

Reasoning and Ontologies for Personalized E-Learning in the Semantic Web

Nicola Henze¹, Peter Dolog², and Wolfgang Nejdl^{1,2}

¹*ISI- Knowledge-Based Systems,*

University of Hannover, Appelstr. 4, D-30167 Hannover, Germany

{henze,nejdl}@kbs.uni-hannover.de

²*L3S Research Center,*

University of Hannover, Expo Plaza 1, D-30539 Hannover, Germany

{dolog,nejdl}@learninglab.de

Abstract:

The challenge of the semantic web is the provision of distributed information with well defined meaning, understandable for different parties. Particularly, applications should be able to provide individually optimized access to information by taking the individual needs and requirements of the users into account. In this paper we propose a framework for personalized e-Learning in the semantic web and show how the semantic web resource description formats can be utilized for automatic generation of hypertext structures from distributed metadata. Ontologies and metadata for three types of resources (domain, user, and observation) are investigated. We investigate a logic-based approach to educational hypermedia using TRIPLE, a rule and query language for the semantic web.

keywords:

Educational hypermedia, Semantic web, Ontologies, Adaptive hypermedia, Reasoning on the semantic web.

Introduction

The vision of the semantic web is to enable machines to interpret and process information in the world wide web in order to better support humans in carrying out their various tasks with the web. Several technologies have been developed for shaping, constructing and developing the semantic web. Many of the so far developed semantic web technologies provide us with tools for describing and annotating resources on the web in standardized ways, e.g. with the Resource Description Framework (RDF [[RDF, 2002](#)]) and its binding to XML (eXtensible Markup Language [[XML, 2003](#)]). In this paper we will show how semantic web technologies and in particular ontologies can be used for building adaptive educational hypermedia systems. Adaptive educational hypermedia systems are able to adapt various visible aspects of the hypermedia systems to the individual requirements of the learners and are very promising tools in the area of e-Learning: Especially in the area of e-Learning it is important to take the different needs of learners into account in order to propose learning goals, learning paths, help students in orienting in the e-Learning systems and support them during their learning progress.

We propose a framework for such adaptive or personalized educational hypermedia systems for the semantic web. The aim of this approach is to facilitate the development of an adaptive web as envisioned e.g. in [[Brusilovsky and Maybury, 2002](#)]. In particular, we show how rules can be enabled to reason over distributed information resources in order to dynamically derive hypertext relations. On the web, information can be found in various resources (e.g. documents), in annotation of these resources (like RDF-annotations on the documents themselves), in metadata files (like RDF descriptions), or in ontologies. Based on these sources of information we can think of functionality allowing us to derive new relations between information.

Imagine the following situation: You are currently writing e-Learning materials for higher education. Especially in e-Learning, it is important to overcome the *one-size-fits-all* approach and provide learners with individual learning experiences. Learners have different requirements (like their individual learning style, their actual progress in the learning process, their individual background knowledge, but also more technical requirements like the device they are currently using for accessing the E-Learning materials, etc.). The e-Learning system you would like to use should provide such a personalized delivery of e-Learning materials. How can you describe instructional material in a way allowing for personalized e-Learning?

In our solution for personalized e-Learning systems we envision personal learning services capable of interpreting metadata-annotated learning resource, *understanding* their annotations with respect to standard ontologies for learning materials like e.g. LOM [[LOM, 2002](#)] or IMS [[IMS, 2002](#)]), and also with respect to specific domain ontologies which describe the particular subject being taught. To enable personalized delivery of the learning resources, ontologies for describing the learner and observations about the learner's interactions with the e-Learning system are required to characterize and model a learners current profile.

Each personal learning service possess reasoning rules for some specific adaptation purposes. These rules query for resources and metadata, and reason over distributed data and metadata descriptions. A major step for reasoning after having queried user profile, domain ontology and learning objects is to construct a temporally valid task knowledge base as a base for applying the adaptation rules. The concluded results of these personal learning services are described using the presentation format of the open hypermedia standard.

The paper is structured as follows: In the following section, we will compare our approach with related work. Section [3](#) describes the representation of resources with semantic web technologies, and shows our use of a domain, user, and observation ontologies. Section [4](#) discusses our approach to generate hypertext structures / associations, and an example set of rules for dynamically generating personalized associations between information. A comparison of our approach to related work and a conclusion end the paper.

Related Work

To describe and implement personalized e-Learning in the semantic web, there are at least three related research areas which contribute: *open hypermedia*, *adaptive hypermedia*, and *reasoning for the semantic web*. Open hypermedia is an approach to relationship management and information organization for hypertext-like structure servers. Key features are the separation of relationships and content, the integration of third party applications, and advanced hypermedia data models allowing, e.g., the modeling of complex relationships . In open hypermedia, data models like FOHM (Fundamental Open Hypertext Model) [[Millard et al., 2000](#)] and models for describing link exchange formats like OHIF (Open Hypermedia

Interchange format) [[Gronbaek et al., 2000](#)] have been developed. The use of ontologies for open hypermedia has e.g. been discussed in [[Kampa et al., 2001](#)]. Here, an ontology is employed that clarifies the relations of resources. On base of this ontology, inference rules can derive new hypertext relations. In [[Weal et al., 2001](#)] the open hypermedia structures are used as an interface to ontology browsing. The links at the user interface are transformed to queries over ontology. Thus links serves as contexts for particular user.

The question whether conceptual open hypermedia is the semantic web has been discussed in [[Bechhofer et al., 2001](#)]. In [[Carr et al., 2001](#)], a *metadata space* is introduced, where the openness of systems and their use of metadata is compared. On the *metadata dimension* (x-axis), the units are the use of *keywords*, *thesauri*, *ontologies*, and *description logic*. The y-axis describes the *openness dimension* of systems starts from *CD ROM / file system*, *Internet*, *Web*, and ends with *Open* systems. Our approach can be seen as employing reasoning capabilities for Web-resources, or, concrete, to be on the crossings of description logic in the metadata dimension and Web in the openness dimension.

Adaptive hypermedia has been studied normally in closed worlds, i.e. the underlying document space / the hypermedia system has been known to the authors of the adaptive hypermedia system at design time of the system. As a consequence, changes to this document space can hardly be considered: A change to the document space normally requires the reorganization of the document space (or at least some of the documents in the document space). To open up this setting for dynamic document or information spaces, approaches for so called *open corpus adaptive hypermedia systems* have been discussed [[Brusilovsky, 2001](#), [Henze and Nejd, 2001](#)]. Our approach to bring adaptive hypermedia techniques to the web therefore contribute to the open corpus problem in AH. The relation of adaptive hypermedia and open hypermedia has for example been discussed in [[Bailey et al., 2002](#)].

In our approach, we use several ontologies for describing the features of *domains*, *users*, and *observations*. Compared to the components of adaptive hypermedia systems [[Henze and Nejd, 2003](#)], an ontology for adaptive functionality is missing. However, such an ontology can be derived using the "updated taxonomy of adaptive hypermedia technologies" in [[Brusilovsky, 2001](#)]. Reasoning over these distributed ontologies is enabled by the RDF-querying and transformation language TRIPLE. Related approaches in the area of querying languages for the semantic web can be found, e.g., in [[Bry and Schaffert, 2002](#)]. Here, a rule-based querying and transformation language for XML is proposed. A discussion of the interoperability between Logic programs and ontologies (coded in OWL or DAML+OIL) can be found in [[Grosf et al., 2003](#)].

Reasoning in open worlds like the semantic web is not fully explored yet, sharing and reusing of resources with high quality is still an open problem. In this paper, we discussed first ideas on the application of rules and rule-based querying and transformation language for the domains of open hypermedia and adaptive hypermedia.

Representation of Resources

Semantic web technologies like the Resource Description Format (RDF) [[Lassila and Swick, 2002](#)] or RDF schema (RDFS) [[RDF, 2002](#)] provide us with interesting possibilities. RDF schemas serve to define vocabularies for metadata records in an RDF file. RDF schemas can be used to describe resources, e.g. the RDF bindings of Learning Object Metadata (LOM) [[Nilsson, 2001](#)] can be used for these purposes, or RDF bindings of Dublin

Core [[Dublin Core, 2004](#)]. There is no restriction on the use of different schemas together in one RDF file or RDF model. The schema identification comes with attributes being used from that schema so backward dereferencing is again easily possible.

For example the RDF model of a lecture can use an attribute `subject` from Dublin Core Standard together with `isPartOf` from dublin core metadata terms, etc. Part of an RDF-description for a course on Java programming can be seen in the following example. We have annotated the online version of the Sun Java tutorial [[Campione and Walrath, 2000](#)], which is a freely available online tutorial on Java programming.

```
<?xml version="1.0" encoding="iso-8859-1"?>

<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:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms#">

<rdf:Description
rdf:about="http://java.sun.com/docs/books/tutorial/index.html">
  <rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-
educational#lecture"/>
  <dc:title>The Java Tutorial (SUN)</dc:title>
  <dc:description>A practical guide for programmers with hundreds of
    complete, working examples and dozens of trails - groups of lessons
    on a particular subject.
  </dc:description>
  ...
</rdf:Description>

<rdf:Description rdf:about="Object-Oriented_Programming_Concepts">
  <dc:title>Object-Oriented Programming Concepts</dc:title>
  <dcterms:isPartOf
rdf:resource="http://java.sun.com/docs/books/tutorial/index.html"/>
  <dcterms:hasPart>
    <rdf:Seq>
      <rdf:li rdf:resource="#What_Is_an_Object"/>
      <rdf:li rdf:resource="#What_Is_a_Message" />
      <rdf:li rdf:resource="#What_Is_a_Class"/>
      <rdf:li rdf:resource="#What_Is_Inheritance"/>
      <rdf:li rdf:resource="#What_Is_an_Interface"/>
      <rdf:li
rdf:resource="#How_Do_These_Concepts_Translate_into_Code"/>
      <rdf:li rdf:resource="#Questions_and_Exercises_Object-
Oriented_Concepts"/>
    </rdf:Seq>
  </dcterms:hasPart>
</rdf:Description>

....

<rdf:Description rdf:about="What_Is_an_Object">
  <dc:title>What Is an Object?</dc:title>
  <dc:description>An object is a software bundle of related variables
    and methods. Software objects are often used to model real-world
    objects you find in everyday life. </dc:description>
  <dc:language rdf:resource=
    "http://www.kbs.uni-hannover.de/~henze/lang.rdf#en"/>
  <dc:subject rdf:resource=
```

```

        "http://www.kbs.uni-hannover.de/~henze/java.rdf#OO_Objects"/>
        <dcterms:isPartOf rdf:resource="#Object-Oriented_Programming_Concepts"/>
    </rdf:Description>

    ...

</rdf:RDF>

```

While RDF schema provides a simple ontology language, more powerful ontology languages which reside on top of RDF and RDF schema are available, too. For example, ontology languages like DAML+OIL [[DAML+OIL, 2001](#)] (the joint initiative of DAML (Darpa Agent Markup Language) and OIL (Ontology Inference Layer)) provide ontology layers on top of RDF / XML. Recently, OWL [[OWL, 2003](#)] (Web Ontology Language) has been developed, further enriching RDF.

An open question is how we can combine reasoning mechanisms on these (distributed) metadata and data resources, in order to generate hypertext presentations, link structures, etc., to bring the interoperability ideas from OHS to the WWW. This section will first describe semantic web tools that we employ in our approach, and then describe some structures for metadata components which allow us to generate link structures according to user features.

Bringing together Resources and Reasoning

On top of the RDF and ontology-layer, we find the layer of logic in the semantic web tower, or, more recently, the layers of rules and logic framework [[Berners-Lee, 2002](#)]. In our approach, the communication between reasoning rules and the open information environment will take place by exchanging RDF annotations: the rules reason over distributed RDF-annotations, results will be given back as RDF-files, too.

A rule language especially designed for querying and transforming RDF models is TRIPLE [[Sintek and Decker, 2002](#)]. Rules defined in TRIPLE can reason about RDF-annotated information resources (required translation tools from RDF to triple and vice versa are provided).

TRIPLE supports *namespaces* by declaring them in clause-like constructs of the form *namespaceabbrev := namespace*, resources can use these namespaces abbreviations.

```
sun_java := "http://java.sun.com/docs/books/tutorial".
```

Statements are similar to F-Logic object syntax: An RDF statement (which is a triple) is written as `subject[predicate → object]`. Several statements with the same subject can be abbreviated in the following way:

```
sun_java:'index.html'[rdf:type->doc:Document;
doc:hasDocumentType->doc:StudyMaterial].
```

RDF *models* are explicitly available in TRIPLE: Statements that are true in a specific model are written as "@model", e.g.

```
doc:00_Class[rdf:type->doc:Concept]@results:simple.
```

Connectives and quantifiers for building logical formulae from statements are allowed as usual, i.e. \wedge , \vee , \neg , \forall , \exists , etc. For TRIPLE programs in plain ASCII syntax, the symbols AND, OR, NOT, FORALL, EXISTS, <-, ->, etc. are used. All variables must be introduced via quantifiers, therefore marking them is not necessary.

Domain Ontologies

First of all we need to determine a domain ontologies. Domain ontologies comprise usually classes (classifies objects from a domain) and relationships between them. One possible domain in hypermedia application can be a domain of documents and concepts described in an application domain.

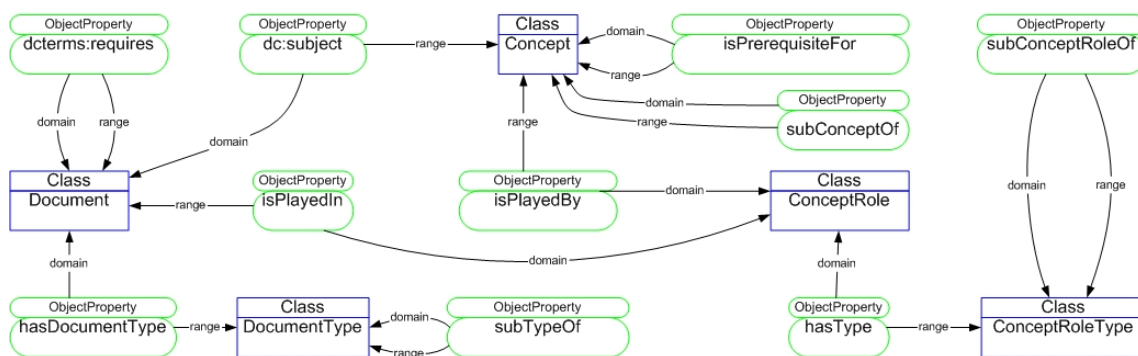


Figure 1: Ontology of documents

A simple ontology for documents and their relationships to other components is depicted in fig. 1. The class `Document` is used to annotate a resource which is a document. Documents describe some concepts. We use class `Concept` to annotate concepts. Concepts and documents are related through `dc:subject` property. Documents can be ordered by `dcterms:requires` relationship. Concepts and documents have a certain role in their collaboration in certain document. We represent these facts by instances of `DocumentRole` class and its two properties: `isPlayedIn` and `isPlayedBy`. Concepts, document roles and concept roles can form hierarchies. We define `subRoleOf`, `subConceptRoleOf`, and `subConceptOf` properties for these purposes. Concepts play a certain role in a document. We recognize `Introduction` and `FullDescription` concept roles.

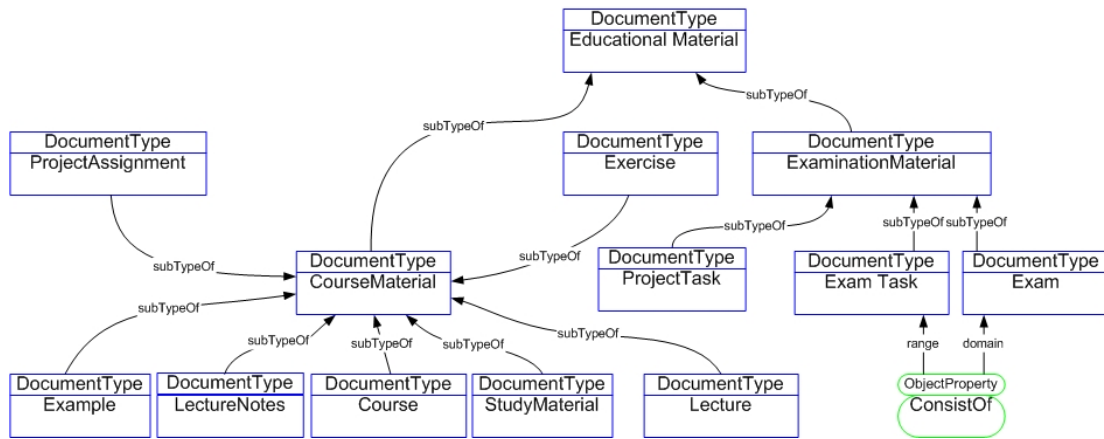


Figure 2: Ontology for documents types

Document can have a type. Figure 2 depicts the ontology with several document types for educational domain. The most general document type is Educational Material. Educational Material has two subtypes: Course Material and Examination Material. Examination Material can be further specialized to Project Task, Exam Task, and Exam. The Exam can consist of the Exam Task-S.

Course Material can be further specialized into Lecture, Example, LectureNote, Course, Exercise, and Project Assignment.

The document roles represent intended usage of the document in general. When a document is authored it is already known whether it will be a Lecture, Example and so on and it hardly fits to another role. Besides document roles, we recognize document types as well. Document types represent different context of a document. It means that we can differentiate at least between examination and study material. These are represented as separate document types StudyMaterial and ExaminationMaterial.

Figure 3 depicts Programming_Strategies concept with its subconcepts: Object_Oriented, Imperative, Logical, and Functional. OO_Class, OO_Method, OO_Object, OO_Inheritance, and OO_Interface are depicted as subconcepts of Object_Oriented.

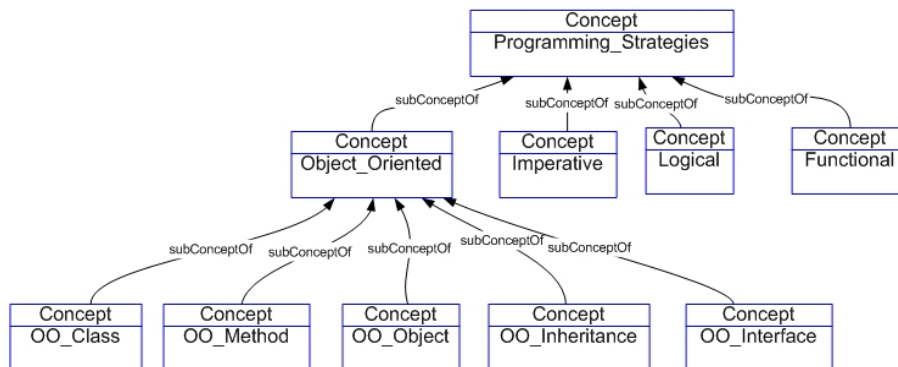


Figure 3: Concept ontology for Java e-lecture

Above described ontologies are used then in annotations of concrete documents/resources. An example of such resource can be a page describing `sun_java:'java/concepts/class.html'`. Following example shows how such a page can be annotated based on ontologies.

```
sun_java:'java/concepts/class.html' [
rdf:type->doc:Document;
dc:subject->doc:OO_Class].

doc:OO_Class [
rdf:type->doc:Concept;
doc:isPrerequisiteFor->doc:OO_Inheritance;
doc:subConceptOf->doc:Classes_and_objects].

doc:ClassesIntroduction [
rdf:type->doc:ConceptRole;
doc:isPlayedBy->doc:OO_Class;
doc:isPlayedIn->sun_java:'java/concepts/class.html';
doc:hasType->doc:Introduction].

doc:Introduction [
rdf:Type->doc:ConceptRoleType;
doc:subConceptRoleOf->doc:Cover].
```

The page is a document (RDF type `Document`). It describes information about classes. Thus it is annotated with `OO_Class` concept covered in the page. The `OO_Class` concept is annotated with type `Concept` and is subconcept of the `Classes_and_objects` concept. The `OO_Class` concept is prerequisite for the `OO_Inheritance`. A page can have prerequisites. Then the `dcterms:requires` property can be used in the annotation.

The `OO_Class` concept plays a role of introduction in the `sun_java:'java/concepts/class.html'` document. This is annotated by `ClassesIntroduction` resource, which is of type `ConceptRole`. The reference to `OO_Class` concept and the document where it plays the introduction role is annotated by using properties `isPlayedBy` and `isPlayedIn` respectively. The role has type `Introduction`. The `Introduction` is of type `ConceptRoleType` and is subtype of `Cover` concept role type.

Users

Data about a user serves for deriving contextual structures. It is used to determine how to adapt the presentation of hypertext structures. Here we define an ontology for a user profile based on IEEE Personal and Private Information (PAPI) [IEEE, 2000]. PAPI distinguishes *personal*, *relations*, *security*, *preference*, *performance*, and *portfolio* information. The *personal* category contains information about names, contacts and addresses of a user. *Relations* category serves as a category for specifying relationships between users (e.g. *classmate*, *teacherIs*, *teacherOf*, *instructorIs*, *instructorOf*, *belongsTo*, *belongsWith*). *Security* aims to provide slots for credentials and access rights. *Preference* indicates the types of devices and objects, which the user is able to recognize. *Performance* is for storing information about measured performance of a user through learning material (i.e. what does a user know). *Portfolio* is for accessing previous experience of a user. Each category can be extended. For more discussion on learner modeling standards see for example [Dolog and Nejd, 2003].

Figure 4 depicts an example of an ontology for a learner profile. The ontology is based on *performance* category of PAPI. We are storing sentences about a learner which has a Performance. The Performance is based on learning experience (learningExperienceIdentifier), which is taken from particular document. The experience implies a Concept learned from the experience, which is maintained by learningCompetency property. The Performance is certified by a Certificate, which is issued by a certain Institution. The Performance has a certain PerformanceValue, which is in this context defined as a float number and restricted to interval from 0 to 1.

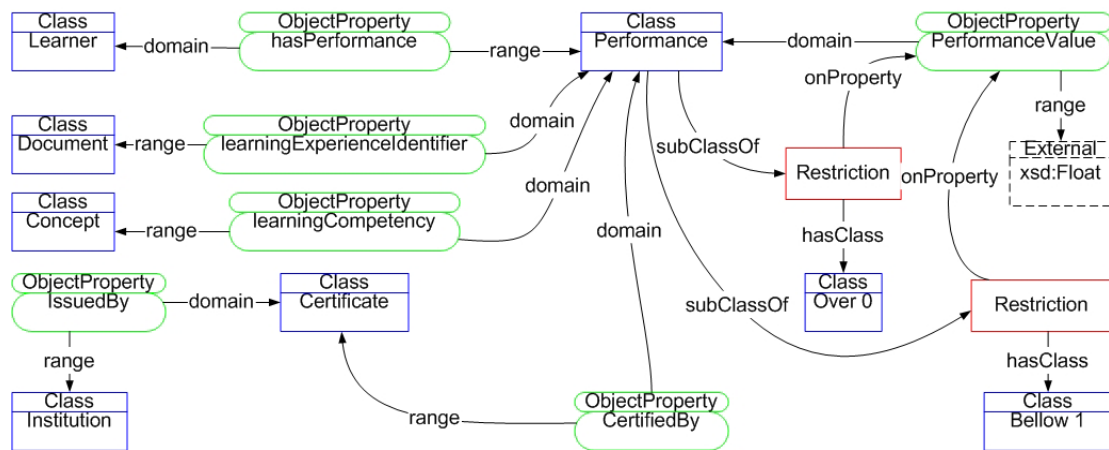


Figure 4: Ontology for learner performance

Another possibility to restrict the PerformanceValue is to define it with a range of LevelOf Knowledge. Then the instances of the class can be taken as measures of the learner performance.

The example of simple learner profile can look as follows.

```

user:user2 [
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user2P].

user:user2P [
  rdf:type->learner:Performance;
  learner:learningExperienceIdentifier-
>sun_java:'java/concepts/object.html';
  learner:learningCompetency->doc:OO_Object;
  learner:CertifiedBy->KBScerturi:C1X5TZ3;
  learner:PerformanceValue->0.9
].

KBScerturi:C1X5TZ3 [
  rdf:type->learner:Certificate;
  learner:IssuedBy->KBSuri:KBS
].

KBSuri:KBS [
  rdf:type->learner:Institution
].

```

The learner `user2` has the performance (`user2P`) record. The performance contains a learning experience about the KBS Java objects resource. The concept covered in the resource is stored in the performance as well. Then a certificate about the performance with performance value and institution who issued the certificate is recorded into the learner performance as well.

Observations

During runtime, users interact with a hypertext system. The user's interactions can be used to draw conclusions about possible user interests, about user's goal, user's task, user's knowledge, etc. These concluded user features can, as described in the previous section, be used for providing personalized views on hypertexts. An ontology of observations should therefore provide a structure of information about possible user observations, and - if applicable - their relations and/or dependencies.

A simple ontology for observations is depicted in fig. 5. The ontology allow us to instantiate facts that a `Learner` has interacted with (`hasInteraction` property) with a particular `Document` (`isAbout` property) via an interaction of a specific type (`InteractionType`). The interaction has taken place in a time interval between `beginTime` and `endTime`, and has a certain level (`Level`) associated, the `ObservationLevel`. Several events (see next section) can contribute to an interaction. Example of `InteractionTypes` are of kind `access`, `bookmark`, `annotate`, examples for `ObservationLevels` are that a user has visited a page, has worked on a project, has solved some exercise, etc.

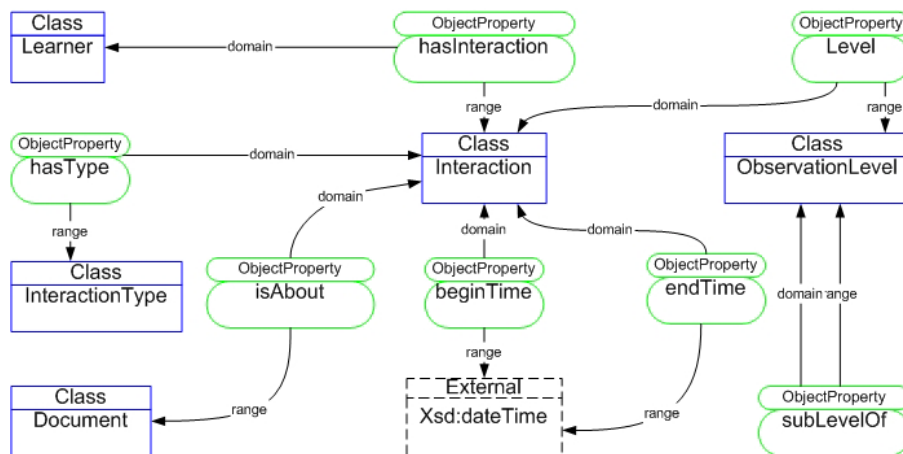


Figure 5: Ontology for observations

Generating Hypertext Structures

Hypertext structures as described in several works on open hypermedia (see e.g. [Millard et al., 2000]) can be generated from metadata reported in the previous section. We do not store the hypertext structures on servers as first class entities but we allow to generate such structures on the fly. In order to generate such hypertext structures we need an ontology for structures. Then transformation rules can be used to generate instances of that structure.

Presentation Ontology

A presentation ontology is used for describing structure relevant for visualization. Such an ontology adapted from FOHM [Millard et al., 2000] is depicted in fig. 6.

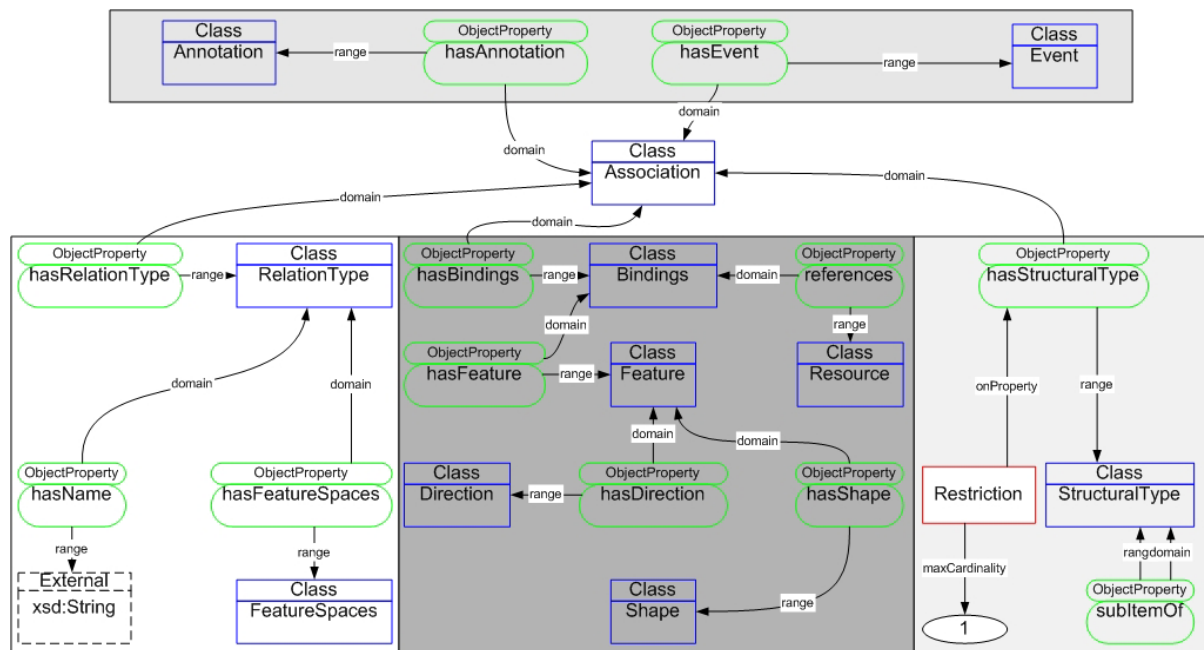


Figure 6: A part of presentation ontology

The main element of the ontology is the Association. Like in [Millard et al., 2000], the Association is built from three components: Bindings, RelationType, and StructuralType (in FOHM they refer to it as Cartesian product of bindings, relation type and structural type). These three components (classes) are related to association through hasBindings, hasRelationType, and hasStructuralType properties.

Bindings references a particular Resource on the web (document, another association, etc.), and Feature-s. A Feature can be a Direction, Shape, etc. Entries for Direction are depicted in figure 7b, entries for Shape are depicted in the figure 7c.

The RelationType has a Name which is a string. The RelationType also points to the FeatureSpaces. Entries for the FeatureSpaces are depicted in figure 7a. A StructuralType is one of stack, link, bag, or sequence of resources.

In addition, Association can have associated events (e.g. click events for processing user interactions) through hasEvent property, and an annotation (e.g. green/red/yellow icon from traffic light metaphor technique from adaptive hypermedia) through hasAnnotation property.

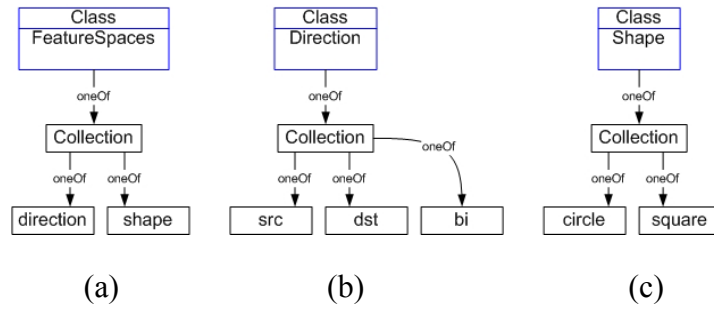


Figure 7: Members of Collection of: (a) Feature Spaces, (b) Direction, (c) Shape.

The `hasEvent` property defines an event which is provided within the document (to be able to get appropriate observation). Whenever the event is generated observation reasoning rules assigned to this type of event are triggered. The `represents` property references a resource, which is stored in observations about learner, after an event is generated as well.

FOHM introduces *context* and *behavior* objects. Filtering and contextual restrictions maintained by the *context* objects in FOHM is substituted by more richer reasoning language and rules in our approach. On the other hand, interactions and observations together with events substitute the notion of *behavior* objects.

Reasoning Rules

In this chapter we show how rules are employed to reason over distributed information sources (ontologies, user profile information, resource descriptions). The communication between reasoning rules and the open information environment will take place by exchanging RDF annotations [RDF, 2002]. Rules are encoded in the TRIPLE rule language (see section 3.1). For further examples on adaptation rules we refer the reader to [Dolog et al., 2003].

In the following, we provide a set of rules that can be used to construct an *example*-relation between resources. Assume a user U is visiting some page D . An example, illustrating the content of this page, can be found by comparing the concepts explained on the current page with the concepts shown on an example page. Several grades of how good an example is can be derived.

The easiest way for deriving an example-relation to a page D is by ensuring that each concept on D is covered by the example E :

```

FORALL D, E example(D,E) <-
  studyMaterial(D) AND example(E) AND
  EXISTS C1 (D[dc:subject->C1]) AND
  FORALL C2 (D[dc:subject->C2] -> E[dc:subject->C2]).

```

The second line in the rule above ensures that D is `StudyMaterial` and E is an `Example` (according to the ontology of documents "docs"). The third rule is verifying that D really is about some measurable concept - thus there exists a metadata annotation like `dc:subject`.

The fourth line then really expresses what our rule should check: Whether each concept on D will be explained in the example E .

Another possibility is to provide relations to examples that cover exactly the same concepts as a page D :

```
FORALL D, E exact_example(D,E) <-
  studyMaterial(D) AND example(E) AND
  EXISTS C1 (D[dc:subject->C1]) AND
  FORALL C1 (D[dc:subject->C1] -> E[dc:subject->C1]) AND
  FORALL C2 (E[dc:subject->C2] -> D[dc:subject->C2]).
```

The second and third line in this rule are the same as in the previous rule. The fourth and fifth line ensure that each concept on D is covered on E and vice versa.

If we want to show examples which might illustrate only some aspects of a page D , we can derive relations to *weaker* examples by

```
FORALL D, E weaker_example(D,E) <-
  studyMaterial(D) AND example(E) AND
  EXISTS C (D[dc:subject->C] AND E[dc:subject->C]).
```

which is be valid whenever at least on concept explained on D is part of the example E .

From the area of adaptive hypermedia, several methods and techniques have been provided to adapt the navigation and / or the content of a hyperspace to the needs, preferences, goals, etc. of each individual user. In [[Henze and Nejd, 2003](#)] we have provided a logical characterization of adaptive educational hypermedia based on First Order Logic (FOL). There, an adaptive educational hypermedia system is described in FOL as a quadruple consisting of a *document space* - a hypermedia system which document nodes and their relations, a *user model* for modeling and inferencing on various individual characteristics of a user, an *observation component* which is responsible for monitoring a user's interaction with the system, and an *adaptation component* which consists of rules which describe adaptive functionality. A way to implement open adaptive hypermedia system is shown in [[Dolog et al., 2003](#)]. In this paper, we will use adaptive hypermedia to provide personalized associations. We can think of a personalized *pedagogical* recommendation of examples: The best example is an example that shows the new things to learn in context of already known / learned concepts: This would embed the concepts to learn in the previous learning experience of a user. The rule for derive this *best_example* is as follows:

```
FORALL D, E, U best_example(D,E,U) <-
  studyMaterial(D) AND example(E) AND user(U) AND example(D,E) AND
  FORALL C ( (E[dc:subject->C] AND NOT D[dc:subject->C]) ->
    p_obs(C, U, Learned) ).
```

The rule for determining whether a user has learned some concept C ($p_obs(C, U, Learned)$) is derived by checking the characteristics of the user profile. A concept is assumed to be learned if we find a *Performance* of this user via the user profile, which is related to the concept in question.

```
FORALL C, U p_obs(C, U, Learned) <- user(U) AND concept(C) AND
  EXISTS P (U[learner:hasPerformance->P] AND user_performance(P) AND
  P[learner:learningCompetency->C]).
```

The results of these rules (on the RDF-annotated and to triple translated resources provided in the Appendix) is e.g. that a page on "objects in Java (object.html)" can be related to pages which show "concepts of object orientation in Java (practical.html)" or "objects and methods in Java (objects_methods.html)". These relations are derived by using the general "example"-rule:

```
D = sun_java:'java/concepts/object.html', E =
sun_java:'java/concepts/practical.html'
D = sun_java:'java/concepts/object.html', E =
kbs_java:'java_script/examples/objects_methods.html'
```

The "exact_example-rule" from above derives for this data set that only the "overview on object-orientation in Java (OO_overview.html)" has an exact matching example.

```
D = kbs_java:'java_script/concepts/OO_overview.html',
E = sun_java:'java/concepts/practical.html'
```

The "weaker_example-rule" suggest the same example page (practical.html) which exactly fits to the document OO_overview.html also to pages about only some aspects like "methods in Java (message.html).

```
D = sun_java:'java/concepts/message.html',
E = sun_java:'java/concepts/practical.html'
```

The "best_example" for a user who is currently visiting a page on "methods in Java (message.html)" and who has already knowledge about "objects in java" is an example illustrating these two concepts (object_methods.html). In the data set provided in the appendix, user2 is currently in this position.

```
D = sun_java:'java/concepts/message.html',
E = kbs_java:'java_script/examples/objects_methods.html',
U = user:user2
```

Further rules for generating personalized hypertext associations can be used by more extensive use of facts from domain, user, and observation ontology. E.g. the mentioned `subConceptOf` relationship in the concept-ontology of the java application domain can be for example utilized to recommend either more general documents introducing a concept of programming strategies in general, or to recommend more specific documents (resources) about object oriented programming strategy based on requirements, level of knowledge, or interest of a user.

Sequencing relationship is another relationship which can be used to recommend documents. A document (resource) which describes a concept (the concept appears in `dc:subject` slot in metadata about the document) from the beginning of the sequence will be recommended sooner than a document which describes a concept from the end of such a sequence.

A dependency relationship referring to whether a concept depends on another concept can be used as well. It can be used to recommend documents which describe dependent concepts together with a document describing a concept which was recommended by another rule.

Conclusion and Further Work

In this paper, we have proposed an approach for dynamically generating personalized hypertext relations powered by reasoning mechanisms over distributed RDF annotations. We have shown an example set of reasoning rules that decide for personalized relations to example pages given some page. Several ontologies have been used which correspond to the components of an adaptive hypermedia system: a domain ontology (describing the document space, the relations of documents, and concepts covered in the domain of this document space), a user ontology (describing learner characteristics), and an observation ontology (modeling different possible interactions of a user with the hypertext). For generating hypertext structures, a presentation ontology has been introduced. We have been developing a demonstrator system showing the realization of the formalism we presented in this paper. This demonstrator, the *Personal Reader* [Dolog et al., 2004a], generates a personalized conceptual context of learning resources. This context is generated by using adaptation rules like those presented in this paper, and integrates this technology with a personalized search facility [Dolog et al., 2004b].

In further work, we plan to extend our demonstrator, and to investigate how to employ further ontologies like an ontology for educational models. This will enable us to add additional rules to enhance adaptive functionality based on the facts modeled in the knowledge-base by utilizing additional relationships.

Bibliography

[Bailey et al., 2002] Bailey, C., Hall, W., Millard, D., and Weal, M. (2002). Towards open adaptive hypermedia. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain.

[Bechhofer et al., 2001] Bechhofer, S., Carr, L., Goble, C., and Hall, W. (2001). Conceptual open hypermedia = the semantic web? In *Second International Workshop on the Semantic Web*, Hong Kong, China.

[Berners-Lee, 2002] Berners-Lee, T. (2002). The semantic web - mit/lcs seminar. <http://www.w3c.org/2002/Talks/09-lcs-sweb-tbl/>.

[Brusilovsky, 2001] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87-100.

[Brusilovsky and Maybury, 2002] rusilovsky, P. and Maybury, M. (2002). *The Adaptive Web*. Communications of the ACM.

[Bry and Schaffert, 2002] Bry, F. and Schaffert, S. (2002). A gentle introduction into xcerpt, a rule-based query and transformation language for xml. In *International Workshop on Rule Markup Languages for Buisness Rules on the Semantic Web*, Sardinia, Italy.

[Campiono and Walrath, 2000] Campione, M. and Walrath, K. (2000). *The Java(TM) Tutorial: A Short Course on the Basics (3rd Edition)*. Addison-Wesley.

[Carr et al., 2001] Carr, L., Bechhofer, S., Goble, C., and Hall, W. (2001). Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the Tenth International World Wide Web Conference*, Hongkong.

[DAML+OIL, 2001] DAML+OIL (2001). DAML+OIL. <http://www.daml.org/2001/03/daml+oil-index.html>.

[Dolog et al., 2003] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2003). Towards an adaptive semantic web. In *Principles and Practive of Semantic Web Reasoning (PPSWR'03)*, Mumbai, India.

[Dolog et al., 2004a] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004a). The personal reader: Personalizing and enriching learning resources using semantic web technologies. Technical report, Univ. of Hannover. submitted for publication.

[Dolog et al., 2004b] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004b). Personalization in distributed e-learning environments. In *International World Wide Web Conference*, New York, USA.

[Dolog and Nejdl, 2003] Dolog, P. and Nejdl, W. (2003). Challenges and benefits of the semantic web for user modelling. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary.

[Dublin Core, 2004] Dublin Core (2004). Dublin Core. <http://dublincore.org/>.

[Gronbaek et al., 2000] Gronbaek, K., Sloth, L., and Bouvin, N. O. (2000). Open hypermedia as user controlled meta data for the web. In *Ninth International World Wide Web Conference*, pages 554-566, Amsterdam, The Netherlands.

[Grosz et al., 2003] Grosz, B. N., Horrocks, I., Volz, R., and Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In *Twelfth International World Wide Web Conference*, Budapest, Hungary.

[Henze and Nejdl, 2001] Henze, N. and Nejdl, W. (2001). Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems*, 12.

[Henze and Nejd, 2003] Henze, N. and Nejd, W. (2003). Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*, Budapest, Hungary.

[IEEE, 2000] IEEE (2000). IEEE P1484.2/D7, 2000-11-28. draft standard for learning technology. public and private information (papi) for learners (papi learner). Available at: <http://ltsc.ieee.org/wg2/>. Accessed on October 25, 2002.

[IMS, 2002] IMS (2002). IMS: Standard for Learning Objects. <http://www.imsglobal.org/>.

[Kampa et al., 2001] Kampa, S., Miles-Board, T., Carr, L., and Hall, W. (2001). Linking with meaning: Ontological hypertext for scholars. Technical report, University of Southampton. citeseer.nj.nec.com/kampa01linking.html.

[Lassila and Swick, 2002] Lassila, O. and Swick, R. (2002). W3c resource description framework (rdf) model and syntax specification. Available at: <http://www.w3.org/TR/REC-rdfsyntax/>. Accessed on October 25, 2002.

[LOM, 2002] LOM (2002). LOM: Draft Standard for Learning Object Metadata. <http://ltsc.ieee.org/wg12/index.html>.

[Millard et al., 2000] Millard, D. E., Moreau, L., Davis, H. C., and Reich, S. (2000). FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In *11th ACM Conference on Hypertext and Hypermedia*, pages 93-102, San Antonio, Texas, USA.

[Nilsson, 2001] Nilsson, M. (2001). Ims metadata rdf binding guide. <http://kmr.nada.kth.se/el/ims/metadata.html>.

[OWL, 2003] OWL (2003). OWL. <http://www.w3.org/2001/sw/WebOnt/>.

[RDF, 2002] RDF (2002). Resource Description Framework (RDF) Schema Specification 1.0. <http://www.w3.org/TR/rdf-schema>.

[Sintek and Decker, 2002] Sintek, M. and Decker, S. (2002). Triple - an rdf query, inference, and transformation language. In Horrocks, I. and Hendler, J., editors, *International Semantic Web Conference (ISWC)*, pages 364-378, Sardinia, Italy. LNCS 2342.

[Weal et al., 2001] Weal, M. J., Hughes, G. V., Millard, D. E., and Moreau, L. (2001). Open hypermedia as a navigational interface to ontological information spaces. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 227-236. ACM Press.

[XML, 2003] XML (2003). XML: extensible Markup Language. <http://www.w3.org/XML/>.

Appendix: Set of Rules for Deriving Relations between Information Pages and Examples

```
daml    := "http://www.daml.org/.../daml+oil#".
rdf     := "http://www.w3.org/1999/02/22-rdf-syntax-ns#".
doc     := "http://www.example.org/doc#".
```

```

results := "http://www.results.org/results#".
sun_java := "http://java.sun.com/docs/books/tutorial/".
kbs_java := "http://www.kbs.uni-hannover.de/".
java := "http://www.kbs.uni-hannover.de/~henze/java.rdf#".

@results:data{
sun_java:'index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/concepts/index.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial].
sun_java:'java/concepts/object.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Object'].
sun_java:'java/concepts/message.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Method'].
sun_java:'java/concepts/class.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Class'].
sun_java:'java/concepts/inheritance.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Inheritance'].
sun_java:'java/concepts/interface.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Interface'].
sun_java:'java/concepts/practical.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:Example;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method';
  dc:subject->java:'OO_Class';
  dc:subject->java:'OO_Inheritance';
  dc:subject->java:'OO_Interface'].

kbs_java:'java_script/examples/objects_methods.html'[rdf:type-
>doc:Document;
  doc:hasDocumentType->doc:Example;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method'].
kbs_java:'java_script/concepts/OO_overview.html'[rdf:type->doc:Document;
  doc:hasDocumentType->doc:StudyMaterial;
  dc:subject->java:'OO_Object';
  dc:subject->java:'OO_Method';
  dc:subject->java:'OO_Class';
  dc:subject->java:'OO_Inheritance';
  dc:subject->java:'OO_Interface'].

java:'OO_Object'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Method'].

java:'OO_Method'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Class'].

java:'OO_Class'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Inheritance'].

java:'OO_Inheritance'[rdf:type->doc:Concept;
  doc:isPrerequisiteFor->java:'OO_Interface'].

user:user1[
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user1P].

```

```

user:user1P[
  rdf:type->learner:Performance].

user:user2[
  rdf:type -> learner:Learner;
  learner:hasPerformance -> user:user2P].

user:user2P[
  rdf:type->learner:Performance;
  learner:learningCompetency -> java:'OO_Object'].
}

@results:simple{

  FORALL O,P,V O[P->V] <-
    O[P->V]@results:data.

  FORALL D document(D) <- D[rdf:type->doc:Document].
  FORALL C concept(C) <- C[rdf:type->doc:Concept].
  FORALL U user(U) <- U[rdf:type->learner:Learner].
  FORALL P user_performance(P) <- P[rdf:type->learner:Performance].
  FORALL E example(E) <- document(E) AND
    E[doc:hasDocumentType->doc:Example].
  FORALL E studyMaterial(E) <- document(E) AND
    E[doc:hasDocumentType->doc:StudyMaterial].

  FORALL C, U p_obs(C, U, Learned) <- user(U) AND concept(C) AND
    EXISTS P (U[learner:hasPerformance->P] AND user_performance(P) AND
    P[learner:learningCompetency->C]).

  FORALL D, E example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C2 (D[dc:subject->C2] -> E[dc:subject->C2]).

  FORALL D, E exact_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C1 (D[dc:subject->C1]) AND
    FORALL C1 (D[dc:subject->C1] -> E[dc:subject->C1]) AND
    FORALL C2 (E[dc:subject->C2] -> D[dc:subject->C2]).

  FORALL D, E weaker_example(D,E) <-
    studyMaterial(D) AND example(E) AND
    EXISTS C (D[dc:subject->C] AND E[dc:subject->C]).

  FORALL D, E, U best_example(D,E,U) <-
    studyMaterial(D) AND example(E) AND user(U) AND example(D,E) AND
    FORALL C ( (E[dc:subject->C] AND NOT D[dc:subject->C]) ->
    p_obs(C, U, Learned) ).
}

/* Several Views */
FORALL D, E <- example(D, E)@results:simple.
FORALL D, E <- exact_example(D, E)@results:simple.
FORALL D, E <- weaker_example(D, E)@results:simple.
FORALL D, E, U <- best_example(D, E, U)@results:simple.

```