Role-Oriented Models for Hypermedia Construction - Conceptual Modeling for the Semantic Web - *

Heidrun Allert, Peter Dolog, Wolfgang Nejdl, Wolf Siberski† Friedrich Steimann‡

ABSTRACT

Current hypermedia modeling techniques such as OOHDM, RMM or WebML rely on class-based domain models to guide the hypermedia construction process. In this paper we discuss the notion of role-oriented models as particularly suited for capturing different semantic contexts of hypermedia interfaces. The key idea is to allow resources to fill different roles which specify how these resources can be embedded in different link structures. After an overview of some class-oriented models, we discuss in detail the main features and benefits of role-oriented modeling compared to conventional class-oriented approaches, and make clear the distinction between attributes and links associated with objects via their natural types and attributes and relationships for an object specified by the different roles the object can fill. Two detailed examples, one discussing the use of role-oriented models for the specification of link structures in learning materials, another one for the specification of contexts for a tourist information system, show how these role-oriented models can be applied in different areas. Finally we show how such models can be easily expressed and implemented using the W3C standard for web annotation, RDFS, and show that a role-oriented semantics is actually closer to the intended use of RDFS schemas than the "default" class-oriented semantics.

1. INTRODUCTION

Complex hypermedia systems, as other systems, need to be designed properly in order to reflect the application domain and its requirements and to provide consistent interfaces suitable for this domain. Hypermedia modeling is therefore an active area of research, and quite a few research groups have contributed to the advancement of the hypermedia construction processes which guide creators of hypertext structures to develop hypermedia with proper design navigation structures. In this context different models for different perspectives of hypermedia application have been studied. The main perspectives [11] for hypermedia models have focused on domain and information models, navigation models and presentation models. These perspectives reflect the main problems the developer has to solve when creating the hypermedia application, i.e. provide the right information, and provide suitable presentation and navigation facilities to indicate other relevant information. In hypertext applications, the main navigation facilities are expressed by links (associations) to other relevant material. Usually, these associations are context specific and are meaningful just in specific contexts.

Most of the current hypermedia modeling methods follow a similar process: developers first create a domain model based on the relevant objects from the application domain, which are classified into classes according to their common features and connected by relationships. Based on this domain model, a navigation model is created, usually as a set of queries over this class model (see e.g. [26, 16]) or by mapping domain model classes, attributes and relationships to navigation model classes, attributes and relationships. Finally, presentation characteristics and different layouts are assigned to navigation and/or conceptual structures.

In this paper we propose a role oriented modeling approach, which extends the domain model by explicitly specifying different contexts expressed as roles and navigational structures based on these roles, thus bridging the gap between the usual class-oriented modeling of the domain and implementation oriented navigation modeling.

In Section 2 we will describe current approaches to hypermedia modeling at the domain level and their connection to navigation design modeling, and discuss the limitations of this approach with respect to specifying different contexts. Section 3 discusses the semantic notions of role types versus natural types and proposes role-oriented models as an extension to class-oriented domain models for hypermedia modeling. Sections 4.1 and 4.2 describe two examples from two different application domains showing the benefits of role-oriented models. As the main characteristic of the Semantic Web is its distributed and heterogeneous nature, we focus in Section 5 on role based annotations, and show that a role-oriented view for the RDFS schemas used to specify such role-based annotations and contexts captures the intended meaning better than a simple class-oriented one.

*This work is partially supported by EU/IST ELENA project IST-2001-37264.

†Learning Lab Lower Saxony, University of Hannover, Expo Plaza 1, 30539 Hannover, Germany, {allert, dolog, nejdl, siberski}@learninglab.de

‡Information Systems Institute, University of Hannover, Appelstrasse 4, D-30167 Hannover, steimann@kbs.uni-hannover.de

2. CLASS ORIENTED MODELING

2.1 Current approaches

Current domain modeling techniques for hypermedia are based mostly on class diagrams and extended entity-relationship diagram approaches. Methods such as the Relationship Management Methodology (RMM) [18] and Hypertext Design Model (HDM) [14] are examples where EERD is used. The RMM uses EERD without modification, and allows to model entity types, which are interconnected by relationships. The attributes are used for modeling structured content. HDM [14] comprises customized EERD and introduces entities, which derive their content from their components (which for example could be sections of documents). Entities represent objects in the real world, and can be grouped into entity types.

Class diagrams allow us to model inheritance, aggregation, association and dependencies between different classes. Similarly to the EERD, classes have attributes. Class diagrams are used for example in the Object-Oriented Hypermedia Design Method (OOHDM) [27, 26], UML based web engineering (UWE) [16]), W2000 [4], or the XML based WebML method proposed in [10]. Again, as with EERD's, the basic assumption is the existence of exactly one domain model, which represents the relevant features of the domain of discourse.

Navigation modeling in these approaches again is based on the notion of classes / types, which then represent the types of nodes in the hypertext system. These node types or rather their instances are then created by queries over the domain model or by mappings of (a subset of) the domain model onto the navigation model primitives. RMM [18] provides us with primitives like indices, indexes, guided tours etc., and the EERD model elements are mapped to these structures in navigation design. HDM [14] provides perspectives for its components and entities in navigation design and the possibility to link them with different type of links. The notion of views is employed in [27]. Views in OOHDM are similar to database views and are created by queries using the classes and/or relationships specified in the domain model. The navigation classes can be grouped, indexed and structured into contexts, the main elements are *context classes*, *class groups*, access structures such as *indexes*, *slices*, *menus* and *guided tours* [25]. UWE [20] uses class diagrams for specifying navigation and follows an approach very similar to OOHDM. Navigation classes can be mapped to indexes, contexts, guided tours, menus and queries. (WebML) [10] provides us with the possibility to compose classes and attributes from the domain model into a composition model and then relate components from this model using links.

2.2 Discussion

A unifying characteristic of all approaches described in the previous section is that they rely on one underlying domain model and completely separate domain analysis from navigation design. An author models all relevant types of information and their relationships, trying to cover all characteristics of the domain in one model, and then derives navigation types and links, as well as presentation characteristics like trails and indices, derived from the domain model specified in the first step.

However, the lack of support for logical modeling of navigation is apparent. Hypermedia applications could definitively benefit from an approach which takes navigation issues into account already at the domain level (though of course the decision for specific navigation constructs like trails and indices will always be a separate second step). Classical type oriented modeling techniques are not sufficient for these purposes. The problem is that we are not able to classify different requirements for linking information types. We can only say that one information type is connected to other types by specific relationships, but these relationships are stated in one common domain model, making it difficult to distinguish between different contexts.

We therefore introduce role-oriented modeling to specify these contexts in the domain model, where each role specifies a certain context plus the associated relationships and attributes. In a way, such role models can be seen as independent conceptual models, and for each context we can have one role model, plus possibly one (usually quite simple) class-based model specifying characteristics common to all contexts. This additional dimension in conceptual modeling, which reflects navigational contexts already in the domain model, can lead to clearer and more focused models, and to conceptual models which are better customized to different domain contexts.

We are aware of one first attempt to deal with navigation at this level, namely [9]. The authors define the navigation model as a net of semantic nodes interconnected by semantic links. The semantic nodes are nodes, which can be considered as prototypes of a user interface bound to some semantic information. Semantic links denote the links a user can follow from that semantic node. The navigation design modeling is then driven by this analytical navigation model. However, [9] define this model independently from the conceptual model of the domain. In contrast, we argue that the conceptual model and the navigation model at the domain level are related through roles, and it is more natural to model links as associations, which appear in the specific contexts of roles in the user interface. This also allows us to derive these roles directly from tasks in a particular domain and to better capture diverse requirements for navigation raised by different roles.

3. ROLE ORIENTED MODELING

3.1 Natural Types versus Role Types

Classification is perhaps the oldest human concept for organizing the universe. However, classification is an utterly artificial construct which, although capable of structuring nature, is itself not found in nature [21, 30]. Instead, it occurs that the concept of a prototype, together with Wittgenstein's family resemblances, more adequately abstracts from and orders individual entities of which there is such a huge diversity. This view appears particularly apt for the classification of content which, by its nature, varies from exponent to exponent and, as regards to classification, is only trivially described by type information in the form of a predefined set of attributes such as title, author, date, etc. However, introducing prototypes into modeling is a difficult decision exactly because prototypes are by definition not strict specifications, but rather "fuzzy" classifications difficult to capture formally. Nevertheless some kind of classification of content seems to be indispensable, if only for pragmatic reasons.

Ontology has made a clear distinction between what has been termed a *natural type* (corresponding to the "usual" classification construct) and a *role type*. This distinction relies on two fundamental properties of types, namely *foundedness* and *semantic rigidity* (see e.g. [15]).

According to this theory, a type is founded if for an entity to belong to (the extension of) the type, it must stand in some relationship (other than aggregation[1]) to another entity. For instance, Reader is a founded type since for a person (or piece of machinery) to be a Reader, there must be something being read (a defining characteristic of Reader). Conversely, Book is not founded, since a Book is a Book in its own terms, independent of it being read or used otherwise.

A type is semantically rigid, on the other hand, if for any entity belonging to (the extension of) the type, the entity cannot drop this type without losing its identity. Book for instance is semantically rigid, since a Book cannot stop being a Book without losing its identity. Reader on the other hand is not semantically rigid, since a Reader can stop reading (thereby leaving the extension of Reader) without losing its identity.

Based on these distinctions, we can now talk about a role type as being defined as a founded and not semantically rigid type. Conversely, a natural type is defined as semantically rigid, but not founded. Reader is a role type, Book is a natural type[2].

Interestingly, this ontological definition is paralleled by a linguistic one: in Fillmore's case grammar [13], each predicate (corresponding to a relationship) comes with a number of semantic roles to be filled by the complements of the predicate. The verb "to rob" for instance comes with the semantic roles Culprit, Victim, and Loot. These roles classify entities in the context of the predicate: they are founded, but not semantically rigid.

3.2 Definitions and Meta Model

To sum things up, the distinction between natural and role types may be formulated as follows:

- A type is a *role type* if
  - for an object to belong to the extension of the type it must engage in a relationship associated with the type, and
  - entering or leaving the extension of the type does not alter the objects identity.
- A type is a natural type if
  - belonging to the type is independent of being engaged in a relationship (except for, perhaps, whole-part; see below) and
  - an object cannot leave the extension of the type without losing its identity.

The distinction between role types and natural types on the one hand and the dependence of role types on relationships on the other can be formalized as follows. Let there be two sets, N and R, of natural type symbols and role type symbols. The elements of N are partially ordered by the subclass relationship $\leq_{NN}$, and the elements of R are ordered by the subrole relationship $\leq_{RR}$. The relationships form separate type subsumption hierarchies, which are connected only by a third relationship, $<_{NR}$, which specifies the instances of which classes can occur where which roles are specified. If one adds to this definition a family of predicate symbols $(P)_w$ with w being a sequence of two or more role symbols so that each element of $(P)_w$ stands for a predicate with the roles of w declared as its places, the metamodel depicted in fig. 1 describes the situation.

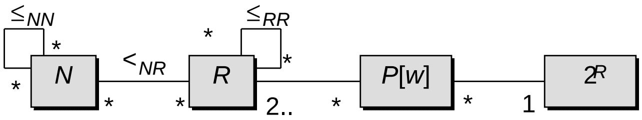

Figure 1: Metamodel defining the relationship between natural types, role types, and relationships (in UML class diagram noatation).

Compared to the approaches discussed in the previous section, our metamodel allows relationships to be defined solely between roles. This does not mention aggregation (which is not a founding relationship), which may impose structure on the instances of natural types. For instance, a document may be defined to consist of chapters and sections. Note that in our hypertext context the term role is not used to denote the different roles of any entities outside the domain relevant for the hypertext system itself, as for instance the different roles of a hypertext reader (eg, Novice, Expert) in the context of adaptivity or the different roles involved in its creation (eg Author, Publisher, etc.)

4. ROLE MODELING EXAMPLES

4.1 Example 1: Learning Objects

In the field of learning and learning materials, the distinction between natural and role types is indeed an important one, but has been neglected by most conceptual models defined for this domain. One prominent example is the Standard for Learning Objects Metadata (LOM, [17]), which describes the attributes and relationships of learning objects. This standard focuses on the type of learning object, and uses additional (often implicit) types, when it comes to describing the attributes and relationships of learning objects. If we look at the LOM standard from a conceptual modeling perspective, we realize that the missing distinction between role types and natural types has lead to a number of modeling errors, which make the unambiguous use of some LOM attributes unnecessarily difficult.

For example, the LOM category Educational (which serves to describe educational attributes and relationships of a learning object) uses several types to constrain the educational attributes/relationships. The data element Learning Resource Type comes with the following set of values: Exercise, Simulation, Questionnaire, Diagram, Figure, Graph, Index, Slide, Table, Narrative Text, Exam, Experiment, Problem Statement, and Self Assessment. In trying to apply or even order these types, the reader soon realizes that there are some fundamental differences between these values. For instance, a narrative text can be a problem statement, but not a figure; a questionnaire can be an exam and/or a self assessment, but not a graph etc. More detailed thinking about this problem leads us to the realization that the problem is rooted in the missing distinction of the types these values represent, and moreover, that some of these types are natural types, while others are

[1]Literally all objects are aggregated from smaller parts. Thus, if foundedness were grounded on aggregation, it would be a meaningless concept.

[2]Note that being founded and being semantically rigid are no absolute properties, but depend on the application domain. For instance, in a hypertext context, Reader may be used synonymously for Person and hence be semantically rigid. However, the ontological distinction remains valid; only the universe it is applied in changes.

role types. For instance, Simulation, Questionnaire, Diagram, Figure, Graph, Index, Slide, Table, and Narrative Text, should most probably be modeled as natural types, whereas Exercise, Exam, Problem Statement, Self Assessment and Experiment seem to be role types[3] Also, one realizes that only certain natural types can serve certain roles, i.e. there appears to be some static relationship between natural and role types.

Learning Management Systems and Learning Environments often implement specific learning theories. These learning theories are addressed either implicitly or (sometimes) explicitly, and therefore also specific assumptions on learning are addressed implicitly or explicitly. If we follow the approach of role-oriented modeling discussed in the previous section, learning objects are natural types which can fill different roles within different pedagogical approaches or learning theories.

Modeling these roles explicitly allows us to design learning repositories and authoring tools which explicitly implement different learning theories, and thus can guide authors in the design of learning sequences conforming to these specific learning theories. Additionally, these learning designs can be explicitly reflected in the interface design for the learning repository, thus providing the learner with useful hints and the ability to choose between different learning styles represented by different navigational structures [1, 2]. The learner can toggle for example between the interfaces called "expository learning" and "problem-based learning". Each interface is designed along a specific learning model. Therefore each of them presents a specific navigational concept and specific links relevant for this learning theory. Here we exemplify two different learning theories, which are based on different epistemologies:

- *Expository Learning.* This learning theory is based on information processing theories and the assumptions of cognitivism. According to models of Expository Teaching [3] and Discovery Learning [8] meaningful learning has to be encouraged. Material has to be presented in a carefully organized form. Learning occurs when there is a fit between the student's schemas and the material to be learned.

- *Problem-Based Learning, solving ill-structured problems.* This learning model focuses on process-oriented learning. It is based on the theories of situated cognition [19]. The assumption on learning is well described by Seufert: Knowledge is interlinked by means of questions arising from business practice and scientific research. Only such inter-linked knowledge can consciously be used in non-school-related situations. [28].

Many learning objects can be used within both models, expository and problem-based, whereas some meet only criteria specified by either model.

Learning Objects in Expository Learning Situations. A learning object which is a text file presenting a theory on "cirrhosis of liver" is a natural type. In the expository model this learning object fills the role "Concept" or "Concept Definition". The customized interface presents four links (or types of links) as described in the following:

[3]What exactly is a role and what is a type depends on the ontological criteria listed above, which cannot be evaluated as long as the exact definitions of these terms are unclear.

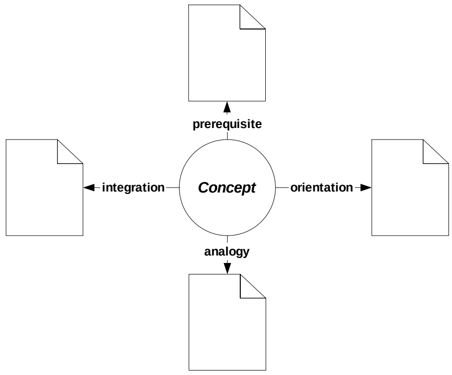

Figure 2: Expository Model (Expository Learning)

- *Prerequisite*. Learners who follow these links will find learning objects presenting knowledge they need to learn in order to understand the theory on "cirrhosis of liver" .

- *Orientation*. Learners who follow these links will find learning objects giving examples, pointing out important issues, asking questions, specifying learning objectives. The purpose is to provide scaffolding and support for the new information which is given by the theory on "cirrhosis of liver".

- *Analogy*. Learners who follow these links will find learning objects which present a similar theory on the same topic. This is relevant for learning as the learning theory recommends to focus on analogies, similarities and differences.

- *Integration*. Learners who follow these links will find learning objects presenting scientific reports and other supporting real-world material. It is important to mention that these reports do not only apply the theory on "cirrhosis of liver" in particular, but also several other theories, as cirrhosis of liver often comes along with for example varicosities of the gastric mucosa. The purpose is to support integration of newly acquired knowledge and connection to other fields of knowledge.

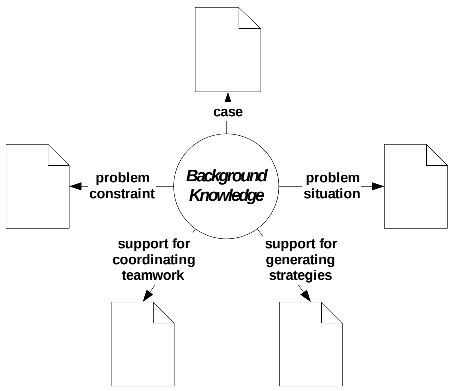

Figure 3: Problem-Based Model (Solving Ill-Structured Problems)

Learning Objects in Problem-Based Learning Situations. The same learning object (the text file presenting a theory on "cirrhosis of liver" fills the role "Background Knowledge" in the problem-based model. The interface now presents a set of different links such as

- *Case*. Learners who follow these links will find learning objects presenting real cases. These cases present different perspectives on the same topic written by patients, doctors, relatives etc., or cases collected by peer-learners.

- *Problem situation* and *Problem constraints*. Here learners find the description of a problem situation which meets the criteria of ill-structured problems: it gives incomplete and inaccurate or ambiguous information as real world problems are ill-structured as well.

- *Support facilities*. Here learners find comments, learning objects and other entities which support argument construction, generating ideas, regulation of cognition, evaluation of solution, development of justification, etc. [29].

Criteria of which learning object can fill a certain role are given by the specific learning theories, and in this way different theories can guide the design of the different interfaces. The author as well as the learner is asked to choose a specific learning model to construct or to use the learning material collection. For the learner, this has the additional benefit that he has to reflect on his preferred way of learning and on different learning strategies and becomes aware of their differences. This metacognitive knowledge is definitively relevant for a life long learner who can not only decide what he wants to learn but also how to learn it.

4.2 Example 2: Travel Information System

Another nice example is the modeling for different interfaces in a travel information system based on different user groups with different background and expectations. In the tourism industry it is very common to bundle transportation, accommodation and additional offers (as day tours or event tickets) in a travel package. However, there are also travelers who prefer to arrange their holidays themselves, often by combining products from different companies. Some agencies have responded to this customer group with a special offer: Customers can compile their own package from selected product components, and book this selection like a predefined package in a second step.

A system suitable for both user groups clearly has to provide two different interfaces to (the same or overlapping) information items, motivated by and based on the role-oriented model specifying the requirements of each group. In such a system, parts of the product description can be reused for all groups. For example the description of a day tour to a tourist attraction (e.g. boat tour on the Nile) can be used as an independent (stand-alone) product offer, in the description of a predefined package or as description of an optional part of a customizable package. But depending on the context the documents plays different roles:

If the tour is offered as stand-alone product, the customer needs information about prices, terms and conditions, as well as a booking facility. He also needs instructions about how to get there and when and where to meet (see Figure 5).

As part of a predefined package, this information is not necessary, because the tour operator cares for these things. Also commercial information can be omitted, because it will be available for the complete package. This context is depicted in Figure 6.

Another possibility is when this information is offered as one element of a customizable package. In this case we need the back link

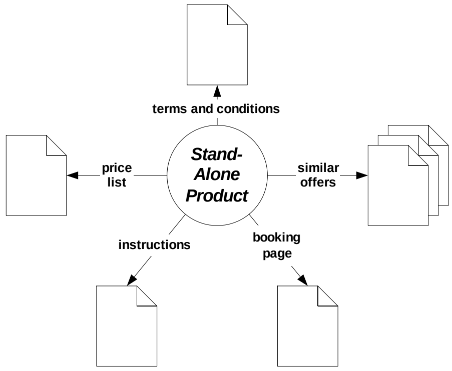

Figure 5: Stand-alone product role

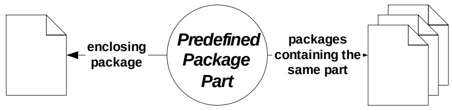

Figure 6: Predefined package part role

to the current (customized) package, and may also show additional offers which fit into the current package configuration, as shown in Figure 7.

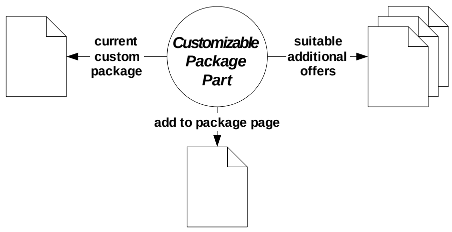

Figure 7: Customizable package part role

Merely using the natural type "day tour description" for classification would not have been sufficient for modeling these different navigational contexts.

5. ROLE-BASED SCHEMAS AND ANNOTATION IN THE SEMANTIC WEB

5.1 Role Types in RDF(S)

As part of the Semantic Web initiative of the W3C [5], the Resource Description Format (RDF, [22]) and RDF Schema (RDFS, [7]) have been introduced as standards for annotation of web resources with metadata. Although the designers of RDF(S) didn't have in mind the distinction between classes and roles, the RDFS type system is very suitable to represent role types as discussed in this paper. RDF schemas define types and properties for the purpose of annotating resources. Properties can be either attributes of the resource (e.g. title) or relations to other resources, types are used to constrain the objects which can be used in these relationships.

Indeed, role types are more appropriate for giving meaning to the RDF(S) type construct than are natural types. If we look at the RDF type model, we realize that it is purely extensional: RDF types cannot be instantiated (a technical procedure by which a new instance of the type would be created and equipped with attributes prescribed by the type). Instead, existing instances (i.e., resources)

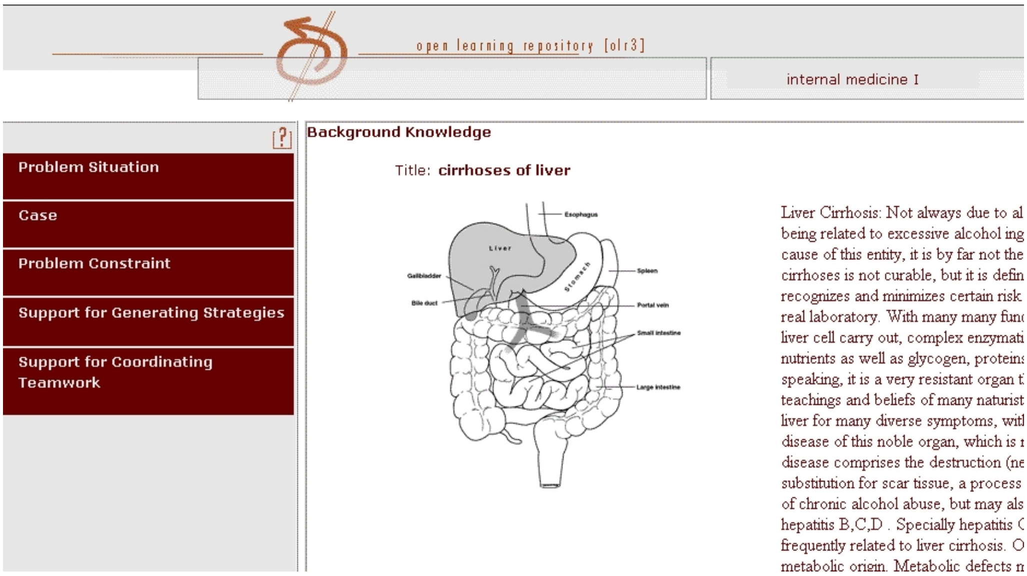

Figure 4: Screenshot of a user interface prototype which the learner sees when choosing 'Problem-Based Learning'. The learning object presented fills the role 'Background Knowledge'. The user can access the role specific links on the left hand.

of arbitrary natural type (whatever this would be for a resource) are declared to belong to a predefined (role) type by annotating them with an RDF type statement. These annotations can be changed without affecting the identity of the annotated resource (for the least, they are established at certain points in time and not pre-existing), so that they should not be considered semantically rigid. RDF does not explicitly support the definition of founded types, but they can be introduced by restricting annotations to relations. Resources can have arbitrarily many types; therefore they can be assigned to as many roles as necessary.

In RDFS it is possible to add domain (relation source) and range (relation destination) declarations for properties. In line with our discussions in this paper we use these declarations to denote which roles are required at the relation ends. This ensures that only links consistent with the role model can be specified. Specifying a role model in RDF is then very straightforward. For each role, define an RDF type; for each link, define a property. For example, a schema for the problem based learning model would look as follows. We show a short extract only which models the *case* link in problem based learning:

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<!-- RDF Schema for Problem based learning -->

<!DOCTYPE rdf:RDF [
    <!ENTITY pbl "http://example.org/pbl#">
]>

<rdf:RDF xml:lang="en"
  xmlns:rdf=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs=
    "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:pbl="&pbl;">

<rdfs:Class rdf:about="&pbl;BackgroundKnowledge">
  <rdfs:comment>
    Learning Object used as Background Knowledge in PBL
  </rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="&pbl;CaseDescription">
  <rdfs:comment>
    Description of a case for the problem
  </rdfs:comment>
</rdfs:Class>

<rdfs:Property rdf:about="&pbl;case">
  <rdfs:domain
    rdf:resource="&pbl;BackgroundKnowledge"/>
  <rdfs:range
    rdf:resource="&pbl;CaseDescription"/>
  <rdfs:comment>
    Link from Background Knowledge to Problem Case
  </rdfs:comment>
</rdfs:Property>
...

</rdf:RDF>
```

Based on this schema we describe some *case* links available from the page shown in figure 4 (namespace declarations omitted):

```
<pbl:BackgroundKnowledge rdf:about=
  "http://med.example.org/liver/cirrosis/bk.html">
  <pbl:hasCase rdf:resource=
    "http://med.mc.ntu.edu.tw/~fm/intern/I04211999.htm"/>
  <pbl:hasCase rdf:resource=
    "http://www.nurse.itan.../VirtualCaseStudy.htm"/>
  <pbl:hasCase rdf:resource=
    "http://www.ctisus.../liver_cirrhosis2000_1.html"/>
  <pbl:hasCase rdf:resource=
    "http://bob.usuf2.usuhs.mil/.../cases/node23.html"/>
</pbl:BackgroundKnowledge>
```

5.2 Distributing Schemas and Annotations

Schemas in the Semantic Web are referenced by URIs, and can be stored anywhere on the Web. Additionally, an important characteristic of RDF metadata is the ability to use distributed annotations for one and the same resource, possibly using more than one of these schemas. So, in contrast to traditional systems, it is not necessary that all annotations of a resource are stored on one server, but rather allows us to have local annotations for globally accessible resources without requiring write access to the system that owns the resource. In our case, link information can be stored at several places and be collected on-demand. Thus third parties can add new links at any time, provided they comply with the role model constraints. The resulting hypertext system is based on role-based annotations, roles and navigational context for these roles is defined in (distributed) schemas, and links can be stored on (distributed) servers and retrieved and added dynamically.

It is then straightforward to generalize this distributed client/server scenario to a peer-to-peer one, where hypertext pages are stored on and used by independent peers, and retrieved by using queries against this network. During the last two years we have been working on such an infrastructure, Edutella, [23, 24], and indeed, a (possibly adaptive) hypermedia system based on RDF-annotated pages, appropriate queries and possibly user models (see [12]) seems to be a very interesting generalization of centralized hypermedia systems.

In the Edutella network, each peer can make its metadata information available to all other peers as a set of RDF statements, suitable for describing his own (and possibly other distributed) resources. Queries for annotations are distributed within the P2P network, and all results are collected and aggregated by Edutella super peers [24] and sent to the client who wants to use them. This also seems to fit well into discussions on this topic in recent years, [31, 6], by providing a suitable and extensible infrastructure for peer-to-peer-based hypermedia systems.

6. CONCLUSIONS AND FURTHER WORK

In this paper we have proposed to extend current hypermedia modeling techniques relying on class-based models to role-based-models, which are able in a natural way to model the use of resources in different contexts associated with different navigational structures. We have discussed the main characteristics and benefits of this approach and exemplified its use in two different application areas. Finally, we have analyzed the suitability of role-oriented models to give meaning to the Semantic Web standard RDF(S) and have pointed the way to distributed hypermedia systems in an RDF(S)-based Peer-to-Peer Network. Future work will refine both modeling techniques for such a distributed environment as well as the underlying peer-to-peer infrastructure, and the numerous issues arising from such a distributed hypermedia architecture.

7. REFERENCES

[1] H. Allert, H. Dhraief, and W. Nejdl. How are learning objects used in learning processes? instructional roles of learning objects in lom. In *ED-MEDIA 2002, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Denver, Colorado, June 2002.

[2] H. Allert, C. Richter, and W. Nejdl. Learning objects and the semantic web - explicitly modelling instructional theories and paradigms. In *AACE E-Learn World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*, Montreal, Canada, Oct. 2002.

[3] D. P. Ausubel. *Educational Psychology: A Cognitive View*. Holt, Rinehart and Winston, New York, NY, USA, 1968.

[4] L. Baresi, F. Garzotto, and P. Paolini. Extending UML for modeling web applications. In *Proc. of 34th Anual Hawaii International Conference on System Sciences (HICSS'34)*, Maui, Hawai, Jan. 2001. IEEE Press.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[6] N. O. Bouvin. Open hypermedia in a peer-to-peer context. In *Proc. of the Thirteenth Conference on Hypertext and Hypermedia*, Washington, USA, June 2002. ACM Press.

[7] D. Brickley and R. V. Guha. Resource Description Framework (RDF) Schema Specification 1.0, 2002. `http://www.w3.org/TR/rdf-schema`.

[8] J. S. Bruner. *Toward a Theory of Instruction*. W.W. Norton, New York, NY, USA, 1966.

[9] C. Cachero and N. Koch. Conceptual navigation analysis: a device and platform independent navigation specification. In D. Schwabe, O. Pastor, G. Rossi, and L. Olsina, editors, *Proc. of Second International Workshop on Web-oriented Software Technology (IWWOST02)*, June 2002.

[10] S. Ceri, P. Fraternali, and A. Bongio. Web Modeling Language (WebML): a modeling language for designing web sites. *Computer Networks and ISDN Systems*, 33(1–6):137–157, June 2000.

[11] P. Dolog and M. Bieliková. Hypermedia systems modelling framework. *Computing and Informatics*, 21(3):221–239, Dec. 2002.

[12] P. Dolog, R. Gavriloaie, W. Nejdl, and J. Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 2003. to appear.

[13] C. J. Filmore. Types of lexical information. In D. D. Steinberg and L. A. Jakobovits, editors, *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, pages 370–392. Cambridge University Press, 1971.

[14] F. Garzotto and P. Paolini. HDM — a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, Jan. 1993.

[15] N. Guarino, M. Carrara, and P. Giaretta. An ontology of meta-level categories. In *Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning*, pages 270–280. Morgan Kaufmann, San Mateo, 1994.

[16] R. Hennicker and N. Koch. A UML-based methodology for hypermedia design. In S. S. A. Evans and B. Selic, editors, *Proc. of UML 2000 Conference*, York, England, Oct. 2000. Springer LNCS 1939.

[17] IEEE Learning Technology Standards Committee. IEEE LOM Working Draft 6.1 (IEEE 1484.12.1–2002). http://ltsc.ieee.org/wg12/index.html, Apr. 2001.

[18] T. Isakowitz, A. Kamis, and M. Koufaris. Extending the capabilities of RMM: Russians dolls and hypertext. In *Proceedings of the 30th Annual Hawaii International Conference on System Sciences*, Jan. 1997.

[19] D. H. Jonassen. Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology: Research and Development*, 45(1):65–95, 1997.

[20] N. Koch, H. Baumeister, R. Hennicker, and L. Mandel. Extending UML for modeling navigation and presentation in web applications. In G. Winters and J. Winters, editors, *Proc. of Modeling Web Applications in the UML Workshop, UML 2000 conference*, York, England, Oct. 2000.

[21] G. Lakoff. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. The Chicago University Press, Chicago, 1987.

[22] O. Lassila and R. Swick. W3C Resource Description framework (RDF) Model and Syntax Specification, 2001. `http://www.w3.org/TR/REC-rdf-syntax/`.

[23] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmr, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *WWW 11 Conference Proceedings*, Hawaii, USA, May 2002.

[24] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Lser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 2003. to appear.

[25] D. Schwabe, R. de Almeida Pontes, and I. Moura. OOHDM-Web: An environment for implementation of hypermedia applications in the WWW. *SIGWEB Newsletter*, 8(2), June 1999.

[26] D. Schwabe and G. Rossi. An object-oriented approach to web-based application design. *Theory and Practise of Object Systems (TAPOS), Special Issue on the Internet*, 4(4):207–225, Oct. 1998.

[27] D. Schwabe, G. Rossi, and S. D. J. Barbosa. Systematic hypermedia application design with OOHDM. In *Proc. of the ACM International Conference on Hypertext (Hypertext'96)*, pages 116–128, Washington, USA, Mar. 1996. ACM Press.

[28] S. Seufert, U. Lechner, and K. Stanoevska. A reference model for online learning communitiess. *International Journal on E-Learning (IJEL)*, 1(1), 2002.

[29] N. Shin Hong. The relationship between well-structured and ill-structured problem solving in multimedia simulation. Pennsylvania State University, College of Education, Doctoral Thesis, 1998.

[30] F. Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data and Knowledge Engineering*, 35(1):83–106, 2000.

[31] U. K. Wiil, N. O. Bouvin, D. Larsen, D. C. D. Roure, and M. K. Thompson. Peer-to-peer hypertext (panel). In *Proc. of the Thirteenth Conference on Hypertext and Hypermedia*, Washington, USA, June 2002. ACM Press.