Description Language – a Blueprint

In addition to machine processability (inference, reasoning) one of the main advantages of ontology-based knowledge representation approaches is to „speak a common language“ and to share a common vocabulary and a common understanding of its terms. In this section the ontology underlying the SPDL is introduced.

### 3.1 Methodology

For the development of the ontology underlying the SPDL, the general assumptions of Methontology [Gomez-Perez et al. 97] have been used. Additionally, some of the Knowledge Engineering Methodology suggested by Brachman for Frame-based systems have been used as well as the techniques used in object-oriented modelling. The structure of the current chapter reflects the adopted methodologies but the space limitations prevent us from introducing the full documentation suggested by Methontology.

### 3.2 Specification of the underlying Ontology

In addition to the informal description of SPDL given in [Section 2], we summarize in this Paragraph the general specification of the model according to Methontology.

**Ontology's primary objective:** Provide a common vocabulary for the modelling, description, design and construction of services.

**Level of formality:** High.

**Scope:** Description of IT enabled service products with a high degree of formalizable customer interaction.

**Sources of knowledge:** Existing product modelling approaches (e.g. [Anderl, Trippner 00]), existing domain specific service product models (e.g. [CIM]).

### 3.3 Conceptualisation of the underlying Ontology

In [Fig. 2] we provide a semi-UML graph representing the ontology underlying SPDL. Concepts presented in the graph are discussed below. For each concept we provide: the name of the concept, eventual synonyms, informal explication, attributes, roles, eventual subconcepts, and examples.

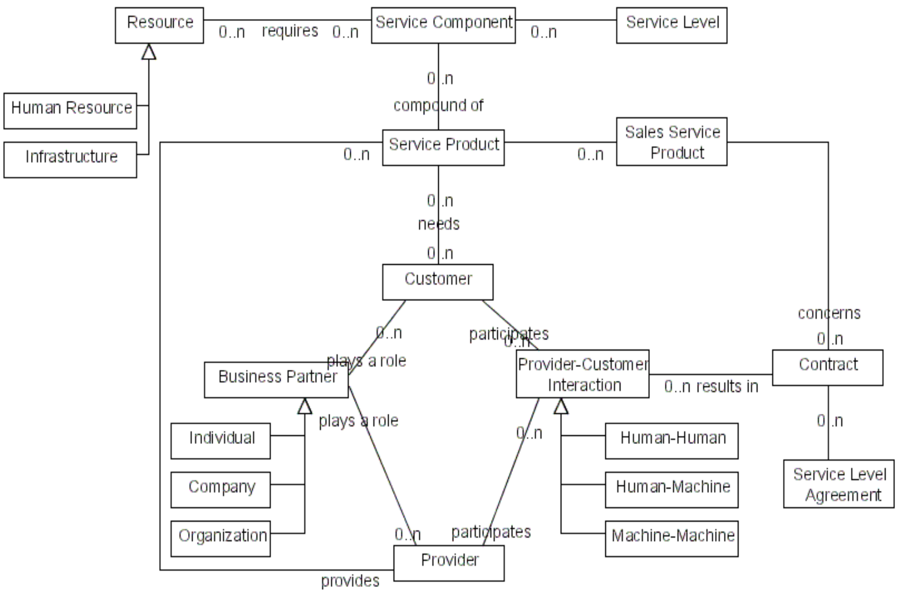


*Figure 2: Ontology underlying SPDL*

*Concept's name:* **Business Partner**
*Synonyms:* Business Party
*Informal specification:* Business Partner is any entity that can be involved in the process of service providing/receiving as a potential customer, customer, user or provider;

*Attributes:* Name, Address etc.
*Roles:* Each Business Partner **may** need **one or many** Service Products; Each Business Partner **may** play a role of **one or many** Customers. Each Business Partner **may** play a role of **one or many** Providers;
*Subconcepts:* Individual, Company, Organization
*Examples:* Any legal organization, company or person.

*Concept's name:* **Customer**
*Synonyms:* User, consumer
*Informal specification:* Customer is a role of the business partner who receives the service and is involved in the customer-provider interactions which is aimed to result in a contract.
*Roles:* Each Customer **must be exactly one** Business Partner. Each Customer **may be** involved in **one or many** Provider-Customer Interactions.
*Attributes:* Profile of interests, billing address, etc.
*Examples:* Customer profile an online shop.

*Concept's name:* **Provider**
*Synonyms:* Carrier, supplier
*Informal specification:* Provider is a role of the business partner who provides the service and who is involved in the customer-provider interactions which is aimed to result in the contract.
*Roles:* Each Provider **must be exactly one** Business Partner, Each Provider **may be** involved in **one or many** Provider-Customer Interactions
*Attributes:* Profile of activities, references, recommendation rank, etc.
*Examples:* Data center carrier.

*Concept's name:* **Provider-Customer Interaction**
*Informal specification:* Provider-Customer Interaction is a single element of the process by which a customer and a provider communicate and which may result in setting a contract between them.
*Roles: E*ach Provider-Customer Interaction **must** involved **exactly one** Customer and **one** Provider, Each Provider-Customer Interaction **may** result in **one** Contract.
*Attributes:* Date, grounding, etc.
*Subconcepts:* Human-Human, Human-Machine, Machine-Machine
*Examples:* Communication by email, transaction using a web site or web service.

*Concept's name:* **Contract**
*Informal specification:* Contract is an establishment of a legal relation between a Service Provider and a Service Customer. The Contract is a result of the sequence of interactions between a Customer and a Provider. The subject of the Contract is a Sales Service Product assigned with an Service Level Agreement.
*Roles:* Each Contract **must be** a result of **one or many** Provider-Customer Interactions. Each Contract **must** concern **exactly one** Sales Service Product. Each Contract **must** specify **exactly one** Service Level Agreement;
*Attributes:* Begin date, End date and others.

*Concept's name:* **Service Level Agreement**
*Informal specification:* A Service Level Agreement is essential for a Contract and combines concrete values for one or many Service Levels.
*Roles:* Each Service Level Agreement **may be** included in **one or many** Contracts.

*Concept's name:* **Sales Service Product**
*Synonyms:*
*Informal specification:* Sales Service Product is a Service Product available for purchase. It in-

cludes all marketing and sales relevant information of a Service Product.
*Roles:* Each Sales Service Product **may be** a subject of **one or many** contracts; Each Sales Service Product **must** have **exactly one** underlying Service Product;
*Attributes:* price, warranty, exclusions of legal liability, customer obligations

*Concept's name:* **Service Product**
*Synonyms:* Service
*Informal specification:* Service Product is a process supplied by a Service Provider that satisfies the Service Customer needs. It may be decomposed into more basic processes called Service Components. Service Product may be purchased as the Sales Service Product.
*Roles:* Each Service Product **must be** provided by **exactly one** Provider**;** Each Service Product **may** satisfy the needs of **one or many** Customers**;** Each Service Product **must be** compound of **one or many** Service Components; Each Service Product **may** underlay **one or many** Sales Service Products;

*Concept's name:* **Service Component**
*Informal specification:* Service Component is the most basic component to which a Service Product can be decomposed.
*Roles:* Each Service Component **may** have **one or many** Service Level; Each Service Component **may be** a part of **one or many** Service Products;

*Concept's name:* **Service Level**
*Informal specification:* Threshold (with unit and optional aggregation rules) characterizing crucial properties of a Service Component.
*Roles:* Service Level **may** characterize **one or many** Service Components**;**
*Attributes:* Value, unit, aggregation rules.
*Examples:* Availability, average availability, bandwidth, storage capacity, processor time, reaction time

*Concept's name:* **Resource**
*Informal specification:* In contrast to Service Components which may be reused as often as needed, Resources may be spend if consumed „to much".
*Roles:* A Resource may provide one or many Features and **may** be consumed by **one or many** Service Components**;**
*Attributes:* Initial availability value, availability percentage and unit.
*Subconcepts:* Human Resource, Infrastructure Resource
*Examples:* Server administrator, Linux server.

The complete SPDL ontology is currently provided for download at the SPDL project website [SPDL 04] in OWL format. The OWL format allows the combination of modelling knowledge and concrete instances (reifications) in a single consistent data structure. Nevertheless other widespread modelling languages such as e.g. UML are planned to be supported as well.

# 4 Product Models in the IT Services Domain

## 4.1 Known Work and Experiences

With the term *IT services* we mean the operations of information technology for the use by others. In the domain of IT service providers there is a growing demand for service specifications and product data management. The increasing maturity of the market players and the competitive environment are the reasons for the interest in concepts to retrieve their business knowledge. However, all known approaches for service modelling focus on the IT infrastructure, lacking other aspects like human resources, processes and business relations. Examples for this kind of technology-oriented service specifications are [SIM-S] for the management of storage services or [CIM] for modelling IT devices.

A consortium of the telecommunications industry is developing a promising domain ontology, called *Shared Information and Data Model* [SID]. SID is not suggesting a formal representation language but it defines various concepts. As IT and telecommunications are related to each other, evidently SID was to be evaluated in the domain of IT services. We have modelled IT services based on the SID concepts, using the Protégé ontology editor and OWL as representation language.

### 4.2 Introduction of a Platform Strategy

Product models have to represent an extensive knowledge about the services: How is the service specified, what are the processes, what resources are needed, how to communicate with customers, and how to calculate the price of service products. As a result, the product models become complex. To reduce the effort of modelling and maintenance we have introduced a component-based approach.

The term "platform strategy" is familiar from the automobile industry. It refers to the standardisation of components. The idea of re-using components in different models has helped automobile companies to optimise production and logistics. It is also possible to successfully adopt this approach for IT concerns. The idea now is that, through standardisation, service components are, wherever possible, only provided once; they can then be used in as many products as possible. This means that a new service product can be developed rapidly and cost-effectively, because a large proportion of the new product consists of components that already exist.

For the introduction of a platform strategy for IT Service Providers we have developed the *Service Innovation Methodology*, a straightforward approach: (1) Complex, IT-based services are first partitioned into their components (decomposition). (2) The components are then standardised to set free potential for consolidation and re-use. (3) A taxonomy for IT service components is introduced next. (4) Then the knowledge required to configure component-based services is mapped with the aid of knowledge representation methods. (5) B2B interfaces are finally set up, so that value creation networks between IT service providers can be linked.

### 4.3 A Taxonomy for IT Services

[Fig. 3] shows an overview of the taxonomy we suggest for IT service components. The taxonomy provides a semantic knowledge about the component classes. Each component class provides a defined functionality that is further specified by certain attributes.

The inheritance mechanism keeps the model consistent and significantly accelerates the process of modelling, as common properties have to be defined just once. For example we define the attribute „availability“ at the root class as all service components share it. „Response times“ is an attribute that is significant for applications so it is defined in the „Application Service“ class. Nevertheless, the defined attributes might have to be treated differently in each class. To use the availability attribute, which is usually measured in a percentage value (like 99% of service operation time), the conditions for availability respectively unavailability also need to be specified. As it turned out, this required us to go very much into detail.

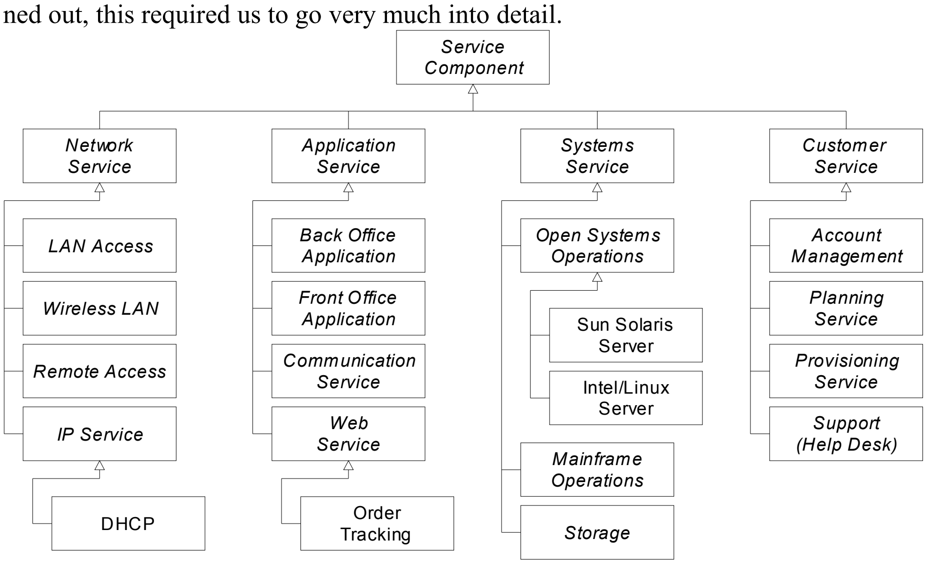


*Figure 3: Taxonomy of IT Service Components.*

We want to annotate here that the service components from the taxonomy must not be mixed up with infrastructure resources (e.g. hardware devices). The component „Sun Solaris Server“ indeed uses such a machine, but the service is actually to operate the server, including all the processes, operations staff, and possibly even a financial service for leasing the hardware.

### 4.4 Composition of Service Products (Product Configuration)

The components form the taxonomy are the building blocks that are used to compose the service products. The starting point is a service product specification that defines what functionality the product requires. To fulfil these requirements, service components are linked to the product. Components might require functionality from other components. This way, we built a cascaded model like the workspace example in [Fig. 4].

The link between the components is rather loose, as a shared component might provide its functionality to several other components and products. In the example this is the case for the network, which is obviously used by others, too. Although it is a loose link, quality and other properties of components affect the consumers of that functio

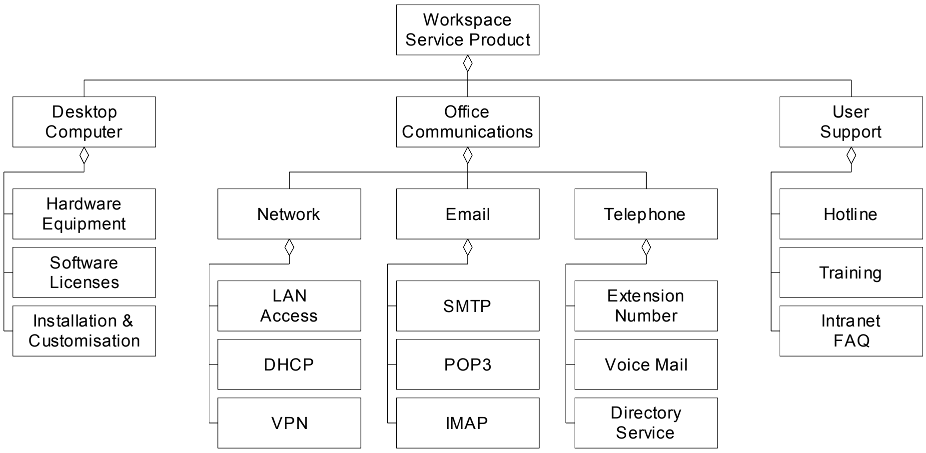


*Figure 4: Exemplary composition of a service product called „workspace“ which provides IT and communication facilities for employees.*

nality. In the workspace example, unavailability of the network results in a decreased service quality of the workspace service product. These interdependencies are the challenge in the configuration process. A tool that aids the user in this configuration process is being worked on.

### 4.5 Lessons Learned and Adoption to SPDL

The introduction of a platform strategy and the loose coupling of service components seems to be a future-proven approach. We expect service providers to build partnerships with others and together offer high-value services.

The taxonomy is a good approach but it does not seems realistic to build one taxonomy that is valid for all kinds of service providers. It is rather the component models itself, which must provide information about their functionality. This independency of one general taxonomy allows service providers to build their own, that just contains the components that they need.

In this project we have so far limited our scope to IT services. However, the step to IT-based services in general is razor-thin. What used to be an application service, is now a flight reservation service with some additional business logic or a community portal with some additional content. Furthermore, IT services may enclose service of other domains, like financing of infrastructure investments. SPDL's approach to have a domain independent representation language, accomplishes the need to mix IT and with other services to build innovative service products.

## 5 Discussion and Future Developments

Detailed analysis of the OWL standard on which SPDL is based showed the need for integration of a consistent framework to enable numerical calculations providing a straightforward mechanism of modelling UML like material relations.

Considering the need for providing well grounded ontological foundations for our model, we want to integrate the categories of SPDL with the categories of the upper level ontology of GOL - General Ontological Language [Guizzardi et al. 02], currently in development by the Onto-Med Research Group.

Common service product related transactions between customer and provider (e.g. obtaining product information, supplying information relevant to render a product, signing a contract) should be made available as automatically as possible. In cooperation with OWL-S it should be possible to implement an application framework which grabs the relevant information out of an OWL repository and supplies the appropriate transactions as WSDL services.

In cooperation with the BMBF funded project ServCASE, a service development platform is in development, in which recent results will be integrated.

## References


[Anderl, Trippner 00] Anderl, R.; Trippner, D. (Hrsg.): „STEP Standard for the Exchange of Product Model Data”. B. G. Teubner, 2000.

[Auer, Fähnrich 03a] Auer, S.; Fähnrich K.-P.: “A Strategy for Formal Service Product Model Specification”. Human Computer Interaction, International Proceedings. Lawrence Erlbaum Associates, 2003.

[Auer, Fähnrich 03b] Auer, S.; Fähnrich K.-P.: “Product Models in Service Engineering”. Proc. 17th International Conference on Production Research.

[Brachmann et al. 90] Brachman, R.J., McGuinness, D.L., Patel-Schneider, P.F., Resnick, L.A., & Borgida, A. 1990. Living with CLASSIC: When and How to Use a KL-ONE-Like Language, in Sowa, J., Ed. Formal Aspects of Semantic Networks. Morgan Kauffman.

[CIM] Desktop Management Task Force: "Common Information Model". http://www.dmtf.org/standards/cim, 2004

[Dean, Schreiber 03] Mike Dean, Guus Schreiber, eds.: „OWL Web Ontology Language Reference“. W3C Proposed Recommendation, 2003, http://www.w3.org/TR/owl-ref.

[Faehnrich, Grawe 03] Fähnrich, K.-P., Grawe, T.: „Systematisches Entwickeln von IT-Dienstleistungsprodukten“. In: Bernhard et al. (Eds.): Strategisches IT-Management, Düsseldorf, 2003

[Glanz, Meiren 02] Glanz, W.; Meiren, T. (eds.): “Service research today and tomorrow”. IAO Stuttgart, 2002.

[Gomez-Perez et al. 97] Gomez-Perez, M. F. A. and Juristo, N. (1997). *Methontology: From ontological arts towards ontological engineering*. Proc AAAI Spring Symp. Series, AAAI Presss, Menlo Park, california, 1997, pp 33-40

[Grawe 03] Grawe, T.: Geschäftsstrategie für IT-Dienstleister: Kundenorientierung. In: IT Management, Sep 2003, p. 44-50

[Grabowski et al. 93] Grabowski, H.; Anderl, R., Polly, A.: „Integriertes Produktmodel“. DIN Deutsches Institut für Normung e.V., Berlin, Wien, Zürich 1993.

[Guizzardi et al. 02] Guizzardi, G.; Herre, H.; Wagner, G.: „On the General Ontological Foundatioin of Conceptual Modeling"; ER 2002, LNCS vol. 1937, p.97-112

[López et al. 99] Mariano Fernández López, Asunción Gómez-Pérez, Juan Pazos Sierra, Alejandro Pazos Sierra. 1999. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems, January/February 1999 (Vol. 14, No. 1).

[OWL-S 03] OWL Services Coalition: „OWL-S: Semantic Markup for Web Services". 2003, http://www.daml.org/services/owl-s/1.0

[SID] Telemanagement Forum: "Shared Information/Data (SID) Model". TMF, GB922, 2002

[SIM-S] Storage Networking Industry Association (SNIA): "Storage Management Initiative Specification". http://www.snia.org/smi/home, 2004

[SPDL 04] Auer, S. et al.: „SPDL Project website", http://www.informatik.uni-leipzig.de/~auer/spdl.

[Strobl 98] Strobl, G.: „Entwicklung und Wiederverwendung wissensbasierter Produktmodelle auf der Grundlage formaler Ontologien". Herbert Utz Verlag, 1998.