Ontology-Based Metadata Generation from Semi-Structured Information

Heiner Stuckenschmidt
Center for Computing Technologies
University of Bremen
P.O.B: 33 04 40 D-28334 Bremen, Germany
heiner@tzi.de

Frank van Harmelen
AIdministrator BV, Amerfoort, and
AI Department, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands
frankh@cs.vu.nl

Abstract

Content-related metadata plays an important role in intelligent information systems. Especially on the world-wide web meaningful metadata describing the contents of a web-site is the key to intelligent retrieval and access of information. Metadata description standards like RDF and RDF schema have been developed and work in progress addresses the use of ontologies to provide a logical foundation for metadata. However, the acquisition of appropriate metadata is still a problem. The main part of the paper is concerned with the specification of ontologies and metadata models. We describe the Spectacle approach, a knowledge-based approach for metadata validation and generation as well as tools related to the ontology language OIL. We conclude that the specification of ontologies and the generation of metadata models are processes that supplement each other and propose a method for semi-automatic generation of metadata models on the basis of ontologies.

Motivation

The information society demands large-scale availability of data and information. With the advent of the World Wide Web, huge amounts of information is available in principle, but the size and the inherent heterogeneity of the Web make it difficult to find and to access useful information. A suitable information source must be located which contains the data needed for a given task. Once the information source has been found, access to the data therein has to be provided. A common approach to this problem is to provide so-called metadata, i.e. data about the actual information. This data may cover very different aspects of information: technical data about storage facilities and access methods co-exist with content descriptions and information about intended uses, suitability and data quality. Concerning the problem of finding and accessing information, metadata help to find, to access and to interpret information.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

In this paper we focus on a specific type of metadata, namely metadata related to the contents of a web-page. A typical approach to capture this kind of metadata is so-called web-page categorization [10]. Here, web pages as a whole are assigned to a set of classes representing a certain topic area the page belongs to. In order to apply this approach there has to be a set of classes to be used as targets for the classification task. The idea of using ontologies in order to define these classes is straightforward and does not need too much argumentation.

A problem that remains is the classification itself which can be a tremendous effort considering the size of normal web-sites or even the web itself. There is a need for automatic or semi-automatic support for the classification process that has already been observed by others. Jenkins and others, for example, use text mining technology in order to generate RDF models describing the contents of web-pages [9]. It has been argued that web page classification can be significantly improved by using additional information like other kinds of metadata [10] or linguistic features [1]. We propose an approach that exploits another kind of additional information namely the syntactic structure of a web page. This can be done because it has been shown that it is possible to identify syntactic commonalities between web-pages information about similar topics [5]. We use an existing approach for classifying web-pages on the basis of their structure and show how this approach can be used to relate web-pages to a pre-existing ontology in such a way that the formal semantics of the ontology remains available for consistency checking and filtering of web-pages.

The paper is organized as follows. In section we introduce the Spectacle approach for semi-automatically classifying individual web-pages based on their structure. In section we present the use of Spectacle for the generation of contents metadata on the basis of ontologies in some more details. The current state of the technology used as well as two case studies using the approach are the topic of section . We conclude with a critical view on the scaleability of the approach and topics for further research brought up by the case studies.

The Spectacle Approach

We have developed an approach to solve the problems of completeness, consistency and accessibility of metadata identified above. This is done on the basis of rules which must hold for the information found in the Web site, both the actual information and the metadata (and possibly their relationship). This means that besides providing Web site contents and metadata, an information providers also formulate classification rules (also called: integrity constraints) which should hold on this information. An inference engine then applies these integrity constraints to identify the places in the Web site which violate these constraints. This approach has been implemented in the Spectacle Workbench, developed by the Dutch company AIdministrator (`http://www.aidministrator.nl`). In this section, we will describe the different steps of our approach. Formulating and applying classification criteria and integrity constraints is done in a three step process [13].

Constructing a Web-site Model

The first step in our approach to content-based verification and visualization of web-pages is to define a model of the contents of the web-site. Such a model identifies classes of objects on our web-site, and defines subclass relationships between these classes. For example, pages can be about water, soil, air, energy, etc. Each of these types of pages can again be subdivided into new subclasses: water-pages can be about waste-water, drinking water, river-drainage, etc. This leads to a hierarchy of pages which is based on page-contents, such as the example shown in Figure 1.

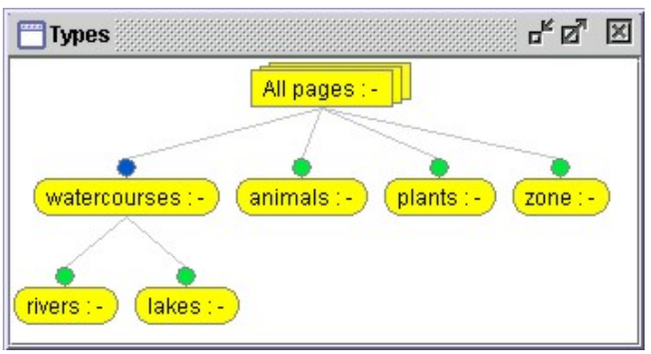

Figure 1: An Example Classification Tree

A subtle point to emphasize is that the objects in this ontology are objects on the web-site, and not objects in the real-world described by the web-site. For example, the elements in the class ”rivers” are not (denotations of) different rivers in a specific region, but they are *web-pages* (in this case: web-pages talking about rivers). As a result, any properties we can validate for these objects are properties of the *pages on the web-site*, as desired for our validation purposes.

Defining Syntactic Criteria for Classes

The first step only defines the classes of our ontology, but does not tell us which instances belong to which class. In the second step, the user defines rules (compare [11]) that determine which Web pages will be members of which class. In this section, we will briefly illustrate these rules by means of three examples.

Figure 2 specifies that a rule is about ”watercourses” if the keyword ”Gewsser” appears in the meta-information of the web-page. The rule succeeds if for example the following code appears in the web-page:

```
<META NAME="Keywords" CONTENT="Gewsser, Bericht">
```

In the typical case, a page belongs to a class if the rule defined for that class succeeds for the page. However, it is also possible to define classes by negation: a page belongs to a class when the corresponding rule fails on that page. This is indicated by a rectangle in the class-hierarchy (instead of a rounded box).

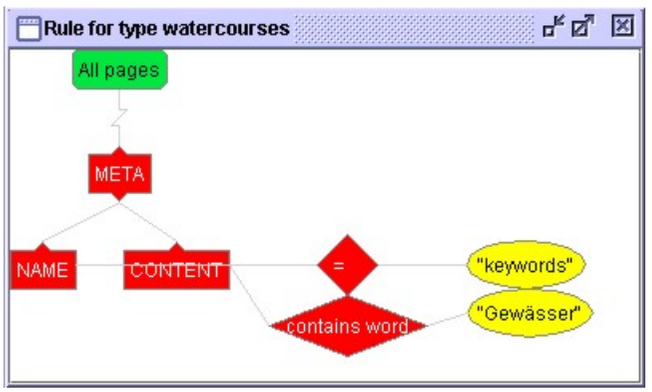

Figure 2: Example of a Classification Rule Using Metadata

Classifying Individual Pages

While the human user of Spectacle performs the previous steps, the next step is automatic. The definition of the hierarchy in the first step and the rules in the second step allow the Spectacle inference engine to automatically classify each page in the class hierarchy. Note that classes may overlap (a single page may belong to multiple classes).

The rule format has been defined in such a way as to provide sufficient expressive power while still making it possible to perform such classification inference on large numbers of pages (many thousands in human-acceptable response time).

Generating Metadata

After these three steps, we have a class hierarchy that is populated with all the pages of a given site. Such a populated class hierarchy can be stored in a combined RDF and RDF Schema format [3]. The following statements are taken from the RDF Schema encoding of the Spectacle type hierarchy. The first three show how of the types "watercourses", "lake" and "river" and their sub-type relationship are encoded in standard RDF Schema.

```
<rdfs:Class rdf:ID="watercourses"/>

<rdfs:Class rdf:ID="lake">
   <rdfs:subClassOf rdf:resource="#water"/>
</rdfs:Class>

<rdfs:Class rdf:ID="river">
   <rdfs:subClassOf rdf:resource="#water"/>
</rdfs:Class>
...
```

The following is an example of an RDF encoding of instance information: the URL mentioned in the "about" attribute is declared to be a member of the class "water" (and consequently of all its super-types, by virtue of the RDF Schema semantics).

```
<rdf:Description about=
    "http://www.umwelt.bremen.de/buisy/scripts/buisy.asp?
    doc=Badegewaesserguete+Bremen">
   <rdf:type resource="#watercourses"/>
</rdf:Description>
...
```

These automatically generated annotations constitute an aggregated description of a web site that can be used to get an overview of its content.

Ontology-Based Metadata Generation

In this section we propose a method to generate content-related metadata in terms of a web-page categorization. The idea behind the method is based on the following observations: Ontologies are intentional models of information contents with a well-defined logical basis which can be used for reasoning. Metadata, on the other hand, are extensional models summarizing existing information and can therefore be extracted from an information source. We conclude that both can supplement each other in the process of generating metadata. In the following we describe an integrated method to generate metadata models on the basis of content ontologies. We illustrate the method with experiments conducted using an existing information system.

Building Content Ontologies

Ontologies have set out to overcome the problem of implicit and hidden knowledge by making the conceptualization of a domain (e.g. environmental protection) explicit. This corresponds to one of the definitions of the term ontology most popular in computer science [7]:

> *"An ontology is an explicit specification of a conceptualization."*

An ontology is used to make assumptions available about the meaning of a term. In the context of the general metadata architecture this means that terms are specified by restrictions on their interpretation and their relation to other terms used in the metadata description. In this section we describe how an ontology about the contents of a web-site can be built and used for reasoning.

The OIL language has been developed in the context of the On-To-Knowledge Project (www.ontoknowledge.org) as a proposal for a language for the specification and exchange of ontologies [6]. OIL tries to provide a core set of features that have been widely accepted to be useful for the description of vocabularies and terminologies. OIL combines object-oriented modeling primitives, reasoning facilities from Description Logics, and a tight interaction with RDF and XML.

A couple of tools have been developed to support the application of the OIL language on the World-Wide Web, including the Ontology Editor OILed which has already been proven useful for real applications [12]. OILed can be used to develop ontologies that contain the following language elements.

Class Hierarchies: The basic structure of an OIL ontology is a set of classes arranged in a subclass hierarchy. Each class is a place-holder for a specific set of entities. We can use class hierarchies to define different sub-disciplines of for example environmental protection, namely emission control, nature preservation, soil protection and water pollution control. These disciplines might be used to structure an information system and can provide guidance for content-based search or navigation. Therefore a clear notion of these terms is important to provide meaningful metadata.

Slot Definitions: OIL is capable of defining binary relations (so-called slots) between classes in the hierarchy. Range and domain of these relations can be restricted to special classes described by their name or an intentional description of their members (see below). Further it is possible to define inverse relations, hierarchies of relations and to assign a couple of mathematical properties (e.g. transitivity) to relations. In our case an 'about' relation, for example, connects disciplines (referred to as *topic area*) to specific contents or spatial locations. Note that we can use Boolean operations on class names to describe this fact.

Concept Definitions: Classes can not only be defined by their position in the class hierarchy, but also by constraints on objects they may relate to. Figure 3 shows a simple class definition.

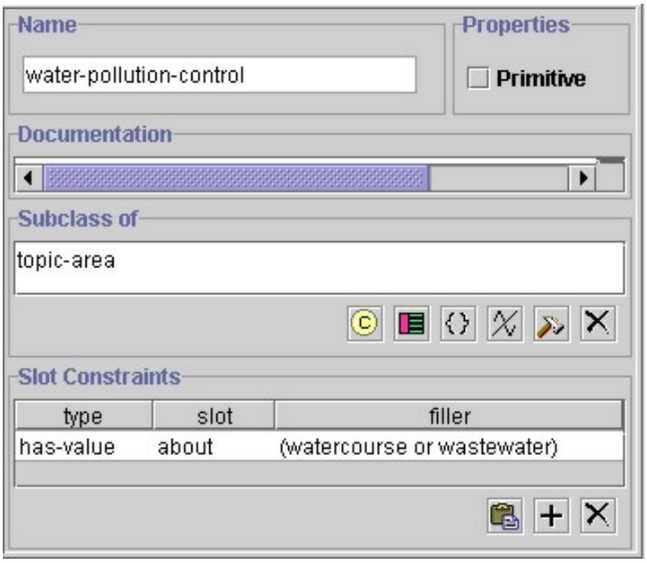

Figure 3: The Class Definition Editor

The figure shows the (strongly simplified) definition of the topic area *water-pollution-control*. The definition claims that the contents of each instance of that topic area are concerned with *watercourses* or with *wastewater*. This definition restricts the way of how a piece of information can be interpreted. For example, it does not allow us to classify an information item which is only concerned with animals as belonging to the topic area of water pollution control.

Individuals: The last feature of the OIL language we need in order to build an ontology about the vocabulary used to describe the contents of a web-based information system (in our case an environmental information system) are instances of classes. In our case, we can use individuals to describe existing objects in the world like real watercourses, lakes and rivers, but also to refer to pages in the information system and relate these pages to real world objects they contain information about. The dummy page shown in the picture, for example, is said to be about the 'Sodenmattsee', a lake in the district 'Huchting'.

Assigning Pages to Classes

The definition of a content ontology provided us with an intentional model of the domain. The next step is to relate this model to real information from the system. This step, also referred to as grounding, is a crucial one, because it is time-consuming and error-prone. The size of modern information systems forces us to provide some tool support. We claim that the Spectacle Workbench is a very helpful tool for this task, because it automatically classifies pages into ontological concepts on the basis of syntactic rules. In order to make use of the system's capabilities we have to import the previously built contents ontology and define classification rules for each concept from the ontology. Note that the ontology already defines criteria for class membership, but does not define criteria that can be checked on a web page. Consequently, we define two sets of criteria for each class in the ontology.

- **Intentional Criteria:** Restrictions on the way a term might be interpreted. This is done in the content ontology.
- **Extensional Criteria:** Properties of information related to that class allowing us to find it in on a web-site: These criteria are specified using the Spectacle System.

In order to use Spectacle for the definition of syntactic criteria, we import the subclass hierarchy from the content ontology into the Workbench and proceed by defining syntactic classification rules for each class. In principle, we have three possibilities:

1. using metadata to classify web-pages
2. using arbitrary page-contents to classify web-pages
3. using external properties of web-pages for classification

The first possibility applies if already some kind of metadata have been included in the system. A typical example is the use of keywords. We discussed this example in section . The rule displayed in figure 2, for example, can be used as a syntactic criterion for the class watercourses. In order to distinguish the sub-types of watercourses we can no longer use existing metadata. In this case, we can use Spectacle to perform a free-text search in the body of web-pages and look for the German terms corresponding to lake and river. Figure 4 shows the result of the search.

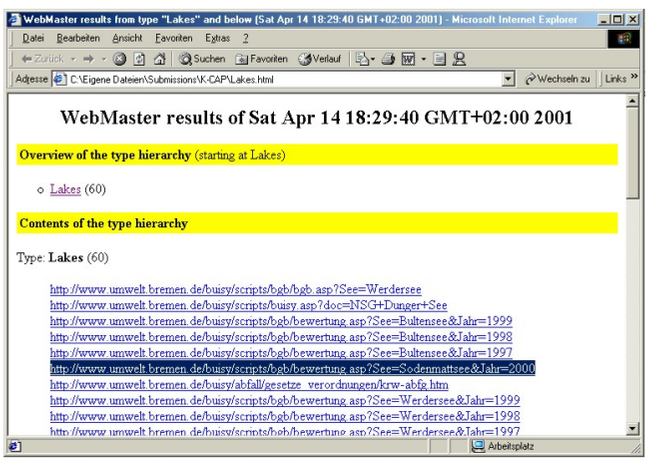

Figure 4: Pages classified to belong to the class 'lake'

From the results displayed in figure 4 we can easily generate a metadata description of the pages classified to belong to the class lakes. The metadata description of the page highlighted in the screenshot is the following:

```
<rdf:Description about=
"http://www.umwelt.bremen.de/buisy/scripts/bgb/bewertung.asp?
        See=Sodenmattsee&Jahr=2000"
   <rdf:type resource="#lakes"/>
</rdf:Description>
```

Corresponding descriptions are generated for all pages on the web-site which could be classified. In parallel, we supplement the contents ontology by creating an individual for every page and assigning it to the corresponding concepts that have been detected.

Ontology-Based Post-Processing

One of the major benefits of using the OIL language for specifying ontologies is the availability of reasoning support for a limited number of tasks concerned with ontology management. The reasoning support is based on Description Logic, i.e. on the correspondence of OIL with the language $\mathcal{SHIQ}$. OIL specifications are translated into this logic and standard reasoning techniques are used to support the following tasks:

Consistency Checking: The reasoner is able to check the satisfiability of the logical model of the ontology. In particular, inconsistent concept definitions are detected. If we, for example, defined animals to have four legs and we try to include an instance of the class animal with five legs, the reasoner will find the contradiction.

Computation of Subclass Relations: An ontology normally contains two different kinds of sub-class relations: explicitly defined relations from the class hierarchy and implicit subclass relations implied by the logical definitions of concepts. The latter can be detected using the reasoning support of the OIL language and included into the ontology thus completing it.

A special case of the computation of subclass relation is the automatic classification of individuals. OIL allows us to describe an individual by its relation to other individuals without naming all classes it belongs to. The reasoner will find the classes we omitted in the definition. An exmaple would be if we only defined our dummy page to be about the 'Sodenmattsee' without assigning it to a special topic area. However, we stated that the domain of the 'about' relation is the class topic area and we defined water-pollution-control to be concerned with watercourses. This information provided and the fact that the 'Sodenmattsee' is a lake and therefore a watercourse enables the reasoner to decide that our dummy page should be classified as belonging to the topic area 'water pollution control'.

OIL uses the FaCT reasoner, a system which implements highly optimized algorithms for providing the above mentioned reasoning support [8]. FaCT is implemented in LISP, but it offers a CORBA interface that allows easy access to the system using a well-defined interface [2]. The OILed Editor can be directly connected with the reasoner providing reasoning support at development time. Inconsistencies are highlighted and missing subclass relations are added. Therefore, OILed and FaCT offer a comfortable development environment for ontologies.

Using this environment we can check the result of the metadata generation for consistency. This is necessary because the criteria used to describe classes in the Spectacle systems only refer to syntactic structures of the page contents. Especially, Spectacle has no possibility to check whether the classification of a page makes sense from a logical point of view. For example, we can include a description of the administrative units in our ontology and classify pages according to the unit which is concerned with the specific topic of the page. We will define the units to be mutually disjoint because the competency is strictly separated. If we now classify one page to belong to both units we get a clash in the logical model. In this case, we have to check the page and assign the right administrative unit by hand. Thus the logical models helps us to find shortcomings of the generated model.

The second benefit of the logical grounding of the metadata model is the possibility to derive hidden class memberships. This is important because the RDF metadata schema makes some assumptions about implicit knowledge. Examples of these assumptions can be found in [4]. We use the following axiom as an example:

$$\frac{\mathcal{T}(r, \texttt{rdf:type}, c_1) \wedge \mathcal{T}(c_1, \texttt{rdfs:subClassOf}, c_2)}{\mathcal{T}(r, \texttt{rdf:type}, c_2)}$$

The equation states that every resource r (i.e. webpage) that is member of class c_1 (indicated by the triple $\mathcal{T}(r, \texttt{rdf:type}, c_1)$) is also member of class c_2 ($\mathcal{T}(r, \texttt{rdf:type}, c_2)$) if c_1 is a subclass of c_2 ($\mathcal{T}(c_1, \texttt{rdfs:subClassOf}, c_2)$). This correlation can easily be computed using the FaCT reasoner by querying all super-concepts of a given concept. The result of this query can be used to supplement the description of a page. The description of the page referred to above, for example, will be extended with the following statement.

```
...
<rdf:type resource="#watercourse"/>
...
```

In the same way, other axiomatic properties of RDF schema can be implemented in order to produce a more complete metadata model.

Applying the Method

The generated metadata model can be used in various ways. In the introduction, we already mentioned the general application areas search, access and interpretation. In this section, we will briefly discuss the use of metadata for intelligent search for web pages. We implemented a universal search engine which relies on an ontology-based metadata model

in order to search for web resources with certain properties. The search engine imports the content ontology and asks the user for a concept to be queried. Based on the definition of that concept (i.e. the attached slots) a query interface is generated that allows the user to specify restrictions on the slot fillers. The query engine searches the metadata model and returns all pages that fall under these restrictions.

The search engine is intended to be used as a component in web-based information systems rather than the complete web. In such a system we can assume the existence of a common ontology which can be used as a basis for generating the metadata model necessary to support the search process.

Tool Support and Interaction

We are currently implementing the approach described above making use of mostly pre-existing technology already mentioned in the previous sections. Figure 5 shows the interaction of these tools, namely the OILed ontology editor, the FaCT reasoning system, the spectacle workbench and our own search engine ASK-Me (Automatic Selection of Knowledge resources based on Metadata).

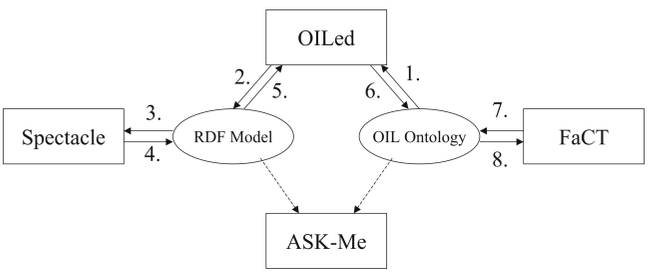

Figure 5: Interaction of Tools in the Overall Process

The figure depicts a typical run through the metadata generation process that contains the following steps.

1. Import of Content Ontology into the ontology editor.
2. Export of the ontology as RDFS model.
3. Import of the Class Hierarchy in to the Spectacle workbench as basis for the classification.
4. Export of instantiated ontology, where each web-page is described and assigned to one or more classes in the hierarchy
5. Import of the instances into the editor in order to supplement the content ontology.
6. Export of the instantiated ontology in OIL format
7. Import of the ontology into the FaCT reasoner for consistency checking and computation of subsumption relations.
8. Export of the verified and completed ontology in OIL

Finally the search engine is supplied with the metadata model as well as with the ontology in order to provide a content filtering service on the basis of a target concept specified by the user. The system uses the Ontology in order to relate the query concept to concepts assigned to web-pages as well as the RDF model in order to retrieve the web-pages assigned to these classes.

At the moment, the right hand side of the figure, namely the interaction between editor, reasoner and search engine is completely implemented. We are currently working on the RDF part. Open tasks include the alignment of the RDF Models supported by Spectacle one and OILed on the other hands. Further we have to extend the search engine to completely work on RDF instead of a relational database we use at the moment.

Case Studies

We have two different case studies we use in order to evaluate the approach presented. The first one the examples found in this paper are taken from is concerned with the environmental information system of the City of Bremen and has already been finalized. The second one is a rather new attempt to provide an integrated information system for scientific services provided by organizations in the city of Bremen. This project called City-of-Science has just started. We briefly describe these case studies in the following.

BUISY: An Environmental Information System The advent of web-based information systems came with an attractive solution to the problem of providing integrated access to environmental information according to the duties and needs of modern environmental protection. Many information systems were set up either on the Internet in order to provide access to environmental information for everybody or in intranets to support monitoring, assessment and exchange of information within an organization. One of the most recent developments in Germany is BUISY, an environmental information system for the city of Bremen which has been developed by the Center for Computing Technologies of the University of Bremen in cooperation with the public authorities. The development of the system was aimed at providing unified access to the information existing in the different organizational units for internal use as well as for the publication of approved information on the internet.

Figure 6 shows the main screen of the BUISY system with the main topic area covered by the system. The first step of the proposed method now consists of the development of an ontology about the domain. The definition of the topic areas and the vocabulary used within these areas is of major interest. In the previous section we already sketched the idea of how such an ontology could be built and showed some example definitions as screenshots from the OILed Editor. The result of this first step will be an extended RDF model that contains additional modeling primitives of the OIL language. Such a model can be generated by OILed without

Figure 6: The main topic areas of the BUISY System

further modeling effort. Below is a corresponding definition of the topic-area water pollution control that we already mentioned in the last section.

```
<rdfs:Class rdf:ID="water-pollution-control">
    <rdfs:subClassOf>
        <rdfs:Class rdf:about="#topic-area"></rdfs:Class>
    </rdfs:subClassOf>
    <oil:hasPropertyRestriction>
        <oil:HasValue>
            <oil:onProperty rdf:resource="#about">
            </oil:onProperty>
            <oil:toClass>
                <oil:Or>
                    <oil:hasOperand>
                        <rdfs:Class rdf:about="#watercourse">
                        </rdfs:Class>
                    </oil:hasOperand>
                    <oil:hasOperand>
                        <rdfs:Class rdf:about="#wastewater">
                        </rdfs:Class>
                    </oil:hasOperand>
                </oil:Or>
            </oil:toClass>
        </oil:HasValue>
    </oil:hasPropertyRestriction>
</rdfs:Class>
```

In the course of our case study on the BUISY system eight groups of AI students with some experience in knowledge representation and knowledge-based systems independently built ontologies covering the contents of the BUISY system. They used the Spectacle System in order to assign web-pages to concepts from the ontology and conducted experiments with querying the system using concept expressions.

City Of Science: An Information System for Scientific Services
The government of the City of Bremen recently recognized the need to support technology transfer from research organizations to the local industry. One of the activities started in connection with this goal is the establishment of an information system for scientific services. The idea is to provide a uniform interface and intelligent access methods to profiles of potential providers of scientific services. A standard profile has been created which each provider has to specify according to the kinds of services he wants to advertise. Examples of information given by each organization are the following:

Type of Organization: Providers of scientific services are categorized due to their legal status and organizational nature. Categories include universities, research institutes, consortia and companies.

Area of Expertise: A rough description of the areas of research the corresponding service provider works in and claims to have expertise.

Technical Equipment: Non standard equipment needed to perform special tasks. Typical examples are laboratories but this notion also includes special function buildings.

There are also other kinds of information like previous projects or mode of funding. However, we only refer to the three properties above.

In a case study, we investigate how contents related metadata can improve the search methods provided to the user in order to find the service he needs. We just finished the development of the content ontology defining the properties mentioned above on a logical basis. Each service provider is modeled as an instance of the concept service-provider with further specifications of the properties using properties from the city of science namespace denoted by `coc`. An example is the following:

```
<coc:ResearchConsortium rdf:about="MARUM">
    <coc:equipment rdf:resource="ResearchShip"/>
    <coc:equpment rdf:resource="ReserachPost"/>
    <coc:expertise rdf:resource="Climate"/>
    <coc:equipment rdf:resource="Laboratories"/>
    <coc:expertise rdf:resource="MarineReserach"/>
    <coc:expertise rdf:resource="EnvironmentalResearch"/>
    <coc:part-of rdf:resource="UniversityOfBremen"/>
</coc:ResearchConsortium>
```

The next step in this case study will be an investigation of how the Spectacle system can be used in order to semi-automatically describe new profiles added by service providers by annotating descriptions like the one shown above.

Discussion and Future Work

We discussed the role of metadata for intelligent search, access, and interpretation of information in web-based information systems. We described the Spectacle approach for the generation of metadata models and the OIL approach to ontology building. We concluded that both approaches can be combined to achieve a semi-automatic procedure to build metadata models. We also described the current status of the implementation and two applications of the approach.

Lessons learned from the case studies: Our approach of combining ontology building with metadata generation comes with benefits for both previously existing approaches. On one hand, metadata generation with Spectacle takes advantage of the logical foundation provided by the ontology in terms of consistency checking and subsumption reasoning. On the other hand it helps to acquire ontological knowledge by providing a tool for the automatic population of the ontology with individuals. The BUISY Case study showed that users with some knowledge in AI are able to build content ontologies and to apply the Spectacle system for generating metadata. However, the definition of syntactical criteria for web-pages of a certain class is still a difficult and time-consuming task which requires some knowledge about the information to be annotated. In order to avoid the effort of analyzing the whole web-site we are currently developing an approach for automatically learning page structure from examples and partial specifications. The city of science case study revealed that it is often not enough to analyze web-pages as a whole. In the case of the city of science project, metadata related to special aspects described in single paragraphs has to be generated. We therefore have to refine the analysis to include single elements on a web-page.

In general, an open problem of the approach is the application to arbitrary web resources. The approach relies on the existence of a single ontology all pages can be related to. At the moment this can only be achieved by restricting the application to information systems with a well-defined domain. Intranets, for example, fulfill this requirement.

REFERENCES

1. Roberto Basili, Alessandro Moschitti, and Maria Teresa Pazienza. Nlp-driven ir: Evaluating performances over a text classification task. In B. Nebel, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI-01*, 2001.

2. S. Bechhofer, I. Horrocks, P. F. Patel-Schneider, and S. Tessaris. A proposal for a description logic interface. In *Proc. of DL'99*, pages 33–36, 1999.

3. D. Brickley, R. Guha, and A. Layman. Resource description framework (rdf) schema specification. Working draft, W3C, August 1998. http://www.w3c.org/TR/WD-rdf-schema.

4. Pierre-Antoine Champin. Rdf tutorial. Available at http://www710.univ-lyon1.fr/ champin/rdf-tutorial/, June 2000.

5. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1-2):69–113, 2000.

6. D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *12th International Conference on Knowledge Engineering and Knowledge Management EKAW 2000*, Juan-les-Pins, France, 2000.

7. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 1993.

8. I. Horrocks. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. Springer-Verlag, Berlin, May 1998.

9. C. Jenkins, M. Jackson, P. Burdon, and J. Wallis. Automatic rdf metadata generation for resource discovery. *Computer Networks*, 31:1305–1320, 1999.

10. John M. Pierre. On the automated classification of web sites. *LInkping Electronic Articles in Computer and Information Science*, 6, 2001.

11. M.-C. Rousset. Verifying the world wide web: a position statement. In F. van Harmelen and J. van Thienen, editors, *Pro-ceedings of the Fourth European Symposium on the Validation and Verification of Knowledge Based Systems (EUROVAV97)*, 1997.

12. R. Stevens, I. Horrocks, C. Goble, and S. Bechofer. Building a reason-bale bioinformatics ontology using oil. In A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors, *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, August 2001.

13. Frank van Harmelen and Jos van der Meer. Webmaster: Knowledge-based verification of web-pages. In M. Ali and I. Imam, editors, *Proceedings of the Twelfth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, (IEA/AEI99)*, LNAI. Springer Verlang, 1999.