

Semantic Integration of Business Component Specifications with RDF Schema

Axel Korthaus, Michael Schwind and Stefan Seedorf

Lehrstuhl für Wirtschaftsinformatik III, Schloss, 68131 Mannheim, Germany
{korthaus|schwind|seedorf@uni-mannheim.de}

Abstract. Semantic Web technologies provide the means for supporting component-based development (CBD) of enterprise applications. This paper builds upon selected component specification proposals for imposing a semantic layer which is used to integrate heterogeneous component specification resources. Most of today's component development methods and specification proposals do not elaborate on how the component specification elements are exchanged and processed for subsequent reuse. The purpose of this paper is to investigate how Semantic Web technologies can be employed for integrating distributed and heterogeneous business component specifications. Moreover, reuse-oriented CBD processes including retrieval and composition require component specifications to be augmented with domain-specific meaning. It is argued that Semantic Web technologies meet the identified requirements. An example is given in which different aspects of a business component specification are represented with RDF Schema. Then, we discuss the alternatives in which graphical notations like the UML can be utilized for Semantic Web-driven business component specification.

Introduction

The development of large-scale distributed enterprise applications is becoming increasingly complex due to global markets imposing constantly changing requirements on enterprises. The systematic reuse of software artifacts is seen as a way of coping with this complexity by breaking systems down into less complex parts in a way that facilitates the composition of a system, at least partly from existing software components.

It is commonly agreed that a standardized method for specifying business components plays a key role in facilitating systematic reuse. Moreover, the semantic description a business component provides to its peers and its environment seems to be a crucial factor for achieving this objective [1, 2]. These requirements led to a number of methodologies and standardized notations that cover the various aspects of business component specification. An example for the continuous efforts to coherently describe the various aspects of business components is the memorandum for the Standardized Specification of Business Components [3]. It provides comprehensive recommendations for the specification of functional as well as extra-functional component aspects. The memorandum proposes seven specification levels that cater

to the different requirements with regard to the composition of enterprise applications from encapsulated business components.

Some aspects of a component specification can be easily represented using widely accepted notations, e.g. the syntactical specification of an interface using CORBA IDL or the formal description of the behaviour a component service provides by specifying pre- and postconditions in the Object Constraint Language (OCL). However, many other aspects have not been standardized by yet. As an example, the lacking standard notations for the specification of quality criteria will possibly display a hindrance to component reuse. Thus, a comprehensive component specification needs to represent both functional and extra-functional component aspects.

This paper elaborates on two issues that are closely connected to the standardization of business component specifications. First of all, the different aspects of business component specifications need to be presented in a uniform and coherent way for being marketable. Furthermore, a repository for component specifications should be able to cope with the specifics of different notations. When an integration layer is imposed on distributed information resources, the automated exchange of specifications between different applications becomes possible.

The second issue being discussed in this paper concerns the augmentation of specification elements with explicit semantics. All specification layers rely on a common notion of the concepts of the domain under observation. Domain-specific references are introduced by describing a component aspect with unequivocal terms. In the memorandum, the layer for describing functional terms is referred to as terminology layer. The authors of the memorandum mention the use of the Resource Description Framework (RDF) as one option for specifying the required terminology, but also encourage the use of natural language descriptions as the primary notation. We think that the latter suggestion inhibits the automatic deduction of assertions about the composability of components with respect to the domain and expected properties of the composed system. Therefore, this paper presents a basic approach for applying RDF to the terminology and interface levels as proposed.

The technologies provided by the Semantic Web are highly suitable for supporting the identified tasks due to their ability to describe all kinds of resources in a machine-readable way. Accordingly, this paper discusses how RDF and RDF Schema, can be employed for integrating distributed, heterogeneous business component specifications. The main rationale behind RDF is metadata representation. Moreover, RDF responds to data integration problems because it facilitates the exchange of metadata between Web applications in a flexible way [4]. Domain-specific functional terms can be represented in RDF as Web resource. RDF Schema defines a standard vocabulary for the serialization of ontologies.

Throughout this paper, the Semantic Web standards are surveyed for both the integration of specification resources and for the full representation of component specifications. In the latter usage scenario, it is required to derive specification data from other notations that provide better understandability for human developers. Therefore, it is shown how component interface specifications in UML can be augmented with domain-specific functional terms. The example given in this paper is limited to the interface layer and the terminology layer. However, the suggested approach applies to other aspects of component specification as well.

The paper is structured as follows: In the first part we deduce the requirements for an integrated and semantically-enhanced specification mechanism based on what we perceive as shortcomings of conventional specification approaches. Afterwards we give a brief introduction to the Semantic Web and RDF Schema, followed by the different ways in which the proposed technologies can be applied to support CBD-related activities. To improve the practical applicability of our approach we discuss how interface specifications in UML might be transformed into RDF Schema.

Basic Requirements for Business Component Specification

Before studying the different ways in which RDF Schema can be applied to business component specification we will briefly state the basic requirements that account for supporting CDB-related activities with Semantic Web technologies.

Component-specific tasks that can benefit from a semantically rich specification are component **retrieval** [5], the **deduction of facts** about the composed system, **compatibility testing** and **component composition** [6]. To enable tool support for these tasks, a number of key requirements must be met. In order to be machine-readable, the component specification data must be presented in a **uniform** and **coherent** way.

The retrieval of business components by specification of search criteria in the terminology of the domain requires tools and methods for representing **domain-specific knowledge**. To allow for the integration of components from different vendors, a **mutual agreement** on the specification elements and representation formats is required. Ideally, CBD applications can cope with **variations** on these elements and the introduction of new termini **without compromising machine-processability**. **Adaptability** and **extensibility** are other important aspects with regard to the integration of new specification aspects as well as new domain knowledge.

Another desired feature is the ability to link specification elements with secondary information resources that do not constitute an integral part of a component specification but rather reflect background knowledge on the underlying design decisions. This feature helps to increase human-understandability of component specifications [6]. In other words, an all-embracing business component specification framework needs to integrate two basic interests, namely machine-readability and human-understandability.

Finally, a business component specification framework is required to be flexible and adaptable to evolutionary changes. In the subsequent sections we will argue that the requirements above can be best fulfilled when relying on Semantic Web technologies. They can be used to solve two main problems: The semantic linking among specification resources and the enhancement of specification elements with explicit semantics which is a prerequisite for software component retrieval and composition. While a graphical notation such as the UML is well-suited for describing various aspects of a business component, it is limited when it comes to the integration of heterogeneous, distributed resources and the sharing of components specifications between different applications.

Business Component Specification with RDF Schema

In this part, we investigate how business components can be described and integrated by employing Semantic Web technologies. At first, a short introduction to the Semantic Web and RDF Schema is given. This is the starting point for exploring the potential uses of RDF for business component specification. It is then analyzed how a business component specification can be fully represented with RDF Schema.

Introduction to the Semantic Web and RDF Schema

In recent years, the upcoming vision of a Semantic Web [7] has led to a series of standardization activities under the umbrella of the W3 Consortium [8]. The main objective of the Semantic Web is to provide web content in a machine-readable way. The Semantic Web is described as “an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” [9].

The Semantic Web activities controlled by the W3C have produced two key technologies, namely the Resource Description Framework (RDF) [4] and the Web Ontology Language (OWL) [10]. Both standards are designated to set up a framework which allows the sharing and reuse of data on the Web. RDF is used to represent information and to exchange knowledge between Web applications. OWL is used to publish and exchange ontologies which are formal sets of concepts, relations and constraints. The technologies developed for the Semantic Web target at a broad range of application areas such as advanced Web search, software agents, knowledge management and enterprise application integration. Accordingly, these technologies are potentially capable of describing component specification-related entities and processing them for various purposes. Moreover, they provide the means for publishing and reusing domain-specific concepts.

RDF is a markup language for describing metadata about web resources [4]. The metadata is represented in an RDF graph which is serialized in the XML format. The captured information can then be exchanged between applications without loss of meaning. The roots of RDF are to be found in the fields of knowledge representation, artificial intelligence and data management. RDF particularly draws from conceptual graphs developed by Sowa [11].

An RDF expression is a collection of labeled nodes and directed arcs linking pairs of nodes. It is also called RDF graph. An RDF Graph is represented by a set of triples, which consist of a subject, predicate and object each. In other words, resources are described in terms of properties and property values. The qualified name of a property is given beside the arc. A resource in RDF is based on a Uniform Resource Identifier (URI). The Web resource identification mechanism can be generalized to represent entities that cannot be directly retrieved on the Web. This notion can be leveraged for the specification of business components, where statements about real-world entities as well as abstract entities, e.g. domain-specific concepts, are made.

The RDF Schema language defines a vocabulary and a simple meta-model for RDF. This way, it becomes possible to express an ontology as a set of RDF triples and to

reason with it. RDF Schema allows the definition of key elements such as classes, instances, sub-properties, domains and ranges of properties.

Figure 1 shows a simple RDF schema and graph which is used to describe business components with standard metadata. The schema defines the classes `BusinessComponent` and `ComponentProvider`. The business component itself is represented as resource using a qualified name within a predefined namespace. The parameter values are given as literals and resources, respectively. Standard component metadata typically involves a component publisher, address data, version info and functional descriptions among other things. In a COTS application scenario, the business component description will be published in an industry-wide banking components directory.

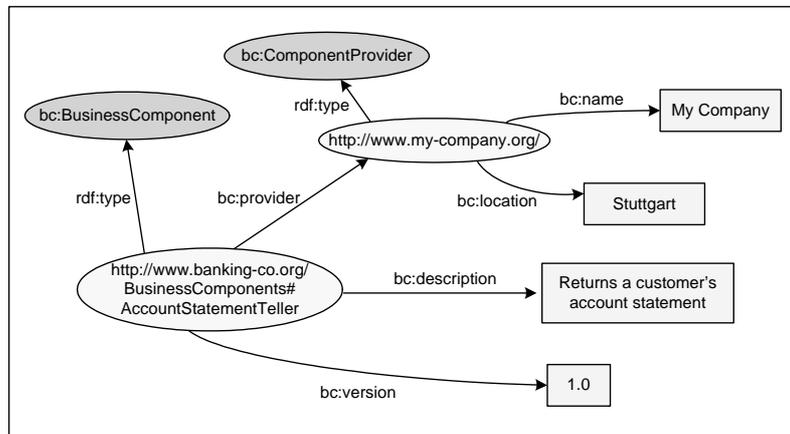


Fig. 1. Representation of component metadata in RDF

The proposed Semantic Web technologies including RDF and OWL are already an established subject in software engineering, e.g. in Semantic Web enabled Web services ([12], [13]), which are primarily for service discovery and runtime integration. The application scenario considered here substantially differs from that because it concentrates on integrating specification resources and supporting CBD related activities from a development perspective.

Potential Uses of RDF Schema for Business Component Specification

The following section analyzes the different applications of RDF Schema that contribute to a multi-purpose business component specification framework. Not only can RDF be used to exchange component metadata as shown in the above example, it is also applicable for integrating specification resources that would remain unconnected otherwise and for representing component specifications entirely. Figure 2 outlines the two different ways of employing RDF Schema.

In the first alternative, a superimposed information space is spanned that semantically describes physical information resources but leaves them unchanged [14]. The superimposed information layer may be implemented as RDF graph which references the information resources on the base layer. It interconnects business components with various component aspects, and provides the resources with a well-defined meaning.

In the second alternative, the relevant information is extracted from heterogeneous specification resources and represented as RDF Graph. This alternative requires the existence of a uniform, standardized component specification meta-model. Moreover, a transformation needs to be defined for every resource type which describes a particular component aspect. As an example, the information may be extracted from UML models that are encoded in XMI and accommodate a business component specification.

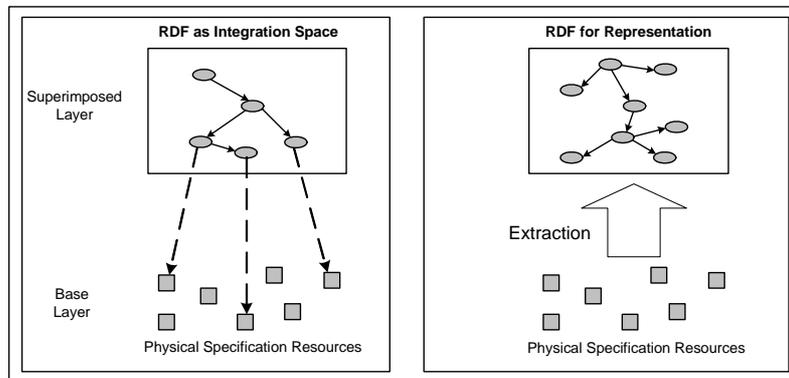


Fig. 2. Different usage scenarios of RDF Schema for business component specification

The decision which usage scenario is preferable depends on the specific application. If the underlying application is resource integration and data management, then the first alternative is preferable. If the application in CBD involves querying and inference, it is necessary to represent all specification elements as a RDF graph.

RDF Schema may contribute to a business component specification framework in more than one way. The general functions of RDF Schema contributing to component specification are summarized in Table 1. As illustrated above, resource integration plays the most pertinent role. Domain referencing refers to the augmentation of specification elements with domain-specific functional terms. Since RDF is founded on the concept of resources it can be applied for capturing domain-specific functional meaning. Moreover, formal representation is required mainly for machine-processing such as compatibility-testing and component composition. Finally, representing component specifications in RDF improves human-understandability because arbitrary resources can easily be augmented with secondary information which provides background knowledge on a particular specification element.

Basic function	Specification task
Resource integration	Integration of distributed, heterogeneous resources that constitute a logical coherent component specification.
Domain referencing	Unequivocal referencing of domain-specific concepts being represented as Web resource and augmentation of specification elements with explicit semantics.
Formal representation	Supporting inference on component specification data.
Documentation	Description of component metadata and involvement of secondary resources which provide background knowledge.

Table 1. Basic functions of RDF Schema for component specification

In this section, we investigated how Semantic Web technologies can be generally employed for supporting the specification of business components. The main advantage of RDF is its ability to structure component specification data in a more coherent way - compared to relying solely on UML model specifications, Interface Definition Languages or normative languages. The second advantage lies in its ability to add semantic information in a superimposed layer. The semantics can either be added by describing specific roles and relations or by augmenting resources with domain-specific knowledge.

Representation of Business Component Specifications with RDF Schema

This section elaborates on applying RDF Schema for the full representation of business component specifications. The scope of our analysis is narrowed down to the interface layer and terminological layer.

In addition to providing component metadata and domain-specific functional meaning on a superimposed layer, it may be reasonable to represent the entire external view of a component as RDF graph. In this case, it becomes possible to query the content of business component specifications using an RDF query language. For example, inference allows compatibility testing of input and output parameters on the semantic level. Another application scenario is the retrieval of business components according to their domain-specific functional meaning. Logical inference with component descriptions and RDF Schema will not be discussed within the scope of this paper but is covered in [15]. However, this feature is a major advantage compared to other representation languages such as the Unified Modeling Language and its supporting standards.

Prior to proposing a representation of a component's external view in RDF, we take a brief look at how a component is specified using UML as graphical notation. Figure 3 depicts a business component specification which offers a service for receiving a

customer's account statement. The interrelation between the service definitions in the component view and the interface specification is established by the association stereotype `<<provides>>`. A similar approach to component specification is illustrated in the UML Components methodology [16] and in the BOOSTER**Process* [17].

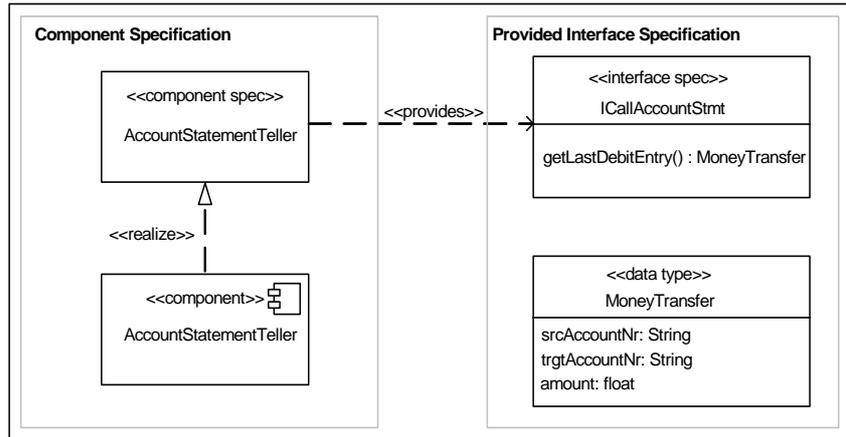


Fig. 3. Excerpt of a business component specification in UML

At the interface level, the syntactic premises for component communication are arranged. The interface specification comprises provided and requested interfaces, operation signatures, attributes, variables, constants, data types, exceptions and error messages. In our example, the operation returns a structured data type which defines the transfer amount and the involved banking accounts as its attributes. Alternatively to the usage of UML, the definition of interfaces might as well be conducted using CORBA IDL.

The elements of a component specification can be transformed into an RDF graph. Figure 4 visualizes one possible representation of a component interface specification with RDF Schema. If a standardized component meta-model is created, it does not matter whether a serialized UML model in XMI or an IDL input file is provided. Thus, heterogeneous sources can be easily integrated using RDF Schema and defining an appropriate transformation for every intermediate representation. The presented example is simplified in order to improve understandability. That is why some RDF constructs like the collection type `rdf:Bag` are omitted. Further, the example is not based on a coherent, formalized component meta-model which will be the next logical step in developing a business component specification framework.

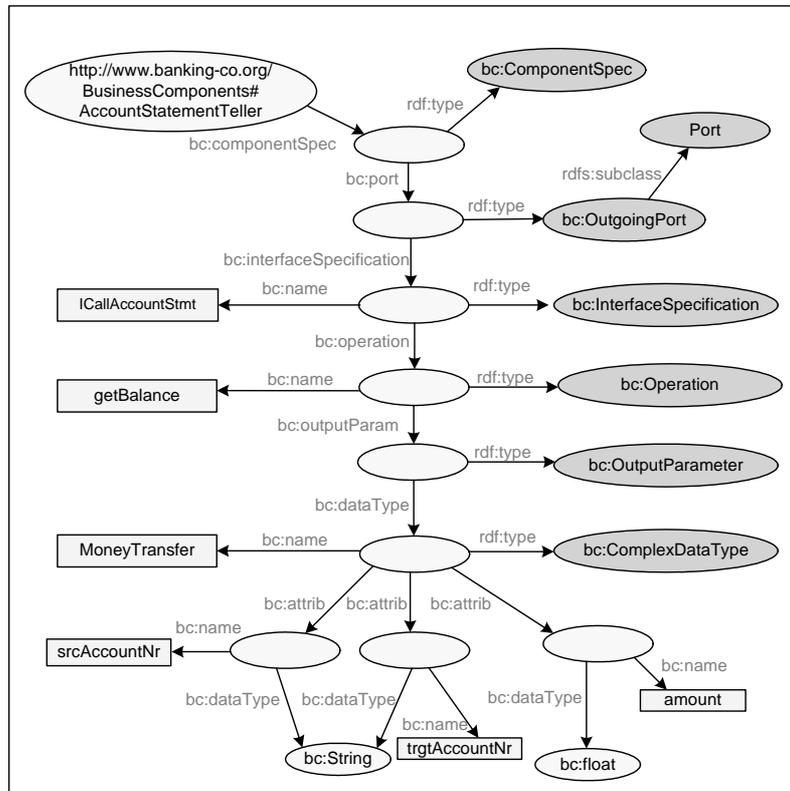


Fig. 4. Example of a business component specification using RDF Schema

A component model to be represented in the form of an RDF graph will usually include both syntactic and semantic descriptions. As mentioned above, RDF Schema is applicable to represent domain-specific functional terms in a domain vocabulary. Explicit semantics are introduced when the domain-specific terms are provided as Web resources. Because there is a correspondence between the interface level and higher levels of abstraction [3], a semantic mapping between specification elements can be established and represented by defining an RDF statement.

Firstly, task definitions on the task level correspond to services offered by a business component. Secondly, functional terms on the terminological level can be mapped to data types on the interface level. The relation between functional meaning and data type is demonstrated in figure 5. It needs to be pointed out that the functional term `IBANAccountNr` is unequivocal only within the defined namespace. The interface specification makes a statement about the relation between a data type and a functional term. This scheme can be applied to other specification elements such as service descriptions as described in [12].

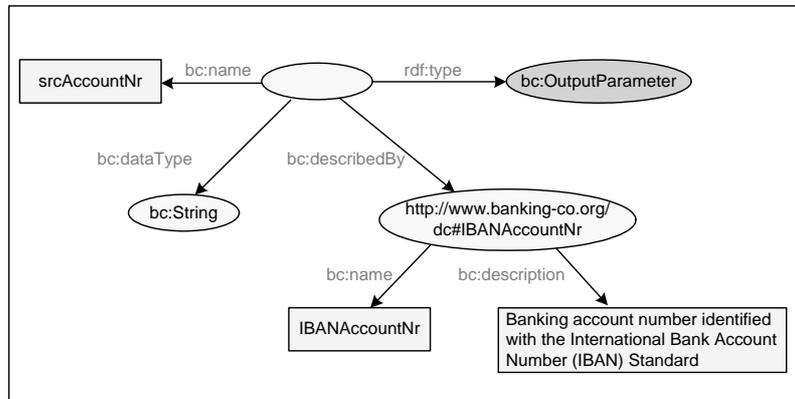


Fig. 5. Augmentation of a data type with domain-specific meaning

A component specification framework will not be designed to replace existing specification agreements like the UML. Instead, it primarily targets at propagating business component specifications as a coherent unit. If UML is maintained as primary notation, it needs to be augmented with domain-specific terms that are provided as Web resources in RDF. A basic solution to this requirement is proposed in [15]. The mechanism which is used to augment UML model elements with explicit semantics are tagged values. Tagged values are name-value pairs whose interpretation lies outside the scope of the UML meta-model. They are similar to an RDF statement which is represented as a triple consisting of subject, predicate and object. In case of a tagged value, the subject is represented by the UML model element. The RDF predicate is represented by the parameter name and the object is represented by the parameter value. To be processable within an RDF-based information system, the individual parts will have to be specified as Web resources. Then it becomes possible to attach additional information to the individual parts. This way, secondary information resources can be provided along with functional terms, e.g. terminological and process task descriptions. In this case, a domain vocabulary provides valuable background knowledge to the component developer, serving the purpose of achieving a better understandability for humans.

Related Work

The technologies that were introduced within the Semantic Web context address a wide range of software engineering issues which were not covered within the scope of this paper. A more thorough discussion of the potential uses of Semantic Web technologies is carried out in [18].

In this paper, we decided to concentrate on using Semantic Web technologies for describing different aspects of a business component. Further, the scope was limited to the interface layer and the terminology layer of a business component specification.

The application of RDF for representing domain-specific terms has already been suggested within the memorandum for the *Standardized Specification of Business Components* [3]. However, no mechanisms for the technical mapping between the specification layers are proposed.

An approach for the semantic integration of different models is presented in [19]. The augmentation of UML model elements including the representation of domain-specific knowledge and reasoning using external fact bases is investigated in [15]. Furthermore, there exist a number of proposals for transforming UML model specifications into an RDF or OWL representation. As an example, a transformation-based approach for representing UML models in RDF is covered in [20] and [21]. A metamodel and UML profile for RDF Schema is currently being specified within the scope of the OMG Ontology Definition Metamodel [22].

Conclusion

In this paper, we proposed to use Semantic Web technologies for the semantic integration of business component specifications. The main advantages of RDF Schema and OWL in contrast to other representation formats are Web-based representation of domain-specific knowledge as well as support for deductive querying and reasoning.

We identified two different ways to apply Semantic Web technologies. Firstly, they can be employed for achieving integration of heterogeneous specification resources. Secondly, they can be used for representing the entire content of a component specification. In the given example, it is proposed to derive an RDF graph from a component specification in UML. In this case, a UML profile and a transformation from the XMI serialization into RDF needs to be defined. The paper gives no recommendation on which of the alternative approaches should be preferred. Instead, the role Semantic Web technology may play in standardizing business component specifications needs to be more deeply discussed.

Further, the consequences on the component-based development process need to be investigated. Especially the impact of semantic enhancements to component specifications will be subject to future research within the CollaBaWue¹ project. There are also plans for a prototypical implementation of a Semantic Web layer for component specification.

The semantic integration of business component specifications promises substantial benefits with regard to implementing tool-support for component retrieval, deduction of system properties and component composition. Especially online markets for commercial business components could benefit from semantically rich component specifications and an integrated presentation format.

¹ CollaBaWue is a research project funded by the Landesstiftung Baden-Wuerttemberg. Its focus is on collaborative software development within the domain of financial service providers and on supporting technologies like component-based development and semantically rich component specifications. For further information see <http://www.collabawue.de>

Acknowledgement

This work was supported by a grant from the Landesstiftung Baden-Wuerttemberg foundation to Michael Schwind and Stefan Seedorf.

References

1. Overhage, S.: Die Spezifikation – kritischer Erfolgsfaktor der Komponentenorientierung. In: Turowski, K. (Editor): Proceedings of the 4th Workshop on Component-oriented Business Applications. University of Augsburg 2002, pp. 1-17.
2. Crnkovic, I., Hnich, B., Jonsson, T. and Kiziltan, Z.: Specification, Implementation, and Deployment of Components. Communications of the ACM Vol. 45, No. 10, 2002, pp. 35-40.
3. Turowski, K., et al.: Standardized Specification of Business Components. Memorandum of the working group 5.10.3, Component Oriented Business Application Systems, February 2002.
4. W3C 2004: RDF, W3C Recommendation 10 February 2004. <http://www.w3.org/TR/rdf/>.
5. Braga, R. M. M., Mattoso, M., Werner, C. M. L.: The use of mediation and ontology technologies for software component information retrieval. In: Proceedings of the 2001 symposium on Software reusability, ACM Press, 2001.
6. Overhage, S.: UnSCom: A Standardized Framework for the Specification of Software Components. In: Weske, M.; Liggesmeyer, P. (Editors): Object-Oriented and Internet-Based Technologies, 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World, NODe 2004, Proceedings, Erfurt 2004, pp. 169-184.
7. Berners-Lee, T., Hendler J. and Lassila, O.: The Semantic Web. Scientific American, May 2001.
8. W3C 2001: Semantic Web, <http://www.w3.org/2001/sw/>, 2001.
9. Hendler J., Berners-Lee, T. and Miller E.: Integrating Applications on the Semantic Web (english version). Journal of the Institute of Electrical Engineers of Japan, Vol. 122(10), October, 2002, p. 676-680.
10. W3C 2004: The Web Ontology Language (OWL) Specification. <http://www.w3.org/TR/owl-features/>, 2004.
11. Sowa, J. F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Course Technology, Boston, 2000.
12. OWL Services Coalition: OWL-S Semantic Markup for Web Services. <http://www.daml.org/services/owl-s/1.0/owl-s.html>, 2004.
13. IST-2001-37134: Semantic Web Enabled Web Services (SWWS) Project. <http://swws.semanticweb.org/>, 2001.
14. Maier, D. and Delcambre, L.: Superimposed Information for the Internet. In: SIGMOD Workshop on The Web and Databases WebDB99, Philadelphia, 1999.

15. Süß, J. G. and Leicher, A.: Augmenting Domain Specific UML Models with RDF. In: Proceedings of the 3rd Workshop in Software Model Engineering, Lisbon, 2004.
16. Cheesman, J. and Daniels, J.: UML components - a simple process for specifying component-based software. Addison-Wesley Longman Publishing Co., Boston, 2000.
17. Korthaus, A.: Komponentenbasierte Entwicklung computergestützter betrieblicher Informationssysteme. Series: Informationstechnologie und Ökonomie, vol. 20, Editors.: Gaul, W. and Schader, M., PhD Thesis, Peter-Lang-Verlag, Frankfurt a.M., 2001.
18. Tetlow, P., Pan, J., Oberle, D. et al.: Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering. W3C, Semantic Web Best Practices and Deployment Working Group, Draft 2005/06/30, <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>, 2005.
19. Wang, H.: Semantic Web and formal Design Methods. PHD thesis, National University of Singapore. <http://www.cs.man.ac.uk/~7Ehwang/thesis.pdf>, 2004.
20. Cranefield, S.: UML and the Semantic Web. In: Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford, 2001.
21. Melnik, S.: Representing UML in RDF. <http://www-db.stanford.edu/~melnik/rdf/uml/>, 2000.
22. OMG: Ontology Definition Metamodel RFP. <http://www.omg.org/docs/ad/05-08-01.pdf>, 2003.