# Semantic Resource Management for the Web: An E-Learning Application

Julien Tane Christoph Schmitz Gerd Stumme

Learning Lab Lower Saxony, Expo Plaza 1, D-30539 Hannover, Germany
<lastname>@learninglab.de

Institute for Applied Informatics and Formal Description Methods (AIFB)
University of Karlsruhe, D-76128 Karlsruhe, Germany

## ABSTRACT

Topics in education are changing with an ever faster pace. E-Learning resources tend to be more and more decentralized. Users increasingly need to be able to use the resources of the web. For this, they should have tools for finding and organizing information in a decentralized way. In this paper, we show how an ontology-based tool suite allows to make the most of the resources available on the web.

## Categories and Subject Descriptors

K.3.1 [**Computing Milieux**]: Computer Uses in Education — Computer Uses in Education; I.7.5 [**Computing Methodologies**]: Document and Text Processing — Document Capture; H.5 [**Information Systems**]: Information Interfaces and Presentation

## 1. INTRODUCTION

In recent years, mobile technology and internet access have improved in such a way that it has become reality to store resources remotely. In the E-Learning domain, the use of notebooks and mobile devices implies a new way of managing resources. The goal is, therefore, to exploit these chances as far as possible. In [20], the increasing role of mobile devices in education is predicted. The authors argue that E-Learning has to be seen as a part of the general framework of knowledge management. To achieve this, it is important to integrate the technologies of these two domains.

Among the current knowledge management techniques, ontologies play a greater role than ever. Current research on ontologies has shown that they facilitate the retrieval, interaction and management of resources; for examples see [13] or [29]. In the E-Learning domain, standards have been developed to help describe learning objects.[1] Although these developments are a good start, there is a need for a more comprehensive approach which integrates the content, structure and evolution of the learning material. We present here our methodology and implementation of an ontology-based Courseware Watchdog, which supports the user in finding and organizing distributed courseware resources by offering a common framework for the retrieval and organization of courseware material. We illustrate this with a usage scenario.

[1]For more information on these standards, see `http://ltsc.ieee.org/index.htm`.


*WWW2004*, May 17–20, 2004, New York, New York, USA.

In the first section we briefly expose the role of ontologies for E-Learning. The following section discusses a prototypical usage scenario, from which we then derive the requirements for an ontology-based courseware management system. This will lead us to an overall presentation of the integrated architecture of our Courseware Watchdog. In the four subsequent sections, we describe the specific modules of the Courseware Watchdog in more detail. Finally, we will sum up our results and discuss further research issues.

The work described in this article builds up on a vision which we presented and analyzed in [26]. Some of the parts which contribute to the Courseware Watchdog have been presented before. The contribution of this paper is twofold: First it discusses the benefit of a tight interaction between the several parts. In particular, it shows up which synergy effects can be gained by combining approaches from different domains. Second, it also shows that the integration is not straightforward, but poses difficulties not only on the technical level, but also on the conceptual level. The paper discusses how these difficulties can be mitigated.

## 2. SEMANTIC WEB AND E-LEARNING

On a personal computer, it is possible to organize resources according to personal needs. In the case of remote resources, this is not possible anymore, since their storage is not under the control of the user. Through the use of hypertext, remote material can be linked and retrieved when needed. But the particular problem of finding and organizing this remote material becomes even more crucial.

In [3], it is shown that standards like LOM or Dublin Core are gaining importance. They provide increased information on the learning material that is to be found in the web. However, their simple structure prohibits their use for modeling more complex knowledge. [28] explains how Semantic Web technologies based on ontologies can improve different aspects of the management of E-Learning resources. Indeed, ontologies are a means of specifying the concepts and their relationships in a particular domain of interest. Web Ontology languages, like OWL, are specially designed to facilitate the sharing of knowledge between actors [27] in a distributed environment. We wish to emphasize here on various advantages.

From the modeling point of view, ontology languages are not only able to integrate LOM[2] and Dublin Core metadata, but also

[2]See `http://www.imsglobal.org/metadata/` on RDF(S) LOM Binding.

allow for the extension of the description of the learning objects with non standard metadata, thus giving users and groups of users more flexibility when sharing resources.

Research in the E-Learning domain shows that standards are needed for interoperability[3], but true interoperability does not only need data integration, it also has to consider the integration of applications. In this paper, we illustrate such an integration of E-learning related applications with the implementation of a Courseware Watchdog. In the next section, we present a prototypical use case where such an integration is needed, from which we derive a set of requirements.

# 3. SCENARIO AND USER REQUIREMENTS

To illustrate the ultimate goal and purpose of our tool, we present a prototypical scenario in this section. It will show the different tasks that need to be addressed when trying to find and organize courseware material.

While we use a teacher as an example for our scenario, similar points could be made for the case of a learner who wants to manage and share her portfolio of learning resources.

## 3.1 Usage Scenario

Professor Meyer is a university professor at a German university in the domain of computer science. His main fields of activity are Data Mining and Knowledge Management. Since these fields of studies evolve very rapidly he has to be aware of the latest developments in these domains. At the beginning of the semester break, Professor Meyer prepares two lectures and two seminars which he will give during the next semester: the lecture "Introduction to Computer Science" designed for freshmen, a "Knowledge Discovery" lecture for more advanced students, a seminar on "Knowledge Management", and a seminar on peer-to-peer and web services. He already has material from previous lectures but he feels that there is still room for improvement.

Professor Meyer expects to find a lot of material accessible on the web for the first lecture and he already has some lecture notes which are more or less ready to be used. He has already annotated part of the material with educational markup using the LOM and Dublin Core standards, and for the domain of computer science he uses the well known ACM taxonomy[4].

### *3.1.1 Browsing Existing Content*

Professor Meyer planned the construction of the two courses in the following way: first, he needs to have a systematic overview over the material that he has collected by and by. Instead of using a generic ontology like the ACM classification, he may also use an ontology that was created by one of his colleagues. In that case, Professor Meyer needs to familiarize himself first with the content and the meaning of the concepts and relations, as the ontology may be created from a slightly different viewpoint.

Beside the most important concepts, the ontology also contains pointers to relevant resources that he collected (e. g., HTML pages, PDF files or powerpoint files). Some of the resources may be stored on the professor's computer, while others may be located at a remote location. The professor can access these resources by means of a standard browser.

### *3.1.2 Finding Relevant Material*

Prof. Meyer now wants to find new material. For this, he considers two approaches: either search for it in the world wide web or in distinct decentralized repositories that provide more structured semantic metadata about learning material.

Both tasks can again be supported by using an ontology. It provides means for extending search term combinations with semantically related concepts, and it serves as an interface to the (semi-)structured repositories.

*Querying a Network of Semantically Annotated Material.* Suppose our professor has access to an Edutella[5] network [21]. Edutella is a peer-to-peer (P2P) framework, in which the different peers provide semantically annotated metadata on learning material. It also allows for the integration of web services to gain access to material offered by libraries or similar repositories, see for example [2].

In order to find new relevant material in the P2P network, Professor Meyer first needs to define a query. Professor Meyer searches for lectures on the topics "Algorithmics" or "Knowledge Discovery". Hence he defines the following query:

> Return every 'Lecture' which 'hasTopic' 'Algorithmics' or which 'hasTopic' 'Knowledge Discovery' and for each match retrieve also the values of the properties 'dc:title' and 'dc:author'.

He sends the query to the network and receives an answer. He can then access the retrieved resources by standard browsers and viewers. He can send more specific queries to the Edutella network to get further information about the specific lectures or authors that are of interest to him.

*Finding Learning Material on the Web.* Professor Meyer knows some web sites that are relevant for his task. He is quite certain that some interesting material (or at least pointers to it) would be accessible there, had he only time to browse the sites and follow the hyperlinks.

An obvious solution would be to apply a crawler that follows the links starting from these pages, and to collect the resources showing up. However, if every individual user starts his own crawler, this would lead to an unnecessary overload of web traffic. On the other hand, Prof. Meyer is not interested in harvesting all pages accessible from the start URLs, but only in a specific subset. He selects a set of concepts from the ontology, which specifies the kind of pages he wants to retrieve. The crawler then scores each page and each hyperlink according to the frequency of these concepts on the whole page and around the hyperlink. Concepts that Meyer did not type in explicitly, but which are semantically related to these concepts within the ontology, also add to the score. The links with the highest score are followed next. This way, the structure of the ontology provides a complex measure of 'relatedness', which supports a focused crawling process, similar to the typical browsing behavior of a a human user.

He decides to send the crawler to search for new material on Knowledge Management and Peer-to-Peer. For this, he selects the corresponding concepts as well as other concepts that seem to be relevant in both domains. Then he launches the crawler at the homepages of the European projects Ontoweb and SWAP which he considers as good starting points. The retrieved results can finally be browsed by using the same ontology.

[3] See for instance the efforts of the Learning Technology Standard Committee.

[4] `http://www.acm.org/class/1998`

[5] In this scenario we will take Edutella as a prototypical distributed network of semantically annotated learning material. Another example is POOL, see [11].

### 3.1.3 Clustering the Resources

Professor Meyer now wishes to organize the learning resources he has retrieved. Ideally, he wants to group similar documents together and structure the document collection according to certain criteria. He may use available clustering algorithms, but he does not want to ignore the additional background knowledge encoded within the ontology. For instance, he would expect that a lecture about 'Machine Learning' and another lecture about 'Data Mining' are grouped in the same cluster, even though they may not share many technical expressions. The ontology will then bridge the gap, as it subsumes both terms under 'Knowledge Discovery'.

Moreover, it is important for him to be able to understand the clusters. For this he needs specific visualization techniques, which allow him to understand why the documents have been grouped together. Here again, the ontology can be used to structure the presentation.

### 3.1.4 Ontology Evolution

Finally, Professor Meyer wants to be aware of the ongoing evolution of the vocabulary of his field. His ontology should reflect the changes and should also include upcoming topics. In the web, one can discover upcoming topics by new terms showing up within the retrieved documents, or by terms which existed before but now increase in frequency. The decision if a topic is relevant for Prof. Meyer, and if it has to be included in the ontology, is finally up to him, but seen the large number of upcoming terms, he would expect some tool support.

## 3.2 User Requirements

From the preceding scenario, one can derive several tasks that need to be supported by a Courseware Watchdog. These are summed up in the following list.

1. Understanding the ontology and browsing the content,
2. retrieving relevant material through focused crawling,
3. querying semantically annotated resource repositories,
4. clustering and organizing the documents according to the ontology,
5. updating the ontology and knowledge base according to current data.

The first task requires a user interface for accessing the ontology and the collected resources. The next two tasks deal with the acquisition of resources: querying P2P repositories and retrieving elements from the Web through crawling. These two tasks can be considered as complementary. Since in both cases the interaction with the ontology is necessary, they have to interact with the browsing interface. The same holds for the last two tasks. Here again, interaction with the user goes along the ontology.

Since ontologies show up in all these tasks, it is rather natural to use a representation of the ontology as central part of the user interface. Whenever one of the other four tasks is performed, then this ontology representation has to be complemented by a second interface specific to the task. By keeping the ontology a constant part of the interface, it serves as a mental fixpoint and thus facilitates the orientation of the user.

Moreover, the scenario shows that there is a strong interaction between these five tasks: The crawling task needs the ontology as background knowledge, and populates the text corpus. The corpus will be structured by the crawling task, and serves itself as input to the ontology update task, which, it its turn, populates the ontology. The updated ontology can be used to launch again the focused crawler, or to retrieve resources in the P2P network. The step from one task to the next has to be made explicit in the user interface to support the user in keeping the orientation in the process.

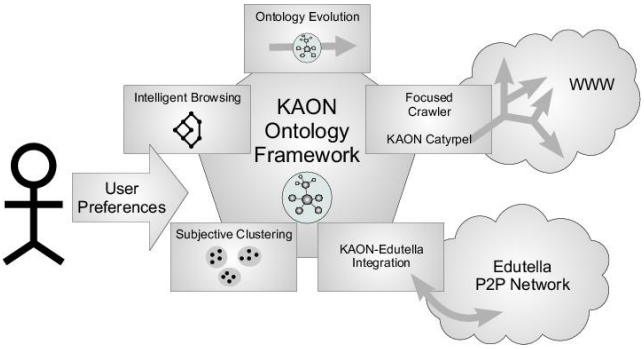


**Figure 1: The components of the Courseware Watchdog.**

In the rest of this paper, we will describe how these requirements are turned into an architecture, and how this architecture is implemented within our Courseware Watchdog. The next section gives the overall picture of the Watchdog; sections 5 to 8 discuss the different parts in more detail. Each section provides first a conceptual view of the part, before discussing implementation details.

# 4. Courseware Watchdog

The Courseware Watchdog described in this paper addresses the abovementioned requirements by using a comprehensive approach which exploits concepts from the Semantic Web, such as ontologies, in an E-Learning scenario [28]. It is part of the PADLR project (Personalized Access to Distributed Learning Repositories) that builds upon a peer-to-peer approach for supporting personalized access to learning material[6].

When developing the Courseware Watchdog, we aimed at addressing the different problems evoked by the previous scenario. The tasks to be solved are addressed by different modules. One important goal was to use a single semantic model to integrate the different tools. We show that their combination offers the user a single simple tool for tasks depending on each other. The Courseware Watchdog consists of the following components which are organized around an ontology management system (see figure 1):

1. *Visualization and interactive browsing techniques* allow for the browsing of the ontology and knowledge base in order to improve the interaction between of the user with the content.
2. A *focused crawler* finds related web sites and documents that match the user's interests. The crawl can be focused by checking new documents against the user's preferences as specified in terms of the ontology.
3. An *Edutella peer* enables querying for metadata on learning objects with an expressive query language, and allows to publish local resources in the P2P network.
4. A *subjective clustering component* is used to generate subjective views onto the documents.

[6] `http://www.learninglab.de/english/projects/padlr.html`

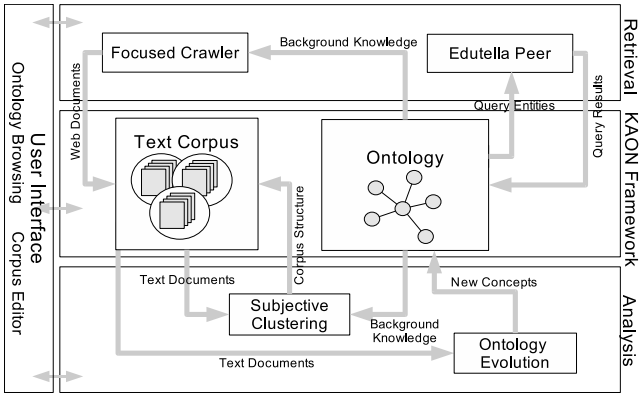


**Figure 2: The architecture of the Courseware Watchdog.**

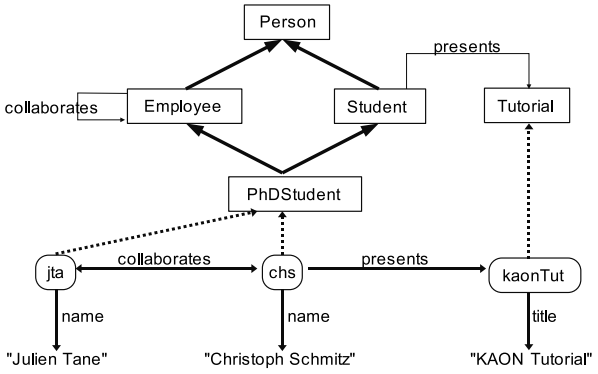


**Figure 4: An ontology example.**

5. An *ontology evolution component* comprises ontology learning methods which discover changes and trends within the field of interest.

The components are not separated but there is a logical data flow between them, as discussed in the previous section. This is reflected in the architecture of the Watchdog, which is shown in figure 2. The Watchdog is organized around the ontology and the text corpus. Section 5 discusses how these two are modeled using the KAON framework.

Both the ontology and the text corpus can be accessed by the user interface described in section 6. A screenshot of the interface is shown in figure 3. The left part of the screen always shows the ontology; it serves as a fixpoint for the interaction of the user with the system. The right part of the screen changes, depending on which of the components crawler, Edutella, clustering, or evolution is currently active. These four components and their interaction with the other components are discussed in sections 7 and 8.

*Implementation Details.* The Courseware Watchdog application is built on top of the KAON Workbench (see section 5). The workbench offers a plug-in interface for extension modules, which can make use of each other and the basic KAON abstractions.

All components of the Watchdog mentioned in this paper are implemented as KAON modules.

# 5. THE KAON FRAMEWORK

All the modules mentioned are built on top of an ontology tool suite named KAON (the Karlsruhe Ontology and Semantic Web Framework) [9]. KAON offers abstractions for ontologies and text corpora, an ontology editor and application framework, inferencing, and persistence mechanisms, etc.

On this platform, integration in the Courseware Watchdog is achieved on several levels:

- at the semantic level – through ontologies
- at the web structure level – the structure of the graph of web documents is stored in an ontology
- at the structure level of the corpus – the different algorithms for the clustering and ontology evolution use the same corpus model

This common integration model allows to use both the browsing and the querying of the resources available or discovered. Or it allows the interaction with the results of algorithm. For instance, it is then possible to use the clustering results as input to the ontology evolution.

## 5.1 Ontologies

KAON ontologies – called OIModels (ontology-instance-models) – consist of concepts, properties relating concepts, instances, and property instances, all of which are referred to as *entities*. These can be used to model a domain of interest and can be expressed canonically in RDFS. Figure 4 shows a small example of a KAON ontology.

Additional noteworthy features of KAON ontologies are

**OIModel inclusion mechanism** OIModels can be nested; this means that an outer OIModel can refer to any entity in another model it includes, but not vice-versa. This facilitates modularization of modeling tasks, as well as ontology reuse.

**Lexical layer** Lexical information about entities is dealt with in a lexical OIModel which models multilingual labels, word stems, and documentation of entities, and can be used via the inclusion mechanism.

For more details on the ontology model, as well as on the KAON API, we refer to [9], or to the KAON Developer documentation [1].

## 5.2 Text Corpora

Text-to-Onto is a KAON extension related to text mining tasks[7] which offers a text corpus abstraction. A text corpus contains a set of documents, i. e. their full text, metadata such as content type, file name or URL, etc. Like ontologies, corpora can be nested. The structure of a corpus is described as an OIModel and can thus be handled with KAON tools.

# 6. THE USER INTERFACE: BROWSING THE WATCHDOG DATA

As shown in the scenario, the interaction of the user with the ontology is crucial for all ontology-based tools. In the Courseware Watchdog, this is done using the browsing component. In this section, we will first explain how the ontology is displayed. Then we describe how it can be used to interact with the ontology.

[7] `http://sourceforge.net/projects/texttoonto/`

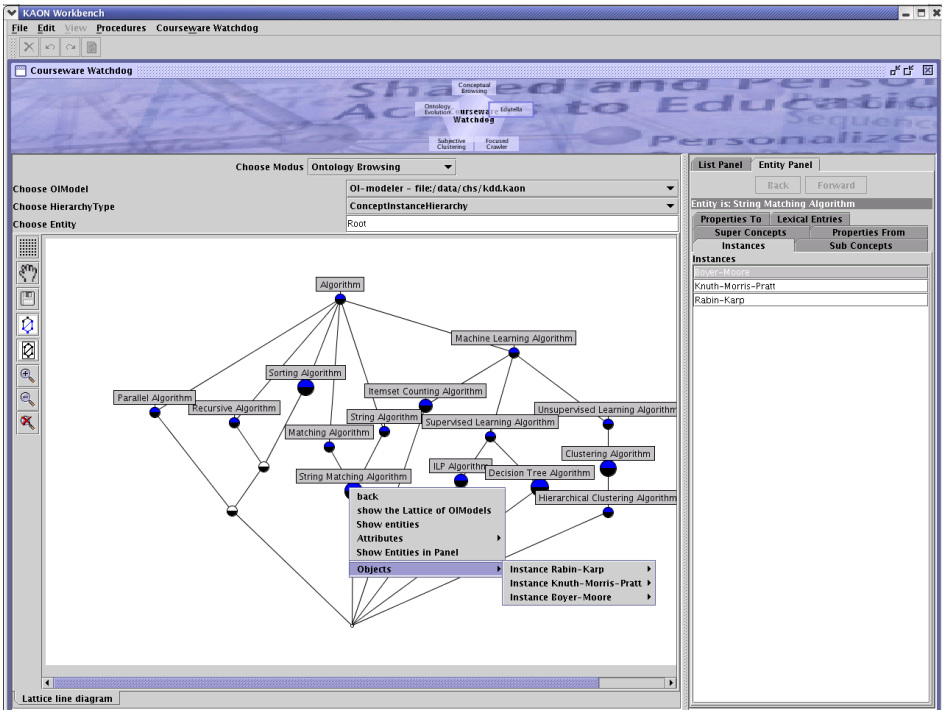


**Figure 3: The browsing interface of the Courseware Watchdog**

## 6.1 Displaying an Ontology

Ontologies are complex structures based two kinds of relations: *hierarchical* and *non-hierarchical* relations. For each kind of relation, we use an appropriate technique: the display of hierarchies through concept lattices in the first case, and relational browsing otherwise.

In order to display specific hierarchies[8], we use techniques from Formal Concept Analysis (FCA) [10, 30], a conceptual clustering technique which allows the display of hierarchies of concepts using lattices. We create the minimal lattice containing the hierarchy. This allows for the display of multiple inheritance. By adding instances, we get new concepts which also reflect eventual multiple instantiation. Using these techniques, some new concepts may appear, even if they were not explicitly modelled within the ontology. For example, 3 displays the hierarchy of subconcepts of "algorithm". One can see that there are instances of "sorting algorithms" which are also "recursive" and "parallel" in the current knowledge base. This suggests that a new concept might be useful in this position (for details on ontology evolution, see section 8.2).

This approach follows the conception of the Conceptual Email Management system CEM [7] which supports exactly such a kind of navigation in collections of emails. We extended this approach to the use of ontologies. When applied to learning material, multiple inheritance within this hierarchy provides a rich conceptual landscape for navigating and retrieving the educational media.

Non-hierarchical relations existing in the ontology represent links between various elements of the ontology (e. g., the "lecturer" of a "course" should be linked by a relation "holdsCourse"). These kind of relation are best understood and used through some kind of exploration. Relational browsing is a technique offering the user different links which he can choose to follow, but in addition to normal browsing, the links are typed according to the ontology instead of the documents. It is possible to navigate and explore the ontology using the relations of the ontologies, and then display different kinds of hierarchies according to one's needs. By clicking on the concept "String matching algorithms", one is, e. g., able to find its instances, such as the "Boyer-Moore" algorithm, or find the lectures which refer to this algorithm by following the relation "isReferredBy" of the instance "Boyer-Moore"(selected in the right panel of figure 3.).

[8]For example, we display concept, property or topic hierarchies.

## 6.2 Interacting with the Ontology

Beside just displaying the ontology, the browsing component is also a generic way of interacting with the ontology. Therefore, this component plays a central role in the Courseware Watchdog. Indeed, we divided the main Watchdog panel in two parts. The left hand side always presents the browsing component, while the content of the right part depends on the actual usage.

We defined different modes, in order to simplify the interaction for the user. These modes correspond to the different tasks explained in 3.2. In each mode, the user sees on the left hand side the lattice of some kind of relation. He may click on the nodes of the lattice to select them in order to reuse them later, in another context, or he may select one node, corresponding to a concept, and display more information on the right hand side. In the context of the evolution, he might introduce, as a subconcept of the selected concept, a new concept detected with the evolution module. In figures 3,5, and 6, you can see three different combinations corresponding to the browsing, the query refinement, and the annotation of clusters. In combination with these components, the browsing interface supports the construction of queries as well as the selection of entities which can then be used in the crawling, clustering or evolution process.

According to the context of use, the interface allows for the selection of entities and the application of certain actions on these. A typical example of interaction would be to look for documents containing information on certain topics. Then user could then select the topics of interests from the topic hierarchies, and look for the document which are related to these by following the relation "isTopicOf". Once the documents have been found, he can either open them, or insert them into a new or existing corpus.

*Implementation Details.* The ontology browsing mechanism uses the open source FCA software conexp[9] as a library. It was extended to be able to browse ontologies. The creation of the data model uses various strategies to display only the necessary nodes in the lattices. Moreover, it was necessary to introduce dummy instances to the FCA software in order to maintain the structure of the concept lattice and the correctness of the ontology model at the same time. These dummy instances represent potential instances of the model for which there is no example present in the knowledge base. Much effort has been done to integrate it with the other components; according to the context of use, various actions are accessible to the user.

# 7. RETRIEVAL COMPONENTS: FOCUSED CRAWLER AND EDUTELLA

In this section, we will describe the two retrieval components which allow the user to find material according to his interests. In both cases, the ontology browsing component is used to define the elements that should be looked for.

While the focused crawler uses these user preferences to look for relevant terms in the full text of web documents, the Edutella network is used for exchanging metadata on learning resources which have been annotated semantically.

## 7.1 Focused Crawler

A web crawler is a program that collects data from the web automatically by following links extracted from web documents. Thus, a portion of the web is traversed in a breadth-first manner, usually without regarding the relevance of the collected documents with respect to the needs of one specific user.

The Courseware Watchdog includes a web crawler to retrieve learning material from the WWW. In order to restrict the traversal to material relevant to the user, the crawling process is *focused* [4]. Focusing here means preferring those links in the crawling process that appear to be pointing to relevant documents.

The focused crawler of the Courseware Watchdog builds upon crawlers previously developed in our institute [17, 25]. It uses an ontology-based focusing strategy, relying on the KAON environment for ontology-based tools.

The user can specify his preferences by assigning weights to selected entities of the ontology. The preprocessing step of the crawler then uses the background knowledge present in the ontology – namely, relations between concepts or instances – to compute relevance scores for other entities.

For example, a user may specify that he is interested in material on "Machine Learning Algorithms". The preprocessing step of the ontology will then also assign weight to an instance of that concept (e. g. "C4.5"), or to related concepts (e. g. "Algorithm", being a superconcept). Various parameters of this spreading of weight can be manipulated: which kind of relations in the ontology to follow, how large a radius around a user-selected entity to consider, etc.

[9]See `http://www.sourceforge.net/projects/conexp`.

During the crawling process, terms in the web pages as a whole and in the anchor texts of hyperlinks are then compared against the precomputed weight tables (after stop-word removal and stemming). Thus, scores for the web pages and for the hyperlinks on a page are computed.

This enables the use of different levels of "sharpness" in the focusing; e. g., one may decide to present only documents relevant according to a stricter weighting strategy to the user. On the other hand, the decision of which links to follow can be made based on a somewhat softer weighting scheme, so that the crawler is able to "tunnel" over a page which is not relevant in itself, but may point to other relevant documents. This reduces the probability of getting stuck on one single non-relevant page.

The crawling process yields two kinds of results. First, the full text of the documents is stored in a text corpus. From there, it can be used in the clustering (see 8.1) and ontology evolution (8.2) modules.

Second, the metadata of the crawling process – which pages have been crawled, which were the most relevant entities on a page, what is the link structure between pages – are stored in an ontology. From there, they can be presented in tabular form or using the ontology browsing component, or provided in the Edutella module (see 7.2).

*Implementation Details.* The crawler has a main-memory based infrastructure which is loosely modelled after Mercator [12]. To improve performance and at the same time limit the load on individual web servers, it assigns each page to one out of several crawl queues, each of which retrieves pages with a limited frequency, e. g. one page per minute.

The crawling strategy – in our case directed breath first search with ontology-based weighting as described above – is factored out in the source code as a distinct class, so that it is easily replaceable.

## 7.2 Integrating the Edutella Peer–to–Peer Network

The Courseware Watchdog includes the possibility to participate in the Edutella peer-to-peer network. Peer-to-peer (P2P) networks are decentralized networks which allow for publishing and searching resources based on direct collaboration between their nodes.

The Edutella[10] network applies the P2P paradigm to the exchange of structured information about available learning resources [22]. A common data model facilitates the integration of data sources such as relational database systems or XML and RDF repositories. Thus, all of these can act as Edutella peers.

The Edutella module allows the Courseware Watchdog to act in the Edutella network as a consumer as well as a provider of learning resource metadata. Other tools have been extended to act as Edutella peers as well, e. g. the Conzilla concept browser [23], and can thus interoperate with the Watchdog.

### *7.2.1 Edutella Consumer*

While Edutella peers support a powerful query language named *QEL* [24], we chose to offer a simplified, easier-to-use interface to Edutella querying in the Courseware Watchdog (see figure 5).

The user is given an extensible *query repository* of template queries, which he can pose as-is, or, at his choice, partially fill in with concrete values. This means that free variables of the query are replaced by literal values or entities (instances, concepts or properties) from the ontology. This provides a query-by-example interface which enables inexperienced users to pose meaningful queries.

[10]`http://edutella.jxta.org`

Consider the following example, where capital letters indicate variables. The query

```
?-s(Entity, rdfs:type, Type),
  s(Entity, dc:title, Title).
```

(meaning: *Give me all resources that have a type and a title and return the respective types and titles*) can be made more specific by supplying entities to fill in variables (see figure 5).

The user might choose `lom:lecture` via the ontology browsing interface to replace `Type`, thus yielding the specialized query

```
?-s(Entity, rdfs:type, lom:lecture),
  s(Entity, dc:title, Title).
```

*(Give me all lectures and their respective titles)*

Of course, writing QEL queries directly and storing them into the query repository for future reference is also possible for advanced users.

### *7.2.2 Edutella Provider*

An instance of the Courseware Watchdog can also act as a provider of information in the Edutella network.

At the users request, ontologies loaded into the workbench can be offered to the Edutella network, enabling other users to query this Watchdog instance for metadata.

Through use of the KAON ontology inclusion mechanism, the user retains control over what pieces of information he wants to provide to the public domain: if he decides to split his repository into a public part *Pub* and a private part *Priv*, he can still have a consistent view over his material by having *Priv* include *Pub*, and at the same time offer *Pub* to other Edutella users.

*Implementation Details.* Providing Edutella connectivity for the Watchdog posed two challenges. We had to cope with the mismatch of general RDF metadata in Edutella on the one hand versus KAON ontologies on the other hand. Furthermore, we needed to integrate two different RDF APIs.

First, the Edutella network is about RDF metadata in general, while the Courseware Watchdog deals with KAON ontologies represented using a subset of RDFS. But as the Edutella community has agreed on a subset of LOM-RDF [8] which fits into the KAON ontology language, only minor adjustments were needed. These include the treatment of labels (which are dealt with differently in KAON and RDFS) and containers (Bag, Seq, etc.). Containers are not a part of the KAON language, but can be approximated to provide a very similar functionality to RDFS containers.

Should RDF statements outside the KAON ontology language be retrieved from Edutella, these are incorporated into the RDF model, but will not be visible in the ontology-based user interface.

Second, KAON comes with its own RDF parser and API, whereas Edutella relies heavily on the Jena RDF library[11]. Thus, we introduced an adapter which wraps the KAON RDF model representing the ontology behind a Jena API. This means that all components other than the Edutella libraries work against a KAON RDF model, while Edutella can be used as-is against the Jena interface to the same model.

## 8. ANALYSIS OF RETRIEVED DATA

Once the lecturer has collected a certain amount of material automatically from remote resources or from his own computer, they are stored as instances in the knowledge base of his ontology. The resources will have to be organized according to their topics. Of course, he will not want to spend much time organizing his documents, Moreover he will have to let the ontology evolve according to the resources he has. For this, we describe in this section two solutions which complement each other: a subjective clustering technique and strategies to manage the evolution of the ontology.

### 8.1 Subjective Clustering

As mentioned earlier, the lecturer needs to organize the learning resources available on his computer or remotely (for example, lectures he retrieved through the focused crawler or Edutella). He is interested in having his documents grouped according to their topics and similarities. However, standard text clustering techniques cluster only using document/term matrices and thus much of the implicit information contained in the language gets lost. A remedy to that problem is to introduce background (or domain) knowledge into the clustering process. Ontologies contain such implicit information which can then be used by the clustering algorithm to group related documents more efficiently. Therefore, we have recently developed clustering mechanisms that allow to provide *subjective views* onto document collections [14, 15, 16], which are based on an underlying ontology. These techniques can be seen as creating views on the clustered resources, thus using the ontology as a means to specify individual interests.

For instance, one view may concentrate on differences and similarities of the content of learning material, while another view may concentrate on its presentation form, or on the levels of skills and experiences needed. The lecturer can then use the first view to select the material which addresses the topics which are most relevant to his planned course. He might then use the second view in order to see how the material is distributed over different types of material like presentation slides, exercise sheets, or online demonstrations. Moreover, it is possible to get an idea of the topic of the cluster by using a visualization technique based on Formal Concept Analysis which we presented in [15]. It displays the distribution of the most relevant terms of the various clusters through the use of a lattice displaying the various combinations of terms occurring in the clusters. This combination of the browsing and the clustering results helps the user to understand better the results of the clustering process and select in a simple way the lectures which interest him.

This allows for a very simple interaction with clustering results and achieves the goal of helping the user in organizing his learning material. For example, figure 6 shows how the user, after having selected two documents, can relate them to the concept “Decision Tree algorithm” through the relation “topicOf”.

The subjective clustering is following a five step approach:

1. Parsing the text, collecting interesting term statistics
2. Building multiword terms (optional)
3. Adding ontological information (various strategies)
4. Clustering documents (with a Bisection K-Means algorithm)
5. Visualization with FCA

*Implementation Details.* The first two parts are using the functionalities of TextToOnto. These are explained in the evolution section 8.2. The third part takes the terms found in the documents and tries to find matching concepts in the ontology. The terms are then weighted using an adapted version of the tfidf measure and the

[11] `http://jena.sourceforge.net`

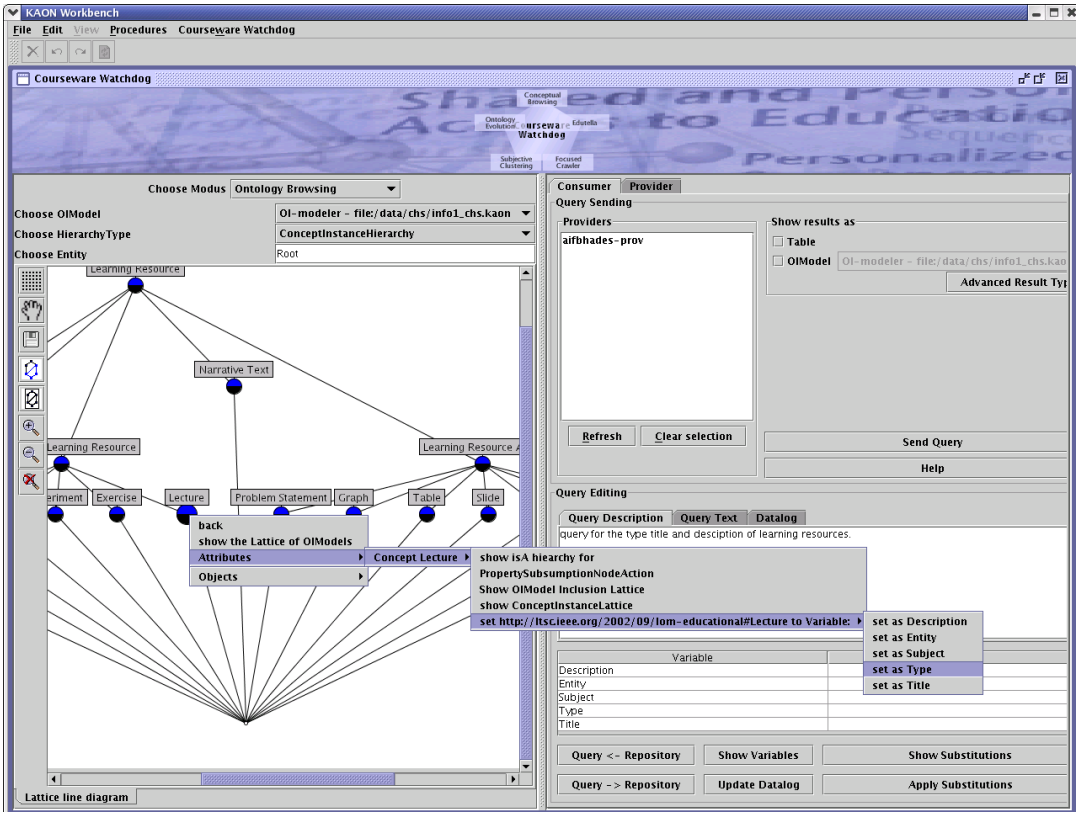


**Figure 5: Refining a query.**

new document vector representation can be used in an implementation of the Bisection K-Means algorithm[12]. Finally, the result can be visualized as a lattice displaying the distribution of the clusters according to their most relevant concepts. For this, we reuse the functionalities of the ontology browsing presented earlier in this paper.

## 8.2 Ontology Evolution

The Courseware Watchdog as presented in this paper so far builds heavily on a proper ontology that reflects what the user is interested in. However, over time such interests will invariably change together with the teaching/learning subject itself. Therefore, the ontology and the topics represented therein need to be updated. One must deal with several requirements incorporated in such updates:

*Modifying the ontology:* The ontology must remain consistent at all times. We use the evolution functionalities of the KAON API, which ensure that changes to the ontology will not corrupt it.

More details about the maintenance and reuse of evolving and distributed ontologies in KAON can be found in [18].

*Introducing new concepts:* The first requirement is about ($i$) recognizing that a new concept (e. g. a new topic) has appeared in the course material available in the network or on the Web, ($ii$) inserting this concept into the right place of the taxonomy, and ($iii$) linking it via further relations to other concepts.

We use methods described in [19] to find relevant concepts. Moreover, we are working on techniques to create ontologies semi-automatically. We will couple current hierarchy building techniques as we presented in [5] and [6] with the courseware conceptual browsing.

For instance, Web Services are today an emerging topic, and will probably have to be included in future courses on the Semantic Web. Hence 'Web Services' will be recognized as a term that denotes a new concept, since it occurs frequently in documents on the Semantic Web. It can be inserted into the concept hierarchy (e. g. as a subarea of computer science). It also must be related to other disciplines (e. g. to business process modeling and E-Business). The user can select, on the left hand side, the place where he wants to insert the new concept or instance. Then he can relate the new instances or concepts to other concepts or instances by selecting in the context menu of the new instance the place where he wishes to insert the given concept.

*Implementation Details.* The technical implementation of the evolution component uses the functionalities of the KAON API to guarantee the coherence of the ontology. In the KAON API, a special care has been taken to make sure changes to the ontology reflect user changes while preserving the logical coherence of the ontology level. This has been realized through different means. The user can choose various strategies for the evolution of the model and instance bases. For instance, he can determine if instances of a deleted concept should be deleted also or whether they should be attached to its superconcepts. The users can choose the strategy which is most suitable for his workflow.

[12]We use the WEKA Data Mining framework which we adapted to our needs. For Weka, take a look at: http://www.cs.waikato.ac.nz/ml/weka/.

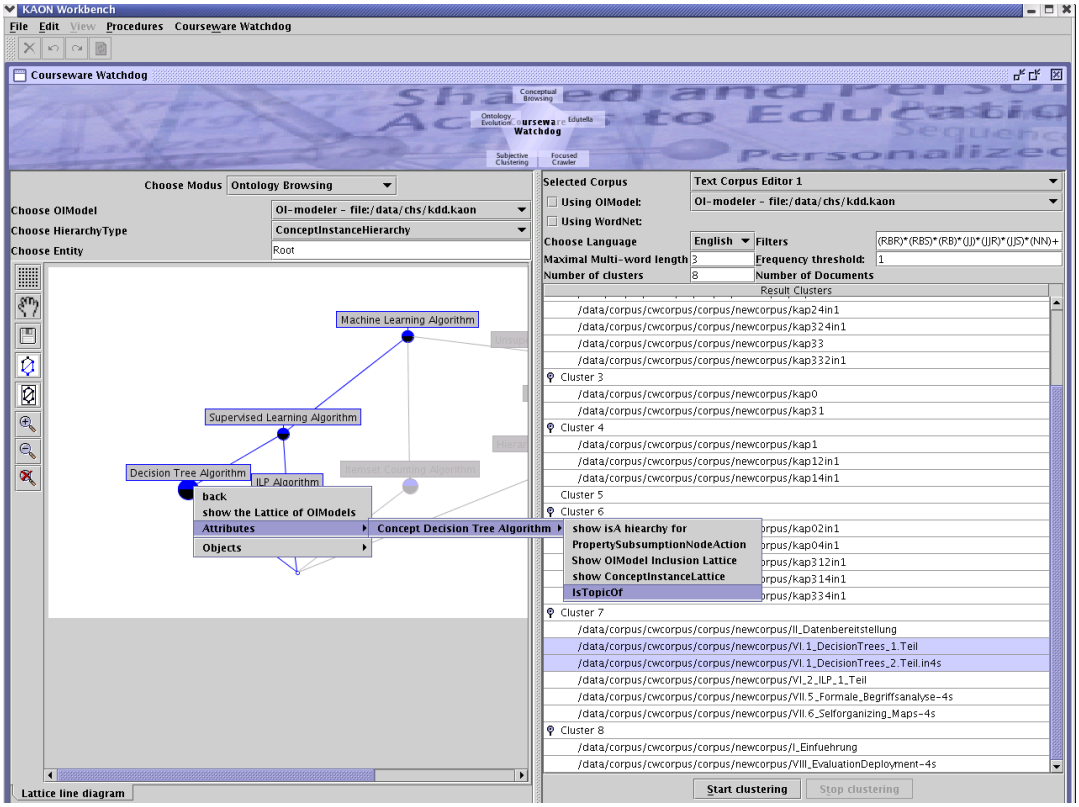


**Figure 6: Simple annotation helped by clustering.**

The evolution component integrates the use of TextToOnto, a KAON component designed to help users in creating ontologies out of texts (see also section 5). We and other members of the TextToOnto team are still improving the functionalities of the ontology learning functionalities. We are particularly interested how the other Watchdog components can be exploited for ontology learning with TextToOnto.

# 9. CONCLUSION

The Courseware Watchdog is a comprehensive approach for supporting the learning needs of individuals in fast changing working environments, and for lecturers who frequently have to prepare new courses about upcoming topics.

As shown in the paper, the Courseware Watchdog addresses the different needs of teachers and students to organize their learning material. It integrates, on the one hand, the Semantic Web vision by using ontologies and a peer-to-peer network of semantically annotated learning material. On the other hand, it addresses the important problems of finding and organizing material using semantical information. Finally, it offers a first approach to the problem of evolving ontologies.

The components of the Courseware Watchdog need further improvement. For instance, focused crawling has to be improved by offering further measures for computing the relevance of documents based on ontologies and available metadata, and ontology evolution needs further techniques for better reflecting changes in the underlying learning material, such as concept drift detection.

Overall, the Courseware Watchdog indicates how a Semantic Web based approach increases the support of retrieval and management of remote (learning) resources, by providing tools for discovering and organizing them.

# Acknowledgements

We thank all members of our groups at AIFB and FZI for intensive discussions and contributions, especially Marc Ehrig, Andreas Hotho, Boris Motik, and Philipp Cimiano.

This research is partially funded by the German Federal Ministry of Education and Research (bmb+f).

# 10. REFERENCES

[1] KAON – the Karlsruhe Ontology and Semantic Web Framework – Developer's Guide. `http://kaon.semanticweb.org/Members/rvo/KAON_Dev_Guide.pdf`, March 2003.

[2] Wolf Siberski Benjamin Ahlborn, Wolfgang Nejdl. OAI-P2P: A peer-to-peer network for open archives. In *ICPP Workshops*, pages 462–468, 2002.

[3] Jan Brase and Wolfgang Nejdl. *Ontologies and Metadata for eLearning*, pages 579–598. Springer Verlag, 2003.

[4] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. *WWW8 / Computer Networks 31(11-16)*, pages 1623–1640, 1999.

[5] Philipp Cimiano, Julien Tane, and Steffen Staab. Automatic acquisition of taxonomies from text: FCA meets NLP. In *Proceedings of ECML/PKDD Workshop on Adaptive Text Extraction and Mining*, Cavtat-Dubrobnik, Croatia, 2003.

[6] Philipp Cimiano, Julien Tane, and Steffen Staab. Deriving concept hierarchies from text by smooth formal concept analysis. In Data Mining Fachgruppe Machinelles Lernen, Wissensentdeckung, editor, *Proceedings of GI Workshop Lernen -Wissen -Adaptivität (LLWA)*, pages 72–79, Karlsruhe, Germany, 2003.

[7] Richard Cole and Gerd Stumme. CEM - a Conceptual Email Manager. In Bernhard Ganter and Guy W. Mineau, editors, *Proc. ICCS 2000*, volume 1867 of *LNAI*, pages 438–452. Springer, 2000.

[8] IEEE LTSC/Mikael Nilsson (ed.). RDF binding of LOM metadata. `http://kmr.nada.kth.se/el/ims/metadata.html`, to be published at `http://ltsc.ieee.org/wg12`, January 2003.

[9] Errol Bozsak et al. KAON – Towards a Large Scale Semantic Web. In G. Quirchmayr (Eds.) K. Bauknecht, A. Min Tjoa, editor, *: Proc. of the 3rd Intl. Conf. on E-Commerce and Web Technologies (EC-Web 2002)*, pages 304–313, 2002.

[10] Bernhardt Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin – Heidelberg, 1999.

[11] Marek Hatala and Griff Richards. Pool, pond and splash: A canadian infrastructure for learning object repositories. In *Proceedings od the 5th IASTED Int. Conference on Computers and Advanced Technology in Education (CATE 2002)*, pages 54–59, Cancun, Mexico, 2002.

[12] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, December 1999.

[13] Ian Horrocks and Jim Hendler, editors. *The Semantic Web - ISWC 2002. Proceedings of the First International Semantic Web Conference*. Springer Verlag, Sardinia,Italy, 2002.

[14] Andreas Hotho, Alexander Maedche, and Steffen Staab. Ontology-based text clustering. In *Proc. of the Workshop "Text Learning: Beyond Supervision" at IJCAI 2001. Seattle, WA, USA, August 6*, 2001.

[15] Andreas Hotho, Steffen Staab, and Gerd Stumme. Explaining text clustering results using semantic structures. In *Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003*, Dubrovnik, Croatia, 2003.

[16] Andreas Hotho and Gerd Stumme. Conceptual clustering of text clusters. In G. Kokai and J. Zeidler (Eds.), editors, *Proc. Fachgruppentreffen Maschinelles Lernen (FGML 2002)*, pages 37–45, Hannover, 2002.

[17] Alexander Maedche, Marc Ehrig, Siegfried Handschuh, Raphael Volz, and Ljiljana Stojanovic. Ontology-focused crawling of documents and relational metadata. In *Proceedings of the Eleventh International World Wide Web Conference WWW-2002*, May 2002. (Poster).

[18] Alexander Maedche, Boris Motik, and Ljiljana Stojanovic. Managing multiple and distributed ontologies on the semantic web. *VLDB Journal*, 12(4):286–302, November 2003.

[19] Alexander Maedche, Viktor Pekar, and Steffen Staab. Ontology learning part one — on discoverying taxonomic relations from the web, 2002.

[20] Hermann Maurer and Marianne Sapper. E-learning has to be seen as part of general knowledge management. In *Proceedings of ED-MEDIA 2001*, pages 1249–1253, Charlottesville, USA, 2001. AACE.

[21] Wolfgang Nejdl. Semantic Web and Peer-to-Peer Technologies for Distributed Learning Repositories. In *17th IFIP World Computer Congress, Intelligent Information Processing / IIP-2002*, 2002.

[22] Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Ambjörn Naeve Michael Sintek, Mikael Nilsson, and Tore Risch Matthias Palmér. Edutella: A P2P networking infrastructure based on RDF. In *Proc. 11th International World Wide Web Conference (WWW 2002)*, pages 604–615, Honolulu, Hawaii, May 2002.

[23] Mikael Nilsson and Matthias Palmér. Conzilla - towards a concept browser. Technical Report (Internal report), Centre for user-oriented Information technology Design (CID), Nada, KTH [CID-53, TRITA-NA-9911] and Department of Scientific Computing (TDB), Uppsala University, 1999.

[24] Mikael Nilsson and Wolf Siberski. RDF query exchange language (QEL) - concepts, semantics and RDF syntax. `http://edutella.jxta.org/spec/qel.html`, June 2003.

[25] Christoph Schmitz. Untersuchung der Graphstruktur von Web-Communities am Beispiel der Informatik. Master's thesis, University of Trier, Trier, Germany, October 2001.

[26] Christoph Schmitz, Steffen Staab, Rudi Studer, Gerd Stumme, and Julien Tane. Accessing distributed learning repositories through a courseware watchdog. In M. Driscoll and T. C. Reeves, editors, *Proc. of E–Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2002)*, pages 909–915, Norfolk, 2002. AACE.

[27] Rudi Studer Steffen Staab, Hans-Peter Schnurr and York Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1), 2001.

[28] Ljiljana Stojanovic, Steffen Staab, and Rudi Studer. E-learning based on the semantic web. In *WebNet2001 - World Conference on the WWW and Internet*, Orlando, Florida, USA, 2001.

[29] Rudi Studer and Steffen Staab, editors. *Handbook on Ontologies in Infornation Systems*. Springer Verlag, Berlin – Heidelberg, 2003.

[30] Gerd Stumme and Rudolf Wille, editors. *Begriffliche Wissensverarbeitung — Methoden und Anwendungen*. Springer, Heidelberg, 2000.