The RDF Schema Specification Revisited

Wolfgang Nejdl and Martin Wolpers and Christian Capelle
Institut für Technische Informatik
Rechnergestützte Wissensverarbeitung
Universität Hannover
Appelstraße 4, 30167 Hannover
{nejdl,wolpers,capelle}@kbs.uni-hannover.de

1 Abstract

In this paper, we will discuss the proposed recommendation (March 1999) for an RDF Schema Specification and compare the approach taken in this specification to conventional meta data models. We will point out several pecularities in the RDF schema specification compared to more conventional approaches, which are basically due to the dual role of properties such as subclass and domain both as primitive constructs used in the definition of the RDF schema specification and as specific instances of RDF properties. We then discuss an approach for specifying RDF schemas, which distinguishes between a set of specific properties, which are needed in the RDF meta model itself, and conventional properties. Finally we show how we use this modified RDF meta model in a structured hypertext system (the KBS Hyperbook System), which uses RDF annotations not only for informational purposes, but also for structuring hypertext according to the semantic relationships recorded by these RDF annotations.

2 Introduction

Semantic annotations and metadata are seen as a crucial technique for transforming the World Wide Web from a huge set of amorphous pages interlinked with each other to a semantic web, which uses semantic annotations in order to give meaning to these pages and their relationships [Lass 1998]. This is in line with the development in other disciplines such as database and software engineering, which use semantic data models and schemas to define and give meaning to data and parts of programs (e.g.[CJB 1998, Guar 1998]).

The standard for initiating this transformation is the RDF (Resource Description Framework) Standard [W3C 1998], which has been defined in a set of two documents defining the syntax and semantics of RDF annotations and of RDF schemas. Using RDF, WWW resources are annotated with semantical information, which uses underlying conceptual models (schemas) to define the classes and properties used for these semantic annotations. The shared use of such a schema makes it possible for a group of people to use semantic annotations, which are comparable with each other, because they use the same set of concepts and properties to describe their WWW pages.

In this paper, we will discuss the proposed recommendation (March 1999) for an RDF Schema Specification [W3C 1999b]. We will point out several pecularities in the RDF schema specification compared to more conventional meta data models, which are basically due to the dual role of the properties rdfs:subClassOf, rdf:type, rdfs:domain and rdfs:range, which are used both as primitive constructs in the definition of the RDF schema specification and as specific instances of RDF properties. This dual role makes it possible to view e.g. rdfs:subClassOf as an RDF property just like other predefined or newly introduced RDF properties, but introduces a self referentiality into the RDF schema definition, which makes it rather unique when compared to conventional model and meta modeling approaches, and makes the RDF schema specification very difficult to read and to formalize.

Based on this observation, we discuss an alternative approach for specifying RDF schemas, which distinguishes between primitive properties such as rdfs:subClassOf, which are needed in the RDF meta model itself, and conventional properties. This alternative meta model separates clearly between meta modeling constructs and their instantiations and avoids any self referentiality for these constructs, making it similar to more conventional meta modeling approaches such as IRDS [IRDS 1990].

Finally we use this modified RDF meta model in a structured hypertext system (the KBS Hyperbook System [FHN 1997, FNW 1998, NeWo 1998, HeNe 1999]), which uses RDF annotations not only for informational purposes, but also for structuring hypertext according to the semantic relationships recorded by these RDF annotations. Using this meta model, the KBS Hyperbook System can display both conventional metadata information for WWW pages, as well as use metadata information concerning semantical relationships to connect semantically related pages based on these relationships.

After discussing the RDF schema specification in Section 2, we introduce and discuss an alternative RDF schema specification in Section 3. The main goal of this alternative model is to make explicit the different meanings of the original RDF schema constructs. We therefore concentrate on the main RDF constructs in order not to complicate our presentation and ignore in this paper some RDF constructs like rdfs:ConstraintProperty or rdfs:ConstraintResource. Finally, Section 4 gives a short introduction to the KBS Hyperbook System as discussed in [NeWo 1999] and uses our alternative RDF schema specification as a basis for connecting semantically related WWW pages based on semantic relationships expressed by RDF annotations.

3 The RDF Schema Specification

First, let us describe the RDF Schema Specification, based on [W3C 1999b], in order to discuss and point out some unconventional design decisions taken in this specification. We will try to make this chapter self-contained, but a working knowledge of [W3C 1999b] will help to understand the discussion in this section. The prefixes rdf: and rdfs: indicate, whether a resource is part of the RDF Data Model [W3C 1999a] or the RDF Schema Specification [W3C 1999b].

RDF schemas are used to define the structure of the metadata that are used to describe WWW resources (i.e. WWW pages or parts of WWW pages, referenced by an URL). The RDF Schema Specification consists of some basic classes and properties, and can be extended by others to fit possibly any given domain. Classes are arranged hierarchically, and the use of properties can be constrained to members of certain classes. The root of the class hierarchy is rdfs:Resource, rdfs:Class is subclass of rdfs:Resource.

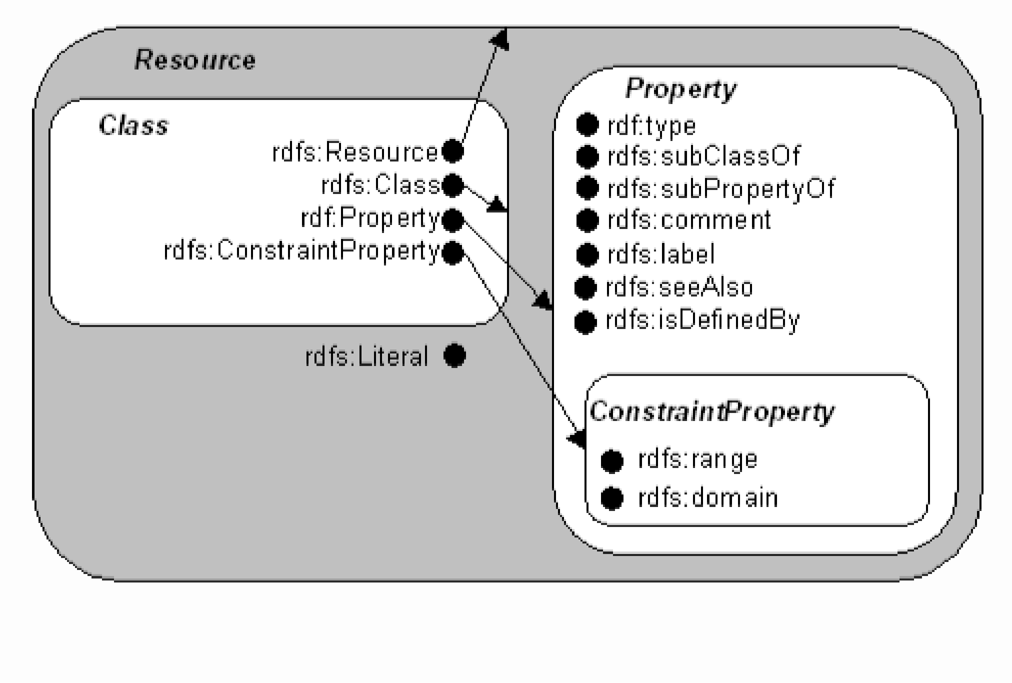


Figure 1: *RDF Classes and Resources as Sets and Elements*

Properties are defined by the rdf:Property class and can be seen as attributes, that are used to describe resources by assigning values to them. Properties are resources themselves. The RDF Schema Specification defines four specific properties (rdfs:subClassOf, rdf:type, rdfs:range, rdfs:domain) that have, unlike other predefined or self-defined properties, certain constraints. These four properties are both used to define the other RDF schema constructs and also as constructs defined in the RDF schema. Additional predefined properties such as rdfs:seeAlso and rdfs:comment are used to specify resources with related subjects, or to give a human readable description of a resource. The fact, that these properties are predefined can be seen as a convenience, they are not needed for the definition of other properties.

Figure 1, 2 and 3 (which we have reproduced from [W3C 1999b]) show the RDF schema specification as a set of pictures. We will use an abbreviated description of these pictures based on the text in [W3C 1999b] and discuss the design issues we want to address in our alternative RDF schema specification model. Figure 1 shows RDF classes, subclasses and resources as sets, subsets and elements. A class is depicted by a rounded rectangle, a resource is depicted by a large dot. Arrows are drawn from a resource to the class it defines. A sub-class is shown by having a rounded rectangle (the sub-class) completely enclosed by another (the super-class). If a resource is inside a class, then there exists either an explicit or implicit rdf:type property of that resource whose value is the resource defining the containing class.

The constraint properties rdfs:range and rdfs:domain are distinguished from the other predefined properties. The property rdf:type is present both as a specific property and depicted as an arrow, rdfs:subClassOf both as a specific property and depicted as set containment.

Figure 2 shows the same information about the class hierarchy as in figure Figure 1, but does so using a „nodes and arcs“ graph representation of the RDF data model. If a class

is a subset of another, then there is an rdfs:subClassOf arc from the node representing the first class to the node representing the second. Similarly, if a Resource is an instance of a Class, then there is an rdf:type arc from the resource to the node representing the class.

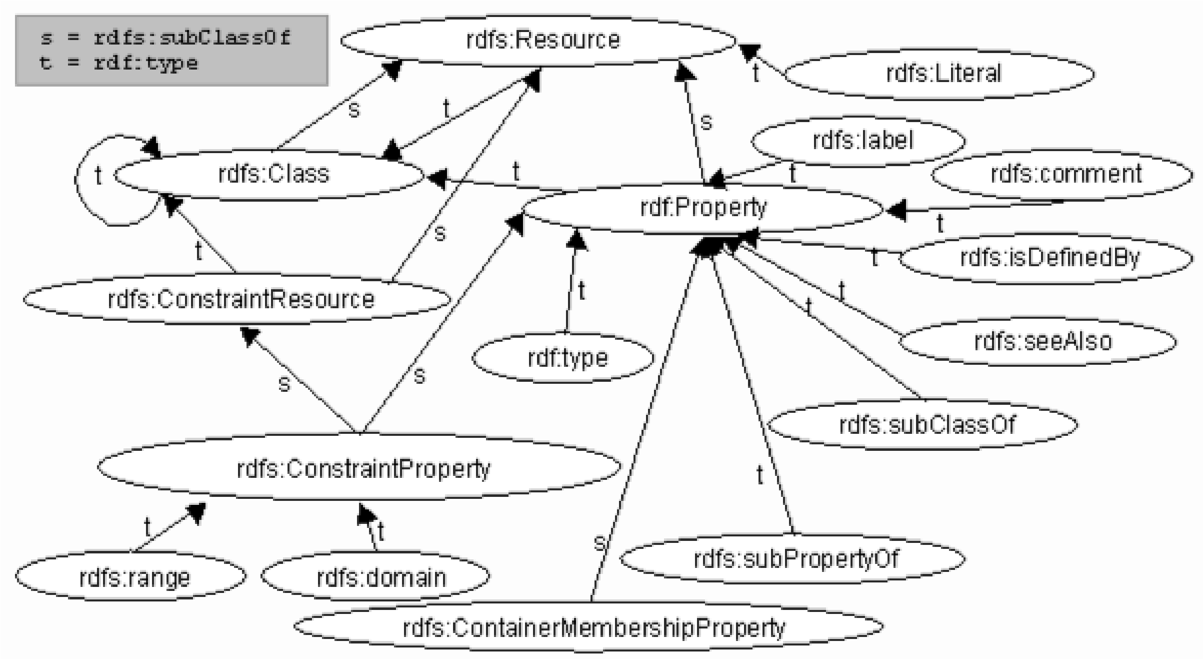


Figure 2: *Class Hierarchy for the RDF Schema*

Again, rdfs:subClassOf is present both as a specific property and a primitive construct (an arrow labelled with „s“), rdf:type as specific instance of property and as primitive construct (an arrow labelled with „t“).

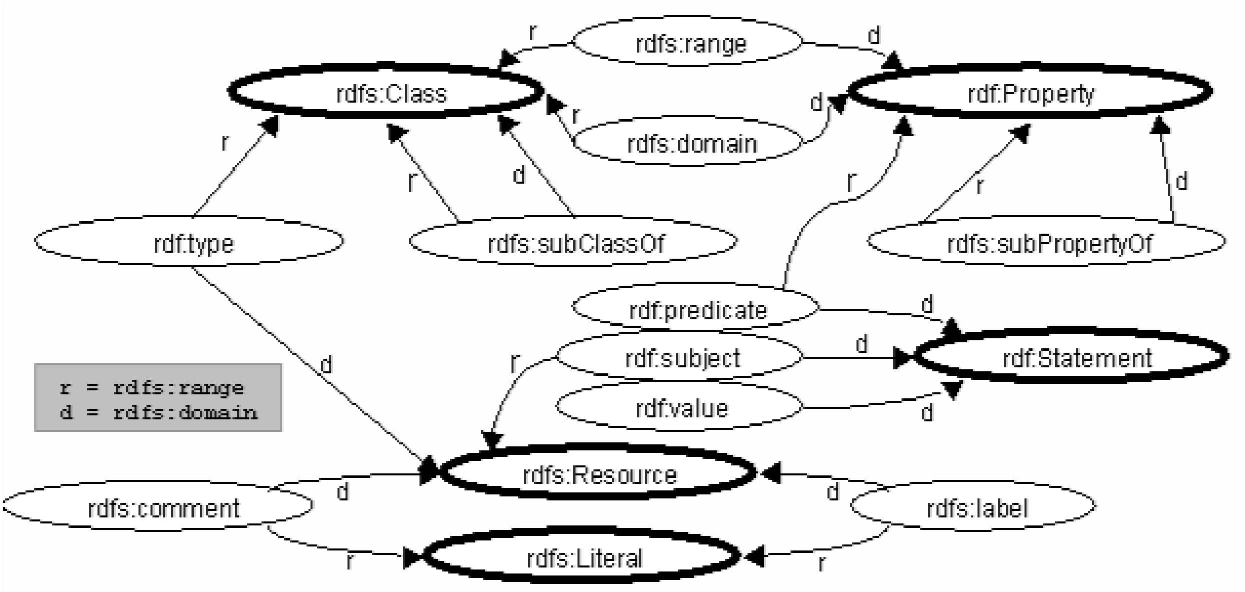


Figure 3: *Constraints in the RDF Schema*

Figure 3 takes a different view compared to the previous two figures, and uses rdfs:range and rdfs:domain as primitive constructs to constrain the relationship between RDF classes and properties. Now, rdfs:range and rdfs:domain occur both as primitive constructs and as explicit properties.

4 An Alternative Approach to the RDF Schema Specification

As discussed in the previous section, the properties rdfs:subClassOf, rdf:type, rdfs:range and rdfs:domain are used both as primitive constructs for specifying RDF schemas and also as specific properties defined by RDF. This is probably at least partially motivated by the goal to make the RDF schemas and all their constructs themselves available as explicit metadata about the RDF schemas.

However, this makes it very difficult to formalize the RDF schema specfication as described in [W3C 1999b] (or even impossible). We are currently exploring this issue in more detail.

In the following, we will therefore distinguish between the specification for RDF schemas (i.e. how RDF annotations are structured and used) and the view, that RDF schemas (and their parts) are themselves data, which are defined by specific WWW resources. Additionally, we will distinguish between the different uses of rdfs:subClassOf, rdf:type, rdfs:range and rdfs:domain, and introduce different constructs whenever they are used in different contexts or represent them by primitive constructs from the modeling language.

As modeling language we will use O-Telos [MBJ+ 1990, SJJ 1996], which is a language very suitable for modeling and meta modeling tasks. The main constructs in O-Telos are Class and Relation (resp. Attribute). Predefined relations are the „isA“-Relation (denoting a class - subclass relationship) and the „in“-Relation (denoting the relationship between a class and its instances). The original RDF schema specification is not expressible in O-Telos, as this would lead to constructs like „type type property“, where the first „type“ is the RDF property with this name, and the second „type“ is the modeling language construct expressing instantiation (i.e. „type“ is a „type“ (an instance) of property).

Figure 4 represents the part of the RDF schema specification, which defines the main RDF constructs used for constructing RDF schemas and how these constructs are used for annotating WWW resources. We have decided not to include a few RDF constructs like rdfs:ConstraintProperty or rdfs:ConstraintResource, as they are not relevant to our discussion. Figure 4 shows the familiar constructs of RDF_Class and RDF_Property. They are related through the relationships RDF_Domain and RDF_Range, which correspond to rdfs:domain and rdfs:range. However, RDF_Domain and RDF_Range are specific relationships defined by this specification, and are therefore no instances of RDF_Property.

rdfs:subClassOf and rdf:subPropertyOf are substituted similarly by the two relationships RDF_Subclass and RDF_Subproperty, which have the same semantics as the metalanguage construct „isA“ in O-Telos, and allow the definition of class - subclass and property - subproperty relationships. rdf:type is used in different meanings in the RDF Schema specification. In case it is used to express the relationship between RDF metadata and a conventional WWW resource, we introduce the corresponding relations of „RDF_Annotates“ and „RDF_Defined_By“. In case it is used as conventional instantiation (needed in our model only for instantiating the RDF

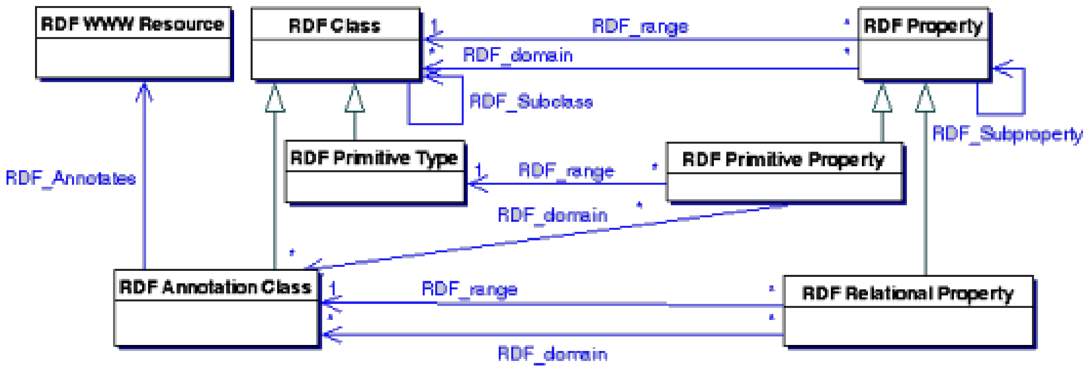


Figure 4: *Main Constructs of the Alternative RDF Schema Specification*

schema to get the actual RDF annotations, as discussed below), we substitute it by the corresponding construct „in“ from O-Telos, which denotes instantiation.

We distinguish between two subclasses of RDF_Class, namely RDF_Annotation_Class and RDF_Primitive_Type. RDF_Annotation_Class (plus subclasses like RDF_SeeAlso) has as instances what we might call RDF_Annotation_Objects, which (together with their instantiated properties) basically correspond to RDF statements and are used to annotate WWW resources. RDF_Primitive_Type with subclasses like String, etc. has as instances specific values for RDF properties (of type String, Integer, etc.) An RDF_WWW_Resource can be annotated by more than one RDF_Annotation_Object (i.e. when different RDF schemas are used to define the metadata for a specific WWW resource).

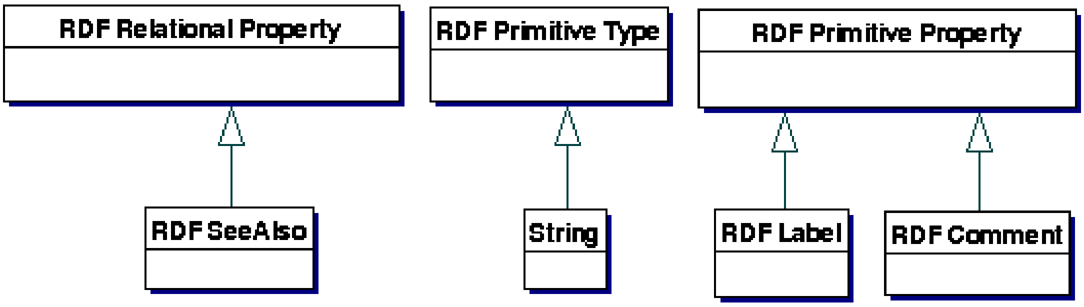


Figure 5: *Predefined properties and types of the alternative RDF Schema Specification*

If we look at the class hierarchy in Figure 2 and compare it to our alternative specification, we have the following additional differences:

- We have not modelled, that an rdfs:Class can be instance of itself. It is not quite clear from the RDF Schema Specification, in which circumstances this additional possibility should be used, and also is suspiciously close to Russels paradox (no set can be element of itself).
- The rdfs:subClassOf relationship between rdfs:Resource and rdfs:Class and rdfs:Property is not modelled. As it is used in the RDF Schema Specification only when specifying domains and ranges (as rdfs:Resource), we have substituted rdfs:Resource in these cases with more specific classes (RDF_Class etc.)

- We have substituted the rdf:type relationship between rdfs:Class and rdfs:Resource with RDF_Annotates and RDF_Defined_By.

Specific types and properties defined in an RDF schema are subclasses and subproperties of RDF_Class and RDF_Property, so more specific properties have more specific classes for their RDF_Range and RDF_Domain attributes.

Figure 5 shows several predefined properties and types, which can be used in RDF schemas. These are subclasses of RDF_Relational_Property, RDF_Primitive_Property and RDF_Primitive_Type. Additional subclasses can be defined as needed (as will be seen in Section 4).

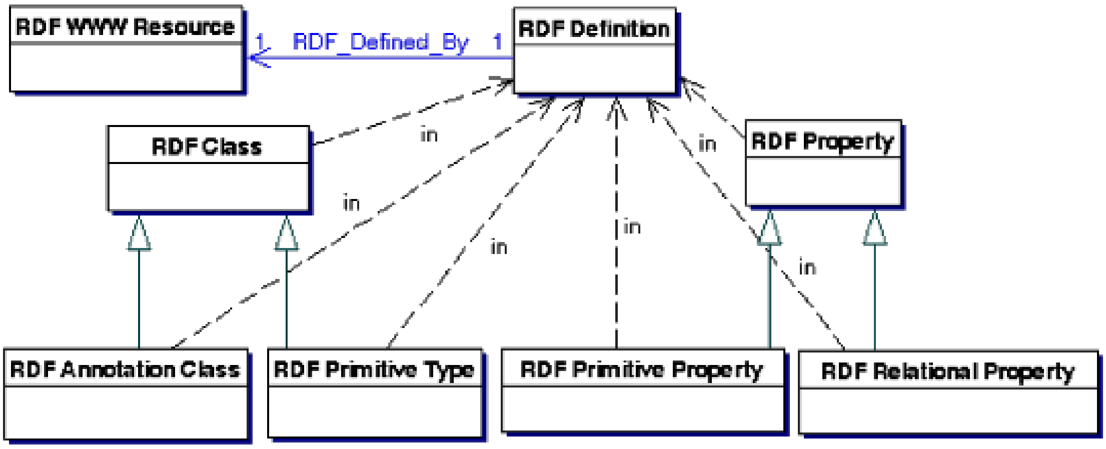


***Figure 6:** Basic RDF Constructs as Instances of RDF_Definition*

Figure 6 shows, that RDF schema constructs (RDF_Class, RDF_Property, etc.) are instances of the class „RDF_Definition“, which is connected to the class RDF_WWW_Resource by the specific relation RDF_Defined_By (i.e. each RDF_Definition is itself defined by a WWW resource).

We think, that this alternative RDF schema specification achieves basically the same goals as the original specification, but makes it much easier to read and to formalize the RDF meta model because of its clear distinction between primitive constructs used for specifying this meta model and other „conventional“ properties, and its use of different relations, whenever different meanings have to be represented.

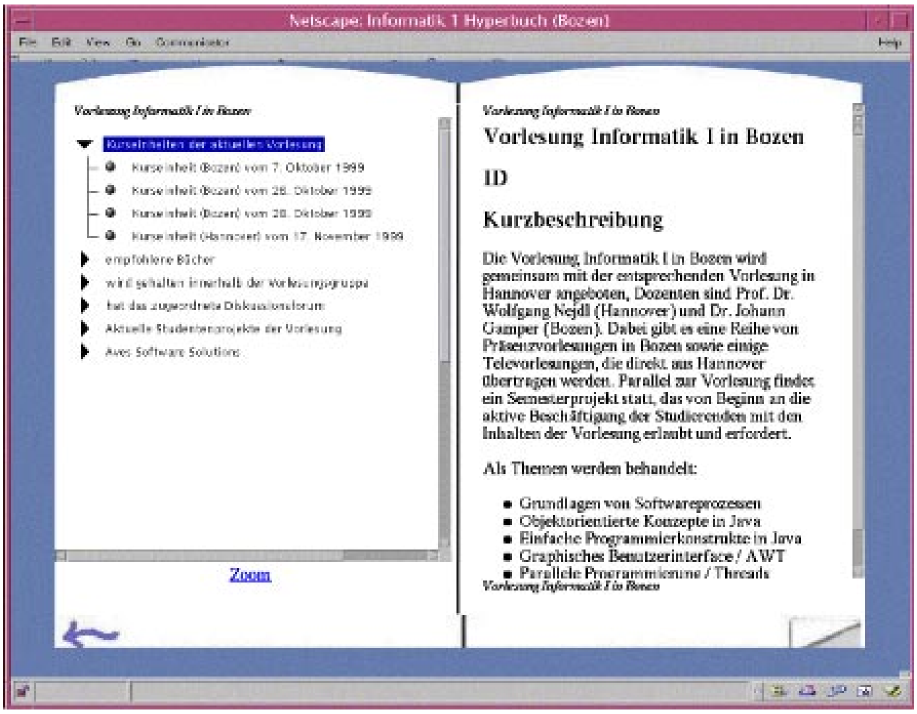
5 Using the KBS Hyperbook for Visualizing Semantic Relationships

Finally, we show how we visualize semantic relationships expressed as RDF-like metadata specified according to our alternative specification described in the previous chapters using the KBS Hyperbook System [NeWo 1999, HNN+ 1999]. Such a hyperbook [HNN+ 1999] is defined as follows:

> A Hyperbook is an information repository, which integrates a set of (possibly distributed) information sources using explicit semantic models and metadata.

Following this definition a Hyperbook uses semantic models and meta data to structure the documents. The KBS Hyperbook (as described in [HNN+ 1999]) employs a general representation model which defines the basic constructs of each hyperbook

Additional presentation classes govern the document layout. Semantic relationships between WWW pages can be modeled according to the representation model and displayed as indices, links or sequences, together with the corresponding WWW page.


Netscape: Informatik 1 Hyperbuch (Bozen)
File Edit View Go Communicator Help
Zoom

Vorlesung Informatik I in Bozen

- Kurseinheiten der aktuellen Vorlesung
  - Kurseinheit (Bozen) vom 7. Oktober 1999
  - Kurseinheit (Bozen) vom 28. Oktober 1999
  - Kurseinheit (Bozen) vom 28. Oktober 1999
  - Kurseinheit (Hannover) vom 17. November 1999
- empfohlene Bücher
- wird gehalten innerhalb der Vorlesungsgruppe
- hat das zugeordnete Diskussionsforum
- Aktuelle Studentenprojekte der Vorlesung
- Aves Software Solutions

Vorlesung Informatik I in Bozen

Vorlesung Informatik I in Bozen

ID

Kurzbeschreibung

Die Vorlesung Informatik I in Bozen wird gemeinsam mit der entsprechenden Vorlesung in Hannover angeboten, Dozenten sind Prof. Dr. Wolfgang Nejdl (Hannover) und Dr. Johann Gamper (Bozen). Dabei gibt es eine Reihe von Präsenzvorlesungen in Bozen sowie einige Televorlesungen, die direkt aus Hannover übertragen werden. Parallel zur Vorlesung findet ein Semesterprojekt statt, das von Beginn an die aktive Beschäftigung der Studierenden mit den Inhalten der Vorlesung erlaubt und erfordert.

Als Themen werden behandelt:

- Grundlagen von Softwareprozessen
- Objektorientierte Konzepte in Java
- Einfache Programmierkonstrukte in Java
- Graphisches Benutzerinterface / AWT
- Parallele Programmierung / Threads

Vorlesung Informatik I in Bozen

***Figure 7:** The KBS Hyperbook using our RDF Schema Specification*

The alternative RDF Schema definition introduced in the previous sections states such a general representation model. It is therefore possible to use our RDF Schema definition as representation model and display the semantic relationships represented by

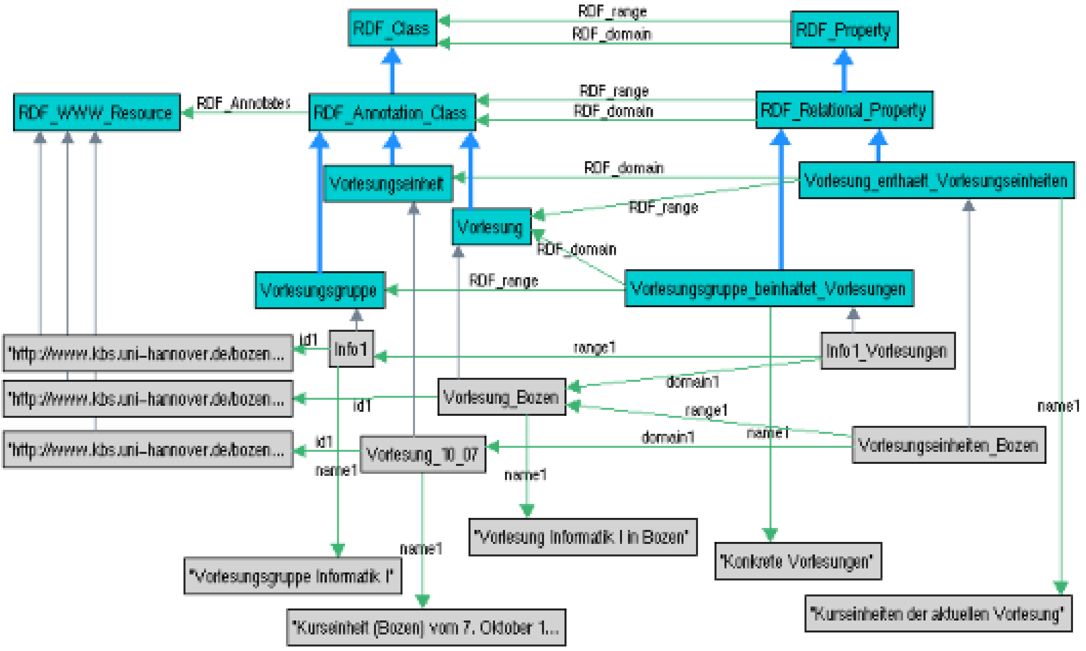

RDF_Class
RDF_range
RDF_domain
RDF_Property
RDF_WWW_Resource
RDF_Annotates
RDF_Annotation_Class
RDF_range
RDF_domain
RDF_Relational_Property
Vorlesungseinheit
RDF_domain
Vorlesung_enthaelt_Vorlesungseinheiten
Vorlesung
RDF_range
RDF_domain
Vorlesungsgruppe
RDF_range
Vorlesungsgruppe_beinhaltet_Vorlesungen
"http://www.kbs.uni-hannover.de/bozen...
id1
Info1
range1
Info1_Vorlesungen
"http://www.kbs.uni-hannover.de/bozen...
id1
Vorlesung_Bozen
domain1
range1
name1
"http://www.kbs.uni-hannover.de/bozen...
id1
name1
Vorlesung_10_07
domain1
name1
Vorlesungseinheiten_Bozen
name1
name1
"Vorlesung Informatik I in Bozen"
"Konkrete Vorlesungen"
"Vorlesungsgruppe Informatik I"
"Kurseinheit (Bozen) vom 7. Oktober 1...
"Kurseinheiten der aktuellen Vorlesung"

***Figure 8:** Part of data model representing part of figure 7*

instancesof RDF_Relational_Property or subclasses thereof. RDF_Relational_Property and its subclasses thus serve as a document structure, which is explicitly modeled by RDF like statements. RDF_Annotation_Class and its subclasses describe the content of the documents. The document itself consists of RDF_WWW_Resources, which reference the documents' content via an URL.

In order to display the documents content as well as the underlying structure in a browser window, the KBS Hyperbook displays in a right frame the documents content and in a left frame the semantical relationships for this current page. Figure 7 shows an example (part of our CS1 lecture material). The lecture material displayed is used for a joint CS1 course in cooperation with the Free University of Bozen in Italy (and in similar form for the corresponding CS1 course in Hannover).

Figure 8 shows part of the underlying metadata and schema for figure 7 in O-Telos graphical notation. The model represents the necessary concepts for displaying the document content as well as the uppermost relation (named „Kurseinheiten der aktuellen Vorlesung") and its first link (named „Kurseinheit (Bozen) vom 7. Oktober 1999"). The figure shows, how our alternative RDF schema specification is used as part of the schema definition for our hypermedia document. As we will see in the following, in the current KBS Hyperbook System we have substituted several constructs (for subclass, primitive property and primitive type) by the corresponding constructs in O-Telos. Other than that, the schema definition used in the hyperbook system is equivalent to the RDF specification model we have proposed in the previous section.

For our CS1 course we have defined additional subclasses of RDF_Annotation_Class like Vorlesungsgruppe (course group), Vorlesung (course) and Vorlesungseinheit (course unit) and subclasses of RDF_Relational_Property like Vorlesungsgruppe_beinhaltet_Vorlesungen (course group contains courses) and Vorlesung_enthaelt_Vorlesungseinheiten (course includes course units). Subclasses and subproperties are both defined using the O-Telos language construct „isA". These subclasses of RDF_Relational_Property structure the (shared) documents used in these courses. Several courses of the same type (in our example the CS1 course held at different universities) reside in the same group (using the „course group contains course" property). A course itself consists of several course units which themselves contain the appropriate text units.

RDF_Primitive_Type and its subclasses are not represented in the figure because the current KBS Hyperbook System uses the primitive types of O-Telos (like String) instead. Additionally, it uses the O-Telos-Class Attribute instead of RDF_Primitive_Property, so primitive attributes are displayed as links in Figure 8. Thus we are able to use the constraint facilities of O-Telos for integrity insurance. The name attribute for relations is instantiated from an additional presentation class „KBS_Displayed_Relation" (specific to the KBS Hyperbook System) and is used for displaying the name of the semantic relationship. We have not included this concept in Figure 8 as it has no corresponding concept in RDF.

Using this schema we are now able to specify the appropriate metadata for our lectures. The figure shows how we group various courses of the same topic, here with the instance „Info1" of „course group". The courses held in Bozen as well as in Hannover are included in this course group. From „Info1" the instantiated relationship „Info1_Vorlesungen" (CS1 courses) connects to the specific course „Vorlesung_Bozen" (course Bozen), and then connect the various course units (here for example

„Vorlesung_10_07“ (course from 10/7/99)) via the relation „Vorlesungseinheiten_Bozen“ (course units Bozen) as shown in figure 7, which is a snapshot from the ConceptBase Browser. Note that the same course units can be used in different courses (e.g. the course units used in the CS1 course in Bozen are partly the same that are used in the CS1 course in Hannover). The instances connected by the name1 relation to the various classes and instances represent the Strings displayed in the browser. id1 is an instantiation of RDF_Annotates.

The current KBS Hyperbook System only supports the display of selected primitive attributes (like name). We are working on an extended version, which also displays the RDF_Primitive_Property instances as an alternative view in the right browser frame.

6 Conclusion and Further Work

In this paper we discussed the proposed RDF Schema Specification from March 1999, and pointed out several pecularities in the RDF schema specification compared to more conventional meta data models, which are basically due to the dual role of the properties rdfs:subclass, rdf:type, rdfs:domain and rdfs:range (used both as primitive constructs in the definition of the RDF schema specification and as specific instances of RDF properties). We then introduced an alternative specification which allowed us to distinguish between modeling and meta modeling features, and make the different meanings of constructs like rdf:type explicit. We specified this alternative using the modeling language O-Telos, and used the KBS Hyperbook system in order to display semantical relationships specified by RDF-like metadata.

We are currently extending the capabilities of the KBS Hyperbook System to visualize these semantical relationships also graphically and to view arbitrary metadata (whose values are not WWW resources, but instances of primitive types like String, Integer, etc).

Bibliography

[CJB 1998] Chandrasekaran, B.; Josephson, J. R.; Benjamins, V. R.: The ontology of tasks and methods. In *Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Alberta, Canada, April 1998.

[FHN 1997] Fröhlich, P.; Henze, N.; Nejdl, W.: Meta-modeling for hypermedia design. In *IEEE Metadata'97*, 1997.

[FNW 1998] Fröhlich, P.; Nejdl, W.; Wolpers, M.: KBS-Hyperbook -an Open Hyperbook System for Education. In *Proceedings of the ED-MEDIA World Conference on Educational Multimedia and Hypermedia*, Freiburg, Germany, June 1998.

[W3C 1998] W3C Working Group. W3c resource description framework (rdf). http://www.w3.org/RDF, October 1998.

[W3C 1999a] W3C Working Group. W3c resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/REC-rdf-syntax/, February 1999.

[W3C 1999b] W3C Working Group. W3c resource description framework (rdf) schema specification. http://www.w3.org/TR/1998/WD-rdf-schema/, March 1999.

[Guar 1998] Guarino, N.: Formal ontology and information systems. In N. Guarino, editor, *Formal Ontology in Information Systems*. IOS Press, 1998.

[HNN+ 1999] Henze, N.; Naceur, K.; Nejdl, W.; Wolpers, M.: Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz*, (4), 1999.

[HeNe 1999] Henze, N.; Nejdl, W.: Student Modeling for the KBS Hyperbook System using Bayesian networks.Technical report, University of Hannover, February 1999. http://www.kbs.uni-hannover.de/paper/99/adaptivity.html.

[IRDS 1990] ISO/IEC. *Information technology - Information Resource Dictionary System (IRDS) framework, ISO/IEC 10027*, 1990 (E).

[Lass 1998] Lassila, O.; Web metadata: A matter of semantics. *IEEE Internet Computing*, (4):30-37, 1998.

[SJJ 1996] Staudt, M.; Jarke, M.; Jeusfeld, M.: Conceptbase v4.1 user manual. Technical report, RWTH Aachen, 1996. http://www-i5.informatik.rwth-aachen.de/CBdoc/userManual/.

[MBJ+ 1990] Mylopoulos, J.; Borgida, A.; Jarke, M.; Koubarakis, M.: Telos: A language for representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4), 1990.

[NeWo 1998] Nejdl, W.; Wolpers, M.: KBS Hyperbook - a data-driven information system on the web. Technical report, University of Hannover, November 1998. http://www.kbs.uni-hannover.de/paper/99/www8.

[NeWo 1999] Nejdl, W.; Wolpers, M.: KBS Hyperbook - a data-driven information system on the web. In *8th International World Wide Web Conference*, Toronto, Canada, May 1999.