

Ontology-based representation of compliance requirements for service processes

Rainer Schmidt
Aalen University
Beethovenstraße 1, 73430 Aalen, Germany
Rainer.Schmidt@htw-aalen.de

Christian Bartsch
Information Process Engineering (IPE)
Research Center for Information Technologies (FZI)
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
bartsch@fzi.de

Roy Oberhauser
Aalen University
Beethovenstraße 1, 73430 Aalen, Germany
Roy.Oberhauser@htw-aalen.de

Abstract: Service processes are becoming increasingly essential in modern economies as traditional, production-oriented industries decline. When comparing service processes to standard business processes, a major distinction is that the quality of their result, i.e., the service produced, cannot be measured in advance. Therefore, the compliance of the service process with quality standards plays an important role in convincing the customer that the services rendered will result in the quality specified. However, the check for compliance is still a tedious task. To address this situation, an ontology-based approach for representing service processes and checking their compliance is proposed. It is based on two ontologies: one to represent the service processes and the other to store the compliance requirements. The process representation ontology uses three so-called views to appropriately represent the service processes. The ontology for storing the compliance requirements differentiates syntactic, semantic and pragmatic requirements.

1 Introduction

Service processes are processes that produce services. Their significance is increasingly growing in modern economies as traditional, production-oriented industries decline. An important difference of service processes when compared to standard business processes is that the quality of their outcome, i.e., the service produced, cannot be measured beforehand [BuSc06]. This is due to the fact that services cannot be produced in advance because they cannot be stored. On the contrary, material products can be tested before they are used since it is possible to store them. Therefore, it is vital to convince the

potential customer that the services rendered will result in the quality specified [BuSc06]. The quality of a service to be provided can only be estimated by checking if the process used to provide it is in compliance with quality standards, such as ISO 20000 [ISO20000]. Standards such as ISO 20000 are a kind of abstract specification which define compulsory elements and structures of a process necessary to provide the services in the quality required. However, they do not specify the service processes in full detail because there are different ways to achieve a certain level of service quality. Thus, there is a large divide between the quality standard and the service process, similar to the IT / process divide described in [HLDW05]. As a consequence, the effort to check whether a service process is in compliance with standards such as ISO 20000 is rather high.

Thus this paper provides an approach to appropriately represent compliance requirements and to provide a vision how to reduce the effort for checking the compliance of service processes with quality standards such as ISO 20000. Two ontologies [Grub95] are used. The process ontology is the basis for the ontology-based representation of the service process; the compliance ontology represents the compliance requirements the service process has to fulfill. The paper proceeds as follows: Section 2 shows how to represent service processes using an ontology-based approach. Section 3 defines the representation of compliance requirements in an ontology. The checking for compliance and the implementation is described in Section 4. Section 5 deals with the analysis of related work. Finally a summary and an outlook are given.

2 Process Ontology

As already stated in [HeRo07], a process is more than the mere connection of activities. Thus, it is necessary to reason about the appropriate representation of service processes, because they contain additional elements when compared with standard business processes. Two kinds of additional elements will be identified later on: interactions and resources.

In most business processes the customer is only interested in the outcome of the process but not in the process itself. On the contrary, in service processes, there are many interactions between the service provider and the customer as well as third party service providers. These interactions often have to follow predefined patterns and have to be documented to serve as proof in latter disputes. Thus, it is necessary to appropriately represent these interactions as shown in Figure 1. Interactions connect two activities: one is executed by the service provider and the other by the service client. Between both, a multitude of communication acts may take place which often cannot be fully specified. For example, the clearing of documents requires frequent interactions between the participants. Neither the type of dialog (telephone, e-mail, etc.) nor the interaction frequency can be specified in advance. However, key events, such as the transmission of documents and their clearance can be specified. Therefore, the internal behavior of interaction is defined by event-condition-action rules. They describe the condition to be tested when a certain event occurs and the action to be taken if the condition is met. The starting and ending points of interactions are defined using pre- and post-conditions. The pre-conditions specify the circumstances that need to be fulfilled to start the interaction.

The post-condition specify the results that have to be obtained before the interaction can be regarded as completed.

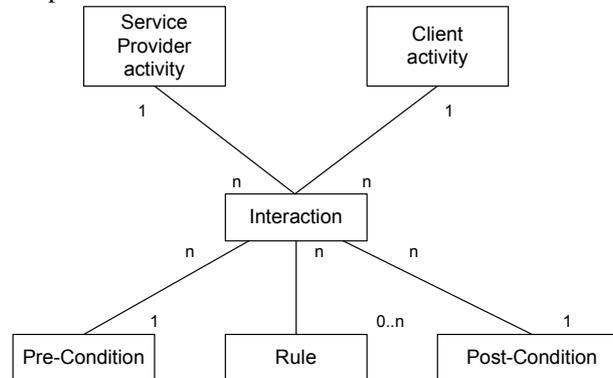


Figure 1: Interactions

Service processes differ from traditional business processes because they extensively use external resources, both from the customer and third party service providers, which have to be appropriately integrated and administered [ZdHe05]. For example, before configuring a customer's computer system, the administrative privileges must have been granted. In addition, if external resources are no longer available but needed for service providing, a procedure to correct these errors has to be started. Finally, customer resources that were used for service providing need to be returned when the service has been completed. In order to properly represent changes in the resource view, adding, changing, and removing resources needs to be a simple process.

The discussion about the proper representation of service processes does not only include the "what" but also the "how", because intermixing independently evolving process elements causes a multitude of side effects. One example for the mixture of views is the "flow dependence" of application programs described by [LeRo97]. Flow dependence means that application programs contain a predefined control flow, making them inflexible to business process changes. To avoid the intermixing of independently evolving functionality, a view-oriented approach is chosen to represent the service processes in an ontology. Views are sets of process elements to mirror aspects of reality evolving completely independent of each other. To appropriately represent service processes, interactions and resources are represented as separate views. Additionally, five basic views are used to organize the process elements common to business processes. They are the functional, operational, control, informational, and organizational view as identified in [Jab194]. The functional view describes the goal of the process. The operational view specifies activities executed during the process. The control view defines the preconditions for certain activities. The informational view specifies the information exchange that takes place between activities. Schema, schema elements, and relations are the elements of the informational view. The organizational view associates roles with activities. As these views have been extensively analyzed in research, e.g., [JaBS97], [Jab194], [BKKR03], they are not discussed here.

An ontology for ontology-based process modeling has to reflect the semantics of the service processes. As a result, the process ontology is organized in three layers as shown in Figure 2, displaying a part of the incident management process of ITIL [ITSM04].

The visual representation follows the suggestions made in [BVEL04]. The layers define the abstraction levels of the ontology. The first layer is called meta layer. It defines the concepts for describing the views in the second layer. Using this approach, extensibility is achieved, allowing easy integration of additional views in the second layer via the concepts of the basic layer. The second layer is called view layer. In the second layer, views and their elements are defined using the concepts defined in the first layer. Seven views are defined in the view layer: functional, operational, control, informational, organizational, interaction, and resource. The concrete process element classes are defined in the third layer, which is called the process layer. The instances of the process element classes are used to create the ontology-based process representation.

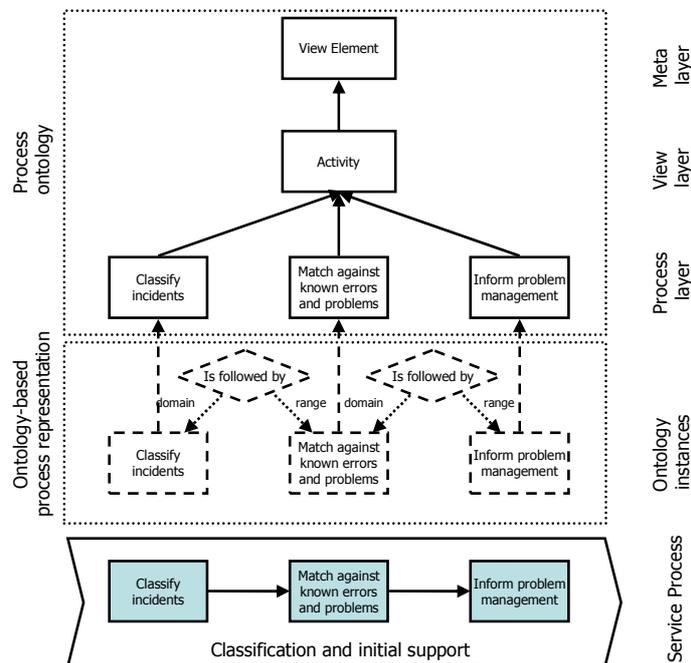


Figure 2: Structure of the ontology

Ontologies are defined using OWL [McHa04]. A one-to-many relationship is used between process element classes in the process layer and the instances in the ontology-based process representation. This is crucial for detecting synonyms and homonyms if a process element appears several times, as shown in Figure 3. The one-to-many relationship permits expressing the semantic identity or non-identity of process elements¹. Synonyms in the ontology-based process representation can be detected because process elements with different names point to the same concrete process element class. Homonyms can be found by detecting that two process elements of the same name point to different process element classes.

¹ Other approaches such as [ThFe06] use a one-to-one relationship which does not support the detection of synonyms and homonyms.

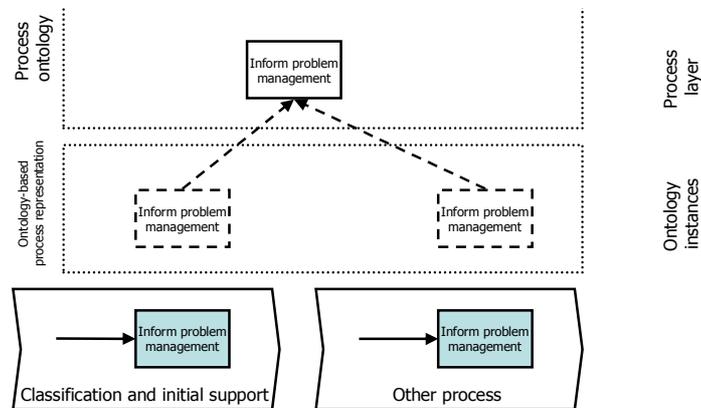


Figure 3: Detection of synonyms and homonyms

The process layer contains not only is-a relationships but also a part-of hierarchy. Is-a relationships allow unambiguously classifying each element of the process model. Part-of relationships allow representing aggregations. For example, it is possible to define the role “Incident Manager” as part of the section “Incident Team”.

3 Compliance ontology

Compliance standards, such as ISO 20000, define objectives that have to be fulfilled by the service processes to assure a certain quality of the services provided. These objectives have a set of supporting requirements, which may be syntactic, semantic, or pragmatic. Syntactic requirements contain rules for the description of the service processes. For example, they postulate that each activity needs to be associated with a role, which is responsible for the activity. Semantic requirements demand the existence of certain objects or structures in the process definition. For example, they postulate the existence of a certain kind of document, such as a service report. Pragmatic requirements describe goals that have to be realized within the process. Syntactic and semantic requirements can be verified automatically or at least semi-automatically. Pragmatic requirements can only be verified manually.

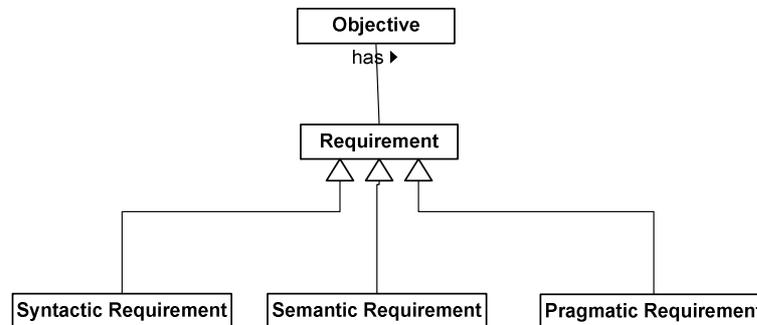


Figure 4: Types of compliance requirements

3.1 Syntactic Requirements

Syntactic requirements are rules defining which model elements have to be used or may be used for the definition of service processes. Therefore, they can be thought of as a “meta model” for the models of service processes. Syntactic requirements can be easily expressed via logic operators. For example, the rule “responsible role required for each activity” can be expressed as follows

$$\text{activity}(a) \Rightarrow \exists r \text{ role}(r) \wedge \text{responsible}(a, r)$$

Syntactic requirements can be directly represented in an ontology. E.g., in order to express that a responsible role needs to be assigned to each activity, a property “responsible” is defined, with activity elements as the domain and roles as the range.

3.2 Semantic requirements

Semantic requirements postulate the existence of certain structures or objects in the service process. Semantic requirements can be further differentiated into assertion requirements, structure requirements and action requirements as shown in Figure 5. Assertion requirements define conditions, which have to be matched. It is not specified who has to assure that the condition is met. For example, ISO 20000 defines that “service levels shall be monitored”. This can be expressed formally as follows

$$\text{servicelevel}(s) \Rightarrow \text{monitored}(s)$$

Action requirements define actions to be made. An action is comprised of a role, responsible for the action, a verb that describes the action and the object of the verb. For example, ISO requires that “management shall conduct reviews”. The responsible role is “management”, “conduct” is the verb and “reviews” are the object.

Structure requirements define structures, which have to exist in the process to be compliant with the objective. Structure requirements consist of structure elements, which may be nested. Structure requirements define structures that must exist in the process to

be compliant with the objective. Structure requirements consist of structure elements that may be nested.

For example, ISO 20000 requires the creation of so-called service reports and defines the content of a service report: it should contain the performance towards service level targets, trend information, etc.

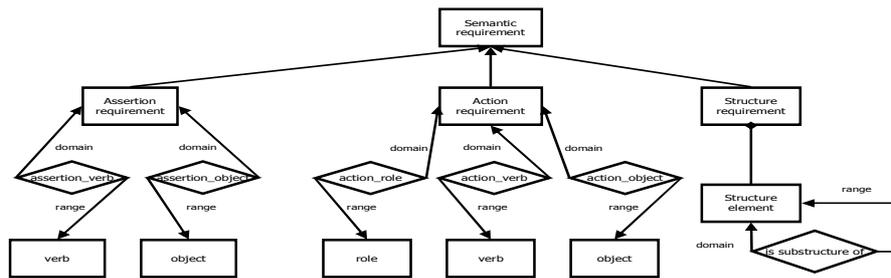


Figure 5: Semantic requirements

The OWL-code to represent the ISO 20000 requirement that the management shall conduct reviews is shown below:

```

<owl:Class rdf:ID="Requirement_Conduct_Reviews">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:hasValue>
            <Object rdf:ID="Reviews"/>
          </owl:hasValue>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:about="#action_object"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#action_role"/>
          </owl:onProperty>
          <owl:hasValue>
            <Role rdf:ID="Management">
              <is_member_of>
                <Section rdf:ID="Section_19"/>
              </is_member_of>
            </Role>
          </owl:hasValue>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty>

```

```

        <owl:ObjectProperty rdf:about="#action_verb"/>
    </owl:onProperty>
    <owl:hasValue>
        <Verb rdf:ID="conduct"/>
    </owl:hasValue>
    </owl:Restriction>
</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
<rdfs:subClassOf rdf:resource="#Action_Requirement"/>
</owl:Class>

```

3.3 Pragmatic requirements

Pragmatic requirements define abstract goals to be achieved. An example is the requirement in ISO 20000 to “communicate the importance of meeting service management objectives and the need for continuous improvement”. The representation of such requirements is very difficult in a machine-readable and especially in a verifiable way. This is because the action outcome does not result in structures that can be represented by a machine, but rather are effects outside the computer system. A possible solution is endorsement. Thus, the effects of actions are not directly measured, but rather the endorsements of people involved.

4 Application scenario

A possible application scenario is the check whether a service process, such as the incident or problem management process [ITSM04], complies with the ISO20000 standard. To do so, the service process is transformed into an ontology-based process representation using the process ontology. For performing the check of compliance, the property of ontologies is used that the membership to an ontology class is not only defined by instantiation, but also declaratively by checking the necessary and required conditions of a class. That means the membership to a class is assigned to an object if the object fulfills all necessary and required conditions of the class. Therefore, all process elements are associated to the compliance ontology, and thus it is checked whether they fulfill the necessary and required conditions of classes in the compliance ontology, as shown in Figure 6. This concept is called realization [SPGK06]. Generally speaking, it is the procedure to find the most specific class an instance belongs to. Realization is supported by reasoners such as Pellet [SPGK06].

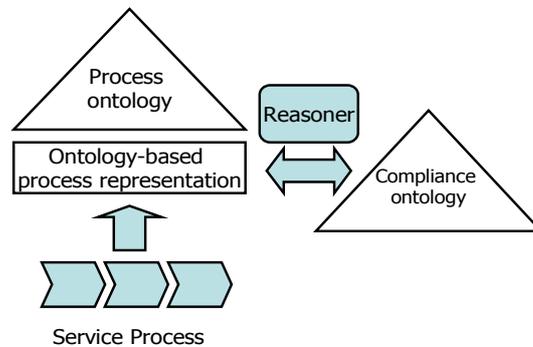


Figure 6: Application scenario

After the reasoner has classified all process elements, the compliance requirements, which are fulfilled, are associated with instances. The compliance requirements, which are not met, have no instances associated. By this means it is possible to determine the set of fulfilled compliance requirements. The concepts defined so far were implemented with the Protégé tool [PROT] and the Pellet reasoner.

5 Related work

There are several areas of related research: the ontology-based representation of business processes and meta-models, the modeling support for service processes, and the view-oriented modeling of processes and workflows.

An important area of related work is the ontology-based representation of business processes and meta-models. An ontology-based approach for formalizing Petri-net-based business processes is given in [KoOb05]. The ontology-based representation of event-driven process chains (epcs) is proposed in [ThFe06]. Both approaches use ontologies for the representation of business processes, but lack the support of service processes.

A first approach to verify business processes against compliance standards has been made in [SoHa06]. A reference model based approach is presented in [KiFo02]. Yet no further details are given on the ontology structure and the procedures for checking compliance. The semantic alignment of business processes using ontologies is described in [BEKO06]. Ontologies are also used to represent the types of modeling methods in [RoGr02], [RoIG04]. In [HeRo07] and [HLDW05] and the basic structure for representing business processes by ontologies is described, however no further details are given.

Service process models have been analyzed by two groups of authors. The support and the modeling of service processes using a coarse-grained architecture is discussed in [KlWe01], [WeKl04], [WeKl03]. A view-oriented approach for the modeling of service processes is presented in [Schm06]. In [BöJK03] a modularization approach for services in the information technology business is proposed.

This work is based on modeling business process aspects and its relations to each other utilizing the Unified Modeling Language to allow a simpler operationalization of

business process reference models. [BWFK04] has designed and developed a simple layer-based model for managing service data. The proposed model is a first step towards ontology.

The view concept is also included in modeling methods, such as ARIS [Sche91], or programming concepts, such as aspect-oriented programming [Kicz96]. It has been described for a range of applications, such as for specifying inter-organizational workflows [BKRR03] or supporting business processes and cross-organizational business processes [Schm03].

6 Conclusions

This paper has introduced an ontology-based approach to represent service processes and their compliance requirements. Thus, it lays the foundation for verifying the compliance of service processes. Two ontologies were defined: The process ontology defines the concepts needed to represent service processes. The compliance ontology contains concepts to represent objectives and requirements for compliance standards. Three types of compliance requirements have been identified: syntactic, semantic, and pragmatic. Syntactic requirements can easily be represented by constraining the properties used for connecting the process elements. Semantic requirements can be further differentiated into assertion, action and structure requirements. Assertion requirements define conditions which have to be met. They consist of a verb and an object. They do not specify directly who is responsible for the condition. Action requirements define actions to be performed as part of the process. An action is comprised of a verb and an object of the verb. Structure requirements define structures that must exist in the process in order to be compliant. Requirements can be checked by applying a reasoner to the ontology-based process representation. Requirements are checked by the classification of process elements. Compliance requirements, which are instantiated, are met. By this means it is possible to determine the set of fulfilled compliance requirements.

7 References

- [BEKO06] Brockmans, S.; Ehrig, M.; Koschmider, K.; Oberweis, A.; Studer, R.: Semantic Alignment of Business Processes. In: Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006), pp. 191-196. INSTICC Press, Cyprus, 2006.
- [BKRR03] Bernauer, M.; Kappel, G.; Kramler, G.; Retschitzegger, W.: Specification of interorganizational workflows - A comparison of approaches. In: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), pp. 30-36. 2003.
- [BöJK03] Böhmman, T.; Junginger, M.; Krcmar, H. 2003. Modular Service Architectures: A Concept and Method for Engineering IT Services. Paper presented at the 36th Annual Hawaii International Conference on System Sciences (HICSS-36), January 6-9, 2003, Big Island, Hawaii
- [BVEL04] S. Brockmans, R. Volz, A. Eberhart, P. Löffler: Visual Modeling of OWL DL Ontologies Using UML. International Semantic Web Conference 2004: 198-213
- [BuSc06] Bullinger, H.J., Scheer A-W. Service Engineering. Springer Verlag, Berlin 2006
- [BWFK04] Böhmman, T.; Winkler, T.; Fogl, F.; Krcmar, H.: Servicedatenmanagement für IT-Dienstleistungen: Ansatzpunkte für ein fachkonzeptionelles Referenzmodell. In: Becker, J.;

- Delfmann, P. (Hrsg.), Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung, pp. 99-124. Physica, Heidelberg, 2004.
- [Grub95] Gruber, T. R.: Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5/6):907-928, 1995.
- [HeRo07] Hepp, M., Roman, D.: An Ontology Framework for Semantic Business Process Management, proceedings of the 8th international conference Wirtschaftsinformatik 2007, February 28 - March 2, 2007, Karlsruhe. In: Oberweis, A; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schmitzler, B.: *eOrganisation: Service-, Prozess, Market-Engineering*, Vol. 1, Universitätsverlag Karlsruhe, pp. 423-440
- [HLDW05] Hepp M., Leymann F., Domingue, J., Wahler, A., Fensel D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management Proceedings of the IEEE ICEBE 2005, October 18-20, Beijing, China, pp. 535-540.
- [ISO20000] <http://www.iso.org/>
- [ITSM04] IT Service Management Forum: IT Service Management – An Introduction. Van Haren Publishing, Amsterdam, 2004.
- [Jabl94] Jablonski, S.: MOBILE: A Modular Workflow Model and Architecture, Proc.of Int'l Working Conference on Dynamic Modelling and Information Systems, Nordwijkerhout, 1994.
- [JaBS97] Jablonski, S.; Böhm, M.; Schulze, W.: *Workflow-Management: Entwicklung von Systemen und Anwendungen — Facetten einer neuen Technologie*. Dpunkt Verlag, Heidelberg, 1997.
- [Kicz96] Kiczales, G.: Aspect-oriented programming. *ACM Computing Surveys*, 28(4), 1996.
- [KiFo02] Kim, Henry M. and Fox, Mark S. (2002). "Using Enterprise Reference Models for Automated ISO 9000 Compliance Evaluation", Proceedings of the 35th Hawaii International Conference on Systems Science.
- [KIWe01] Ralf Klischewski, Ingrid Wetzel: Modeling Serviceflow. *ISTA 2001*: 261-272
- [KoOb05] Koschmider, A.; Oberweis, A.: Ontology based Business Process Description. In: Proceedings of the CAiSE'05 WORKSHOPS, no. 2, pp. 321-333, Portugal, 2005.
- [LeRo97] Leymann, F.; Roller, D.: Workflow-based applications. *IBM Systems Journal*, 36(1), 1997.
- [McHa04] McGuinness, D. L.; Harmelen van, F.: OWL Web Ontology Language - Overview. <http://www.w3.org/TR/owl-features>, 2004.
- [PROT] <http://protege.stanford.edu/>
- [RoGr02] Rosemann, M.; Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. In: *Information Systems (27)*, pp. 75-91, 2002.
- [RoIG04] Rosemann, M.; Indulska, M.; Green, P.: A Reference Methodology for Conducting Ontological Analyses. In: Proceedings of the 23rd International Conference on Conceptual Modelling (ER 2004). pp. 110-121, Shanghai, 2004.
- [Sche91] Scheer, A.-W.: *ARIS – Modellierungsmethoden, Meta-modelle, Anwendungen*; Springer Verlag Berlin, 1991
- [ScBa07] Schmidt, R.; Bartsch C.: Ontology-based Modelling of Service Processes and Services. In: Proceedings of the IADIS International Conference, Applied Computing 2007, pp. 67-74, Salamanca, Spain, 2007.
- [Schm03] Schmidt, R.: Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes. International Conference ICWS - Europe 2003-10-09, Erfurt. In: LNCS 2853, Springer, pp. 123-136, 2003.
- [Schm06] R. Schmidt: Sercomp: A Component-Oriented Method for Flexible Design and Support of Inter-Organizational Service Processes. To appear in *Software Process: Improvement and Practice" (SPIP)*. Preview available:
- [SoHa06] Ljiljana Stojanovic, Hans-Jörg Happel. Ontoprocess - a prototype for semantic business process verification using SWRL rules. 3rd European Semantic Web Conference, Budva, Montenegro, 2006

- [SPGK06] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. Pellet: A practical OWL-DL reasoner, *Journal of Web Semantics (To Appear)*, 2006. (Download)
- [SuSt05] Sure, Y.; Studer, R.: Semantic Web Technologies for Digital Libraries. www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2005_sw_for_dl.pdf, 2005.
- [ThFe06] O. Thomas, M. Fellmann: Semantische Ereignisgesteuerte Prozessketten. DW 2006, Friedrichshafen, Germany.
- [WeKl03] I. Wetzel, R. Klischewski: Serviceflow Beyond Workflow? Concepts and Architectures for Supporting Inter-organizational Service Processes. *CAiSE 2002*: 500-515
- [WeKl04] I. Wetzel, R. Klischewski: Serviceflow beyond workflow? IT support for managing inter-organizational service processes. *Inf. Syst.* 29(2): 127-145 (2004)
- [ZdHe05] Zdravkovic, J.; Henkel, M.: Enabling Flexible Integration of Business and Technology in Service-based Processes. *Proceedings of the CAiSE'05 Workshop*, p. 107 – 114.