



UvA-DARE (Digital Academic Repository)

Logic programming for knowledge-intensive interactive applications

Wielemaker, J.

Publication date
2009

[Link to publication](#)

Citation for published version (APA):

Wielemaker, J. (2009). *Logic programming for knowledge-intensive interactive applications*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

Chapter 9

Semantic annotation and search

About this chapter The material presented in this chapter is composed from two articles about the ICES-KIS project “Multimedia Information Analysis” (MIA). Schreiber, Dubbeldam, Wielemaker, and Wielinga (2001) (section 9.1) appeared in *IEEE Intelligent systems* and Wielemaker, Schreiber, and Wielinga (2003a) appeared as a book chapter in the series *Frontiers in Artificial Intelligence and Applications*.

Both papers study the use of ontologies for image annotation. The annotation and search prototypes were built in SWI-Prolog, using XPCE (chapter 5, Wielemaker and Anjewierden 2002) for the GUI. The knowledge representation is based on RDF, using a pure Prolog-based RDF store which provided the experience for chapter 3 (Wielemaker et al. 2003b). This chapter describes early work in applying emerging technology from the Semantic Web community to annotating multi media objects. It has been included in this thesis as motivation and background for chapters 2 to 4 and 10. The lessons learned are described in section 9.3 at the end of this chapter. The search and user evaluation sections of section 9.1 have been removed because they are considered irrelevant to this thesis.

9.1 Ontology-Based Photo Annotation

*Guus Schreiber, Barbara Dubbeldam, Jan Wielemaker and Bob Wielinga
IEEE intelligent systems, 2001*

9.1.1 Introduction

For magazine editors and others, finding suitable photographs for a particular purpose is increasingly problematic. Advances in storage media along with the Web enable us to store and distribute photographic images worldwide. While large databases containing photographic

images exist, the tools and methods for searching and selecting an image are limited. Typically, the databases have a semistructured indexing scheme that allows a keyword search but not much more to help the user find the desired photograph.

Currently, researchers promote the use of explicit background knowledge as a way out of the search problems encountered on the Internet and in multimedia databases. The *semantic Web* (Berners-Lee 1999) and emerging standards (such as the resource description framework (RDF, Brickley and Guha 2000) make creating a syntactic format specifying background knowledge for information resources possible.

In this article, we explore the use of background knowledge contained in ontologies to index and search collections of photographs. We developed an annotation strategy and tool to help formulate annotations and search for specific images. The article concludes with observations regarding the standards and tools we used in this annotation study.

9.1.2 Our approach

Companies offering photographic images for sale often provide CDs containing samples of the images in reduced jpeg format. Magazine editors and others typically search these CDs to find an illustration for an article. To simulate this process, we obtained three CDs with collections of animal photo samples with about 3,000 photos.

Figure 9.1 shows the general architecture of our annotation tool. We specified all ontologies in RDF Schema (RDFS, Brickley and Guha 2000) using the Protégé-2000 (Fridman Noy et al. 2000) ontology editor (version 1.4). This editor supports the construction of ontologies in a frame-like fashion with classes and slots. Protégé can save the ontology definitions in RDFS. The SWI-Prolog RDF parser (chapter 3, Wielemaker et al. 2003b) reads the resulting RDFS file into the annotation tool, which subsequently generates an annotation interface based on the RDFS specification. The tool supports reading in photographs, creating annotations, and storing annotations in an RDF file. A query tool with a similar interface can read RDF files and search for suitable photographs in terms of the ontology.

The architecture shown in figure 9.1 is in the same spirit as the one described in Lafon and Bos 2000. However, we place more emphasis on the nature of the ontologies, the subject matter description, and the explicit link to a domain ontology.

9.1.3 Developing ontologies

To define semantic annotations for ape photographs, we needed at least two groups of definitions:

- *Structure of a photo annotation.* We defined a *photo annotation ontology* that specifies an annotation's structure independent of the particular subject matter domain (in our case, apes). This ontology provides the description template for annotation construction.
- *Subject matter vocabulary.* We also constructed a *domain-specific ontology* for the animal domain that provides the vocabulary and background knowledge for describing

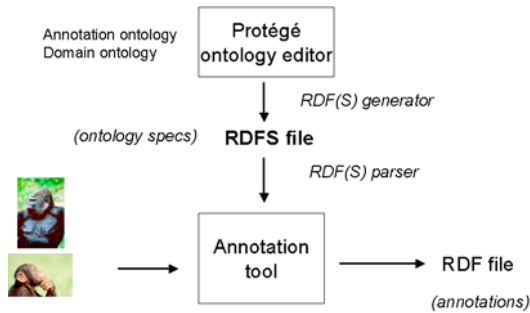


Figure 9.1: Overview of the approach used in this study. We used the Protégé ontology editor to construct ontologies and store them in RDFS format.

features of the photo's subject matter. In this case, the ontology consisted of definitions of the phylum hierarchy of ape species with the corresponding species' attributes and constraints.

9.1.3.1 Photo annotation ontology

The first decision was whether we could use an existing annotation template as a starting point. After evaluating metadata standards such as Dublin Core, (Dublin Core Metadata Initiative 1999) it was clear that they were developed for another purpose and weren't well suited for extensive content-oriented annotations. Because ontology-based annotation is relatively new, we decided to set the existing annotation standards aside and define the annotation ontology based on our own analysis.

When looking at a photo, what kind of things do we want to state about it? We distinguished three viewpoints:

- *What does the photo depict?* We call this the photo's *subject matter feature*. For example, a photo depicts a gorilla eating a banana. This part of the photo annotation ontology links to the domain ontology.
- *How, when, and why was the photo made?* We call this the *photograph feature*. Here, we specify metadata about the circumstances related to the photo such as the photographer or the vantage point (for example, a close-up or an aerial view).
- *How is the photo stored?* We call such photo characteristics the *medium feature*. This represents metadata such as the storage format (such as jpeg) or photo resolution.

In this study, we focused mainly on the subject matter description. In Dublin Core, the single element *subject* represents this aspect.

Figure 9.2 gives an overview of the annotation ontology represented as a UML class diagram. A *photo annotation* contains at least one subject matter description and an arbitrary number of photograph features and medium features. The subject matter description has an internal structure. The actual photograph and medium features are subclasses of the abstract feature concepts. The subclasses, shown in gray, represent just a sample collection of features. The annotation tool only makes a number of minimal assumptions about the annotation ontology. This lets us add new features to the ontology.

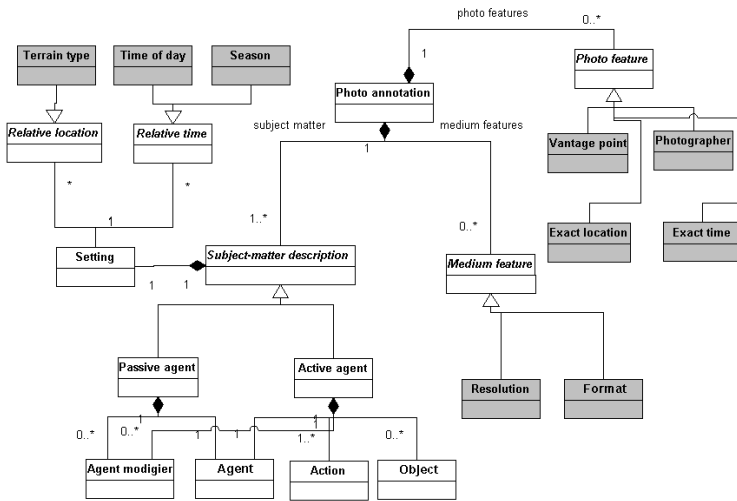


Figure 9.2: Structure of the photo annotation ontology represented as a UML class diagram.

When constructing an ontology, we often needed to incorporate definitions already available in other corpora. For example, to define a *colour* feature, we don't want to type in all the possible values for "colour." A resource such as WordNet (Miller 1995) already contains this information. In this study, we used the WordNet plug-in for Protégé. It provides a cut-and-paste method for importing sections of WordNet into an ontology.

9.1.3.2 Structure of the subject matter description

From the content perspective, the subject matter description is the most interesting part of the ontology. For this, we used the notion of *structured annotation* described in Tam and Leung 2001. They propose a description template consisting of four elements:

1. An *agent*, for example, "an ape." An agent can have modifiers such as "colour =

orange.”

2. An *action*, for example, “eating.”
3. An *object*, for example, “a banana.” Objects can also have modifiers (colour = “green”).
4. A *setting*, for example, “in a forest at dawn.”

We used this general scheme to define two description templates that we found useful in our example domain:

1. A *passive agent* is a restricted form of the scheme with a single agent, any number of agent modifiers, and a setting.
2. An *active agent* is the complete template with a single agent, agent modifiers, an action, optionally an object, and a setting.

The setting is typically restricted to two context dimensions, namely relative time (for example, time of day or season) and relative location (for example, terrain type). Here we copied parts of the WordNet vocabulary.

9.1.3.3 The subject matter ontology

The subject matter ontology describes the vocabulary and background knowledge of the photo’s subject domain. For this study, we developed a domain ontology based on the phylum hierarchy of animal species. Phylum is a metaclass with a number of properties (*slots* in the Protégé terminology). Table 9.1 shows the properties we currently use.

A particular species represents a class that is an instance of metaclass phylum. This organisation of the ontology lets us define instance-type features of species—for example, that an orangutan has an “orange” colour and has as geographical range “Indonesia,” while still being able to treat a species as a class. This sloppy class-instance distinction is a feature of Protégé-2000 that makes it well suited for complex metamodeling.

We specified the phylum hierarchy through subclass relations between species classes. For example, an orangutan is a “great ape,” a subclass of “ape,” a subclass of “primate,” and so forth. Features that are characteristic of apes in general are specified at the hierarchy’s appropriate level and are inherited by the species subclasses.

Figure 9.3 shows a snapshot of the Protégé-2000 ontology editor with the species hierarchy (left) and some characteristics defined for an “orangutan” (right). Sometimes, characteristics are inherited from classes higher up in the hierarchy (for example, the life-stage terms).

9.1.3.4 General terminology

Both the annotation ontology and the domain ontology use general terminology. Instead of defining this ourselves, we used parts of WordNet (Miller 1995) and IconClass (van der

Species feature	Description
Geographical range	The geographical area where the animal typically lives; for example, “Africa,” “Indonesia.”
Typical habitats	The terrain where the animal usually lives; for example, “rain forest,” “savanna.”
Life-stage terminology	Terms used to talk about life stages of animals; for example, “lamb,” “cub.”
Gender terminology	Terms used to talk about male and female animals; for example, “lioness.”
Group terminology	Terms used to talk about a group of these animals; for example, “troop,” “herd.”
Colour features	Description of general colours (“orange”) or colour patterns (“striped”).
Other characteristics	A catch-all category for characteristics such as “captive animal” and “domestic animal.”

Table 9.1: Features of animal species defined in the domain ontology.

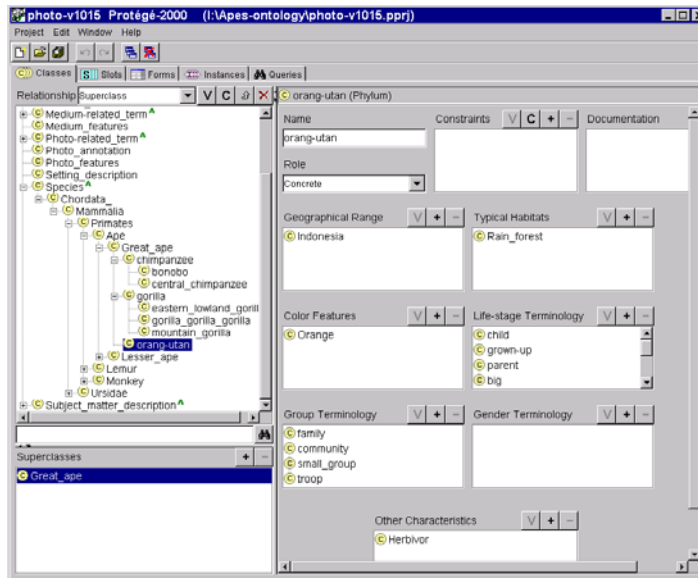


Figure 9.3: Snapshot of the Protégé-2000 ontology editor showing part of the domain ontology.

Waal 1985) —for example, WordNet includes a collection of *vantage points* (a photograph feature). In other cases, WordNet provides a partial value set for a feature value—for example, when we want to describe an ape’s colour aspects, we want to use both general colours (“orange”) as well as animal-specific colour terms (“striped”). Therefore, we can expect that in general we might want to include definitions from many different sources in the ontologies required for annotations. To take another domain, if we want to annotate pictures of art objects, we would like to use the Art and Architecture Thesaurus (AAT) thesaurus, IconClass, and possibly many other sources. This means we need a structured approach for linking domain vocabularies.

Figure 9.4 shows a graphical overview of the ontologies and vocabularies using the UML package notation. The links represent UML dependencies: “*<source>* depends on *<destination>*.” Due to technical limitations we were forced to realise ‘depends on’ by physically importing ontologies due to two problems. First, the Protégé tool does not support ontology modularisation—we can import an ontology by copying, but cannot create a separate module for it. Second, RDFS versions of most vocabularies do not exist. It seems reasonable to expect that such a version of WordNet will become available in the near future, but it is not known whether domain-specific vocabularies (such as AAT) will. The alternative is to write dedicated access routines for vocabularies. In the European GRASP project (“Global Retrieval, Access and information System for Property items”), a CORBA-based ontology server directly links descriptions of art objects to elements of the AAT.

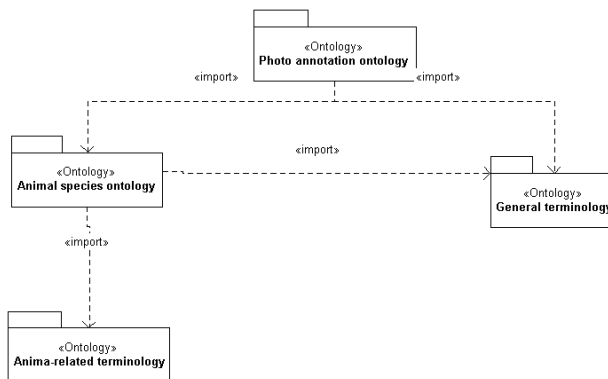


Figure 9.4: Import relations between the ontologies for animal photograph annotations using the UML package notation.

9.1.3.5 Linking the annotation ontology with the domain ontology

To keep the annotation ontology and the subject matter ontology separate, we defined an explicit mapping between the subject matter description in the former ontology to the phylum hierarchy in the later ontology. Figure 9.5 shows part of this mapping. This figure contains a snapshot of the RDFS browser part of the tool we developed. In the figure, we see the description of the RDFS class *passive agent*, a subclass of subject matter description. The class has three properties: the *setting* property links to a resource of type *setting description* (which in turn consists of *relative time* and *relative location* properties); the property *agent modifier* links a passive agent description to an *animal characteristic*.; and the property *agent* indicates that the agent should be some species. The classes *species* and *animal characteristic* both belong to the domain ontology.

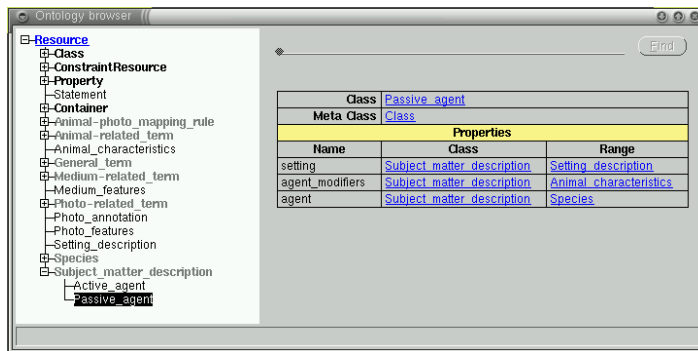


Figure 9.5: Snapshot of the tool's RDFS browser. Two properties of the subject matter description link to the animal domain ontology (agent → species; agent modifier → animal characteristic).

Although our mappings are simple, we expect other mappings will be more complex, especially in cases where there is no simple one-to-one mapping. Research on ontology mapping and merging (McGuinness et al. 2000) is required.

9.1.4 Annotating photographs using our multimedia information analysis tool

The tool we developed reads an RDFS file containing ontology specifications. The RDFS produced by Protégé conforms to the W3C standard, (Brickley and Guha 2000) except for the range definition of properties. RDFS only allows a single type for a range constraint; this is too limited for Protégé. We handled this inconsistency by simply allowing multiple range constraints. The RDFS specification document indicates that we should specify a superclass

for multiple range classes, but this syntactic solution is not desirable from an ontological-engineering perspective because it breaks modular design: the new class is conceptually not part of the original concept hierarchy or hierarchies but belongs to the RDFS annotation template.

From the RDFS specifications, the tool generates a user interface for annotating photos. Figure 9.6 shows a snapshot of the annotation interface. There are three tabs for the three groups of features: subject matter, photograph, and medium. The figure shows the passive agent template for a subject matter description. In the example, the annotation says the agent is a chimpanzee with two modifiers, namely “life stage = young” and “posture = scratching-the-head.”

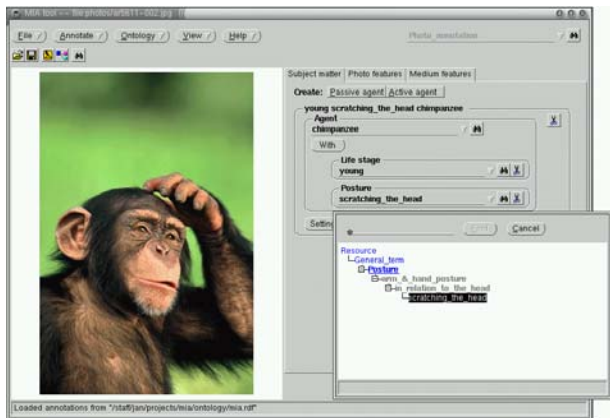


Figure 9.6: A snapshot of the annotation interface. The user has selected “chimpanzee” as the animal species. Two agent modifiers are defined: “life stage” and “posture.” At the lower right, the part of the domain ontology is shown containing fillers for the “posture” feature.

The user can enter terms in two ways. He or she can type in a term and use a completion mechanism to see whether it matches a term in the ontology (the typed text becomes bold when matched). Alternatively, the user can click on the magnifier icon to browse the relevant part of the ontology to select a term. The pop-up window at the lower right shows this for the agent modifier *posture*. The window lets the user select a term from the hierarchy under the class *posture*. The hierarchy of terms comes from WordNet and IconClass. The tool also supports administrative tasks such as reading in new photos, storing photo annotations, and loading existing photo annotations.

The user interface generator can be defined almost independently of the ontology. The generator reads the RDFS representing an annotation schema. For each property of this

schema that represents another compound schema, it generates a *tab* or sentence item. If a property refers to an ontological term, it generates an item providing completion, search, and hierarchical browsing. Finally, for properties defined as RDF-literal, it generates a simple text item. Schemas entered as a “sentence” require an additional declaration to tell the generator the order in which the properties appear in the sentence.

Such a generic interface also has some drawbacks. The RDFS’s structure maps directly to the interface, but grouping based on ontological motivations is not necessarily the best choice for the UI, and properties in RDF provide no ordering information. Also, the interface uses abstract terms such as *agent* that might not be intuitive for users. For practical applications, the user must provide additional grouping, ordering and label information to the UI generator.

9.1.5 Discussion

This study has only scratched the surface of the problems encountered when trying out a content-oriented approach to annotate and search for photos.

9.1.5.1 What do ontologies offer over keywords?

In photo collections indexed with keywords, a small subset of the controlled keyword set is associated with an image. The keywords themselves are unrelated atoms. If we consider the terms of the ontology to be our controlled keyword list, using an ontology and a structured description based on this ontology changes the annotation and querying process in a number of ways:

- It guides the annotation process using restrictions and default information.
- It makes the relation between property values and agents explicit, telling which property value is connected using which property to which element of the subject matter or the photo itself. Consider “chimpanzee under large tree.” Reduced to keywords, “large” can refer to the chimpanzee, the tree, or even the photo.
- The ontology provides relations between the terms; in our case, default information (“orangutans live in Indonesia”) and inheritance. Inheritance provides a controlled means to widen or constrain a query.

In our view, there is enough evidence to warrant further research, but there is a long way to go to actually prove that ontology-based search is better (in some respects) than keyword search.

9.1.5.2 Architecture feasibility

The architecture presented in this article provides a Semantic Web standard-conforming framework for representing ontologies and photo annotations. The tool makes two limiting assumptions. First, it assumes the annotated object is an image. Second, it assumes an

RDF Schema defining a set of classes with properties that represent an annotation. Each class either has atomic attributes or has attributes for which a sentence schema is defined.

Currently, the tools represent the RDF data as triples in the Prolog database. The triple database includes three modules: one for the ontological data, one for the annotations, and one for the queries. Neither this representation nor the used RDF query algorithm will scale to large databases (greater than 100,000 triples). Scalability of RDF storage and query systems is an active research topic (see the RDF mailing list, rdf-interest@w3c.org).

9.1.5.3 Guidelines for using Web standards

During this study, we used RDFS, RDF, XML, and XML-DTD. We started using the latter to define the photo annotation structure until we realised we were using a language intended for defining syntactical structure for the specification of semantical structure. At that point, we decided to treat a photo annotation as a semantic unit described in RDF and to define its structure as an RDFS.

Likewise, an ontology can be expressed in RDF (the OpenDirectory project¹), but this approach loses the frame semantics of RDFS. Ontological class definitions typically require constrained properties and inheritance. This means that RDFS is a much more suitable formalism than plain RDF. If one limits the formalism to pure RDF, the ontology itself is machine-readable, but not machine-understandable.

We extended RDFS by refining *class* to add additional knowledge to the model, such as default values for properties. OIL (Fensel et al. 2000) uses the same mechanism to extend the semantics of RDFS. Unfortunately, these extensions are not generally machine-understandable. RDFS can only grow by the acceptance of OIL and OIL-like extensions as additional standards.²

9.1.5.4 The link with Dublin Core

In this article, we focused on annotations about the content of a photograph. In terms of the Dublin Core element set, (Dublin Core Metadata Initiative 1999) our structured subject matter description is an elaborate refinement of the *subject* element. For some of the photograph features and medium features, the link with Dublin Core is more straightforward. Table 9.2 shows the mapping between features in our annotation ontology and Dublin Core elements. Assuming an official RDFS specification of Dublin Core becomes available, we can redefine these features as subproperties of the corresponding Dublin Core elements. (Currently, there is only an unofficial one in Appendix B of the RDFS document mainly intended as an example of the use of RDFS.) In this case study, the Dublin Core *type* element will always have the value *image* (following the DCMI type vocabulary). In this way, we can ensure that the resulting photo annotations comply with Dublin Core's *dumb-down principle*, which states

¹<http://www.dmoz.org/>

²This role is played by OWL (Dean et al. 2004), the Semantic Web language that is in part based on DAML and OIL.

that refinements of the element set are allowed provided it is still possible to access the annotation through the basic element set.

Annotation ontology	Feature type	Dublin Core element (qualifier)
Copyright holder	Photo feature	Rights
Photographer	Photo feature	Creator
Exact time	Photo feature	Coverage (temporal)
Exact location	Photo feature	Coverage (spatial)
Format	Medium feature	Format
Resolution	Medium feature	Format
Size	Medium feature	Format (extent)
Photograph colour	medium feature	Format

Table 9.2: Correspondence between features of the annotation ontology and Dublin Core element.

9.1.5.5 Support tools

Support tools are crucial for making the architecture sketched in figure 9.1 work. The Protégé-2000 tool proved useful for our study. The RDFS generated by Protégé conformed to the W3C standard, with the exception of range constraints. The main problems we had with Protégé was that it does not support multiple ontologies with import relations and it was incapable of loading large ontologies such as WordNet. Lack of modularity clutters the ontology definitions. Once multiple ontologies can be handled, it is likely that tool requirements will come up with respect to ontology-mapping mechanisms (for example, defining the link between the subject matter description and the domain ontology).

9.1.5.6 Preprocessing existing annotations

Most photos in existing collections are annotated. This was also true for the photos we used in our study. The nature of the annotation varied considerably—one CD contained free-text annotations and another used keywords. Depending on the amount of useful information in the existing annotation, it might be worthwhile to consider the construction of a preprocessor to generate a (partial) semantic annotation from the existing annotation. Natural language analysis techniques are likely to be required for such preprocessing.

Acknowledgments

The Dutch government's ICES-KIS project "Multimedia Information Analysis" supported this work. We also thank Audrey Tam for the discussion about combining structured annotations with ontologies and Hans Akkermans, Michel Klein, Lloyd Rutledge, and Marcel Worring for other beneficial discussions. We are also grateful to the anonymous reviewers for their detailed and helpful comments.

9.2 Supporting Semantic Image Annotation and Search

*Jan Wielemaker, Guus Schreiber and Bob Wielinga
Frontiers in Artificial Intelligence and Applications, 2003*

Abstract In this article we discuss an application scenario for semantic annotation and search in a collection of art images. This application shows that background knowledge in the form of ontologies can be used to support indexing and search in image collections. The underlying ontologies are represented in RDF Schema and are based on existing data standards and knowledge corpora, such as the VRA Core Categories 3.0, the Art and Architecture Thesaurus and WordNet. This work is intended to provide a “proof of concept” for indexing and search in a Semantic Web.

9.2.1 Introduction

In this article we are exploring the possibilities of using background knowledge for indexing and querying an image collection. Many of such collections currently exist and users are increasingly faced with problems of finding a suitable (set of) image(s) for a particular purpose. Each collection usually has its own (semi-)structured indexing scheme that typically supports a keyword-type search. However, finding the right image is often still problematic.

In this work we investigate an alternative approach. We explore the question whether ontologies can support the indexing and querying process in any significant way. This work should be seen as a “proof of concept”. At the moment, the use of explicit background knowledge is often-mentioned as a way out of the search problems such as currently arise on the Internet and in multimedia databases. This new research area is part of the “Semantic Web” (Berners-Lee 1999). Emerging web standards such as RDF (Lassila and Swick 1999) are providing a syntactic format for explicating background knowledge of information resources. However, the added value of such semantic annotations still has to be proven. Within the scope of this work we have not done any detailed evaluation studies of the approach presented. Our sole aim is to see whether we can find sufficient evidence that this is a plausible approach worth further investigation.

9.2.2 Approach

Figure 9.7 shows the general architecture we used within this study. All ontologies were represented in RDFS Schema (Brickley and Guha 2000). The resulting RDF Schema files are read into the tool with help of the SWI-Prolog RDF parser (chapter 3, Wielemaker et al. 2003b, Parsia 2001). The annotation tool subsequently generates a user interface for annotation and search based on the RDF Schema specification. The tool supports loading images and image collections, creating annotations, storing annotations in a RDF file, and two types of image search facilities.

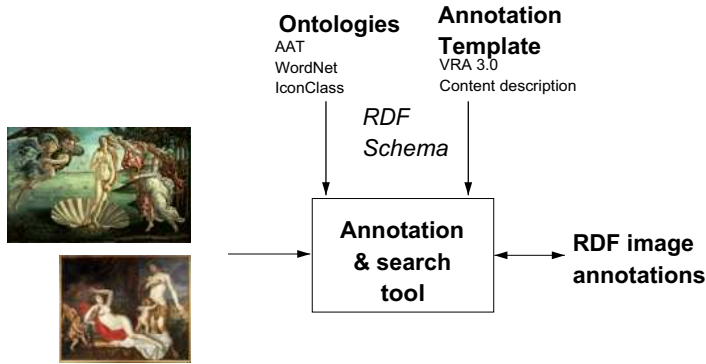


Figure 9.7: Overview of the approach in this study. The RDF Schema specifications of the ontologies and of the annotation template are parsed by the RDF/RDFS parser into the tool. The tool generates an annotation and search interface from these specifications. This interface is used to annotate and query images. The annotations are stored in an RDF file.

The architecture shown in figure 9.7 is in the same spirit as the one described by Lafon and Bos 2000. The main difference lies in the fact that we place more emphasis on the nature of the ontologies. In the rest of this paper the architecture depicted in figure 9.7 is described in more detail. The background knowledge is discussed in section 9.2.3. The annotation and query aspects of the tool are discussed in section 9.2.4 in the form of an application scenario. section 9.2.5 discusses the experiences gained.

This work is a sequel to earlier work on semantic annotation and search of a collection of photographs of apes (section 9.1, Schreiber et al. 2001). As the sample domain for the present work we used 200 images of paintings.

9.2.3 Background Knowledge

9.2.3.1 Ontologies

For this study we used three thesauri, which are relevant for the art-image domain:

1. The Art and Architecture Thesaurus (AAT, Peterson 1994) is a large thesaurus containing some 120,000 terms relevant for the art domain.
2. WordNet (Miller 1995) is a large lexical database. WordNet concepts (“synsets”) are typically used to describe the content of the image, e.g., “woman sitting on bed”
3. IconClass (van der Waal 1985; van den Berg 1995) is an iconographic classification system, providing a hierarchically organised set of concepts for describing the content of visual resources.

AAT, WordNet and IconClass were translated into the RDF Schema notation. In a prior publication (Wielinga et al. 2001) one can find a discussion on issues arising when representing AAT in RDF Schema.

These sources share a significant number of terms. This could be used to design a new ontology by *merging* them. In the context of the Semantic Web however, it is important to *re-use* existing work rather than modifying it. Therefore we have added equivalence relations between terms appearing in multiple ontologies that refer to the same entity in the world. For example, the IconClass concept **king** is linked to the AAT concept **kings (people)**, making all IconClass subterms of **king** valid subterms of AAT **kings (people)** and visa-versa.

9.2.3.2 Annotation template

For annotation purposes the tool provides the user with an annotation template derived from the VRA 3.0 Core Categories (Visual Resources Association Standards Committee 2000). The VRA template provides a specialisation of the Dublin Core set of metadata elements, tailored to the needs of art images. The VRA Core Categories follow the “dumb-down” principle (i.e., a tool can interpret the VRA data elements as Dublin Core data elements).

The 17 VRA data elements (i.e., properties of an art image) were for visualisation purposes grouped into three sets:

1. *Production-related descriptors* such as creator (“maker”), style/period, technique and, culture
2. *Physical descriptors* such as measurements and colour
3. *Administrative descriptors* such as collection ID, rights and current location

In figure 9.8 one can see a tab with the set of production-related descriptors. The other VRA descriptors can be found on two other tabs.

In addition, we needed to provide ways for describing the subject matter of the painting. VRA provides a subject element, but we were interested in providing more structured content descriptions. For this purpose we used a simple “sentence structure” which was developed for a previous experiment (section 9.1, Schreiber et al. 2001):

<agent> <action> <object> <setting>

Each content-description sentence should contain at least one agent or object. Agents and objects may have attributes (“modifiers”), such as colour, cardinality and position. The “setting” can be described with relative time (e.g., “sunset”) and relative place (e.g., “a forest”). In addition, the scene as a whole can be characterised. The application scenario in the next section gives an example of the use of this template. Multiple sentences can be used to describe a single scene.

9.2.3.3 Linking the annotation template to the ontologies

Where possible, a slot in the annotation template is bound to one or more subtrees of the ontologies. For example, the VRA slot *style/period* is bound to two subtrees in AAT containing the appropriate style and period concepts (see also figure 9.9). Four VRA data elements are currently linked to parts of AAT: technique, style/period, culture and material. Most parts of the subject-matter description are also linked to subtrees of the ontologies. Here, heavy use is made of WordNet and IconClass. The latter is in particular useful for describing scenes as a whole.

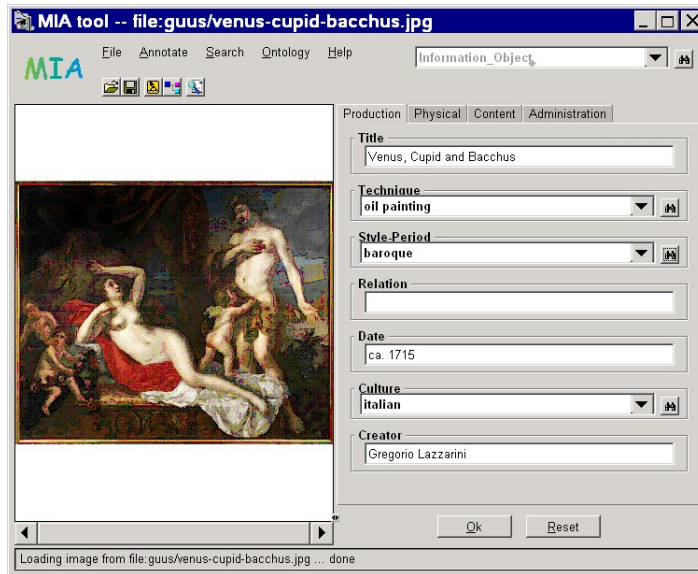


Figure 9.8: Snapshot of a semantic annotation and search tool for art images. The figure shows a fragment of the annotation window showing one tab with VRA data elements for describing the image, here the production-related descriptors. The slots associated with a “binoculars” button are linked to one or more subparts of the underlying ontologies, which provide the concepts for this part of the annotation.

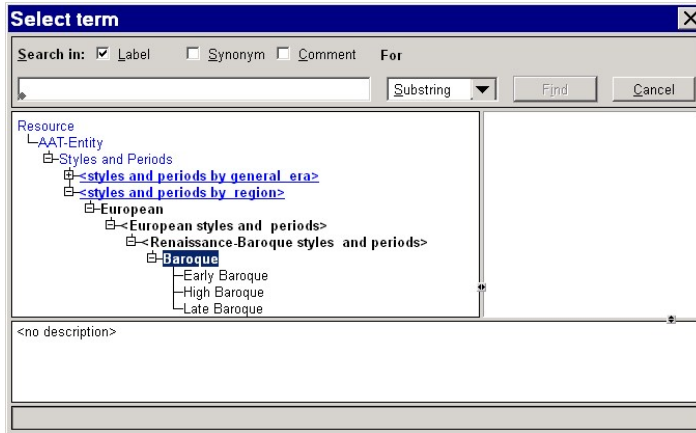


Figure 9.9: Browser window for values of style/period. The concept **baroque** has been selected as a value for this slot. The top-level concepts of the AAT subtrees from which we can select a value for style/period are shown with an underlined bold font (i.e., **<styles and periods by general era>** and **<styles and periods by region>**). The browser shows that there are three more specific concepts that could have been used (e.g., high baroque)

9.2.4 An Application Scenario

9.2.4.1 Annotating an image

Figure 9.8 shows a screen shot of the annotation interface. In this scenario the user is annotating an image representing a baroque Italian painting, depicting a mythological scene featuring Venus, Cupid and Bacchus. The figure shows the tab for production-related data elements. The three elements with a “binoculars” icon are linked to subtrees in the ontologies, in this case AAT. For example, if we would click on the “binoculars” for style/period the window shown in figure 9.9 would pop up, showing the place in the hierarchy of the concept **baroque**. We see that it is a concept from AAT. The top-level concepts of the AAT subtrees from which we can select a value for style/period are shown with an underlined bold font (i.e., **<styles and periods by general era>** and **<styles and periods by region>**). The ontology makes it easier for people to select the correct concept. For example, seeing that **baroque** contains three specialisations the user might want to use one of these terms, e.g., **high baroque**.

The user interface provides some support for finding the right concept. For example, the user can type in a few characters of a term and then invoke a completion mechanism (by typing a space). This will provide a popup list of concepts matching the input string.

In the browser window, more advanced concept search options can be selected, including substrings and use of synonyms.

The domain knowledge can be extended to cover more slots. For example, the creator slot could take values from the Getty thesaurus ULAN (Union List of Artist Names, Getty 2000).³

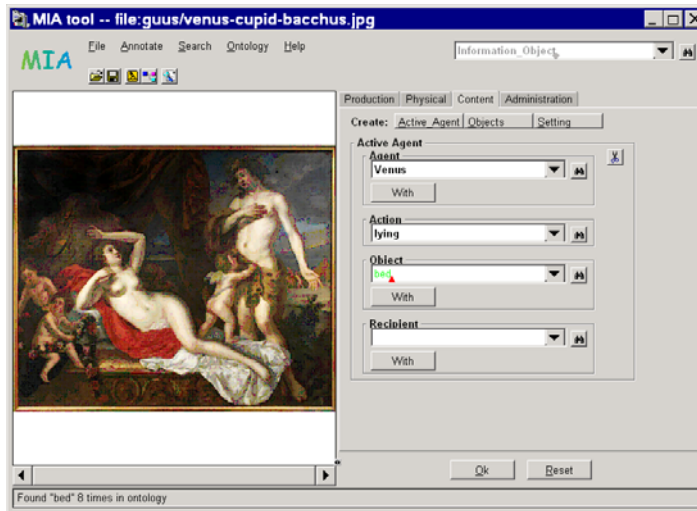


Figure 9.10: Description of the subject matter. Here, the content is described with the sentence “Venus (agent) lying (action) bed (object)”. Multiple content descriptions can be entered for an image

For annotation purposes the ontologies serve two purposes. Firstly, the user is immediately provided with the right context for finding an adequate index term. This ensures quicker and more precise indexing. Also, the hierarchical presentation helps to disambiguate terms. An example of this is shown in figure 9.10. Here, the user is describing the subject matter of the painting. Suppose she wants to say that “Venus is lying on a bed”. The template on the right-hand side implements the content template as described in section 9.2.3.2, in this case:

Agent:	Venus
Action:	lying
Object:	bed

³Including ULAN exceeded the scalability of our prototype. ClioPatria, as described in chapter 10 can easily deal with this scale of background information.

When the user types in the term “bed” as the object in a content-description template, the tool will indicate that this an ambiguous term. In the user interface the term itself gets a green colour to indicate this and the status bar near the bottom shows the number of hits in the ontologies. If one clicks on the binocular button, the tool will provide the user with a choice of concepts in AAT and WordNet that are associated with this term (piece of furniture, land depression, etc.). Figure 9.11 shows two of the concepts associated with “bed”. From the placement of the terms in the respective hierarchies, it is usually immediately clear to the indexer which meaning of the term is the intended one.

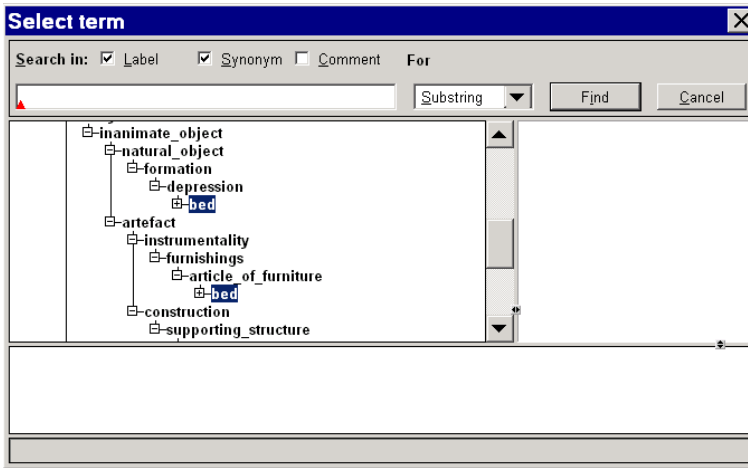


Figure 9.11: Disambiguating the term “bed”. This window shows two of the concepts associated with `bed`. From the placement of the terms in the respective hierarchies, it is usually immediately clear to the indexer which meaning of the term is the intended one.

Term disambiguation is a frequent occurrence in this type of application. For example, the term “Venus” may be associated with the Roman deity or with the planet.⁴

9.2.4.2 Searching for an image

The tool provides two types of semantic search. With the first search option the user can search for concepts at a random place in the image annotation. Figure 9.12 shows an example of this. Suppose the user wants to search for images associated with the concept **Aphrodite**. Because the ontologies contain an equivalence relation between Venus (as a Roman deity,

⁴In the “Venus lying bed” example, this disambiguation of “Venus” is not required, because the planet is not considered a legal value for an “agent” (this is debatable, by the way). However, general search for “Venus” (see the next subsection) will require disambiguation

not the planet nor the tennis player) and Aphrodite, the search tool is able to retrieve images for which there is no syntactic match. For example, if we would look at the annotation of the first hit in the right-hand part of figure 9.12, we would find “Venus” in the title (“Birth of Venus” by Botticelli) and in the subject-matter description (**Venus (a Roman deity) standing seashell**). The word “Venus” in the title can only be used for syntactic marches (we do not have an ontology for titles), but the concept in the content description can be used for semantic matches, thus satisfying the “Aphrodite” query.

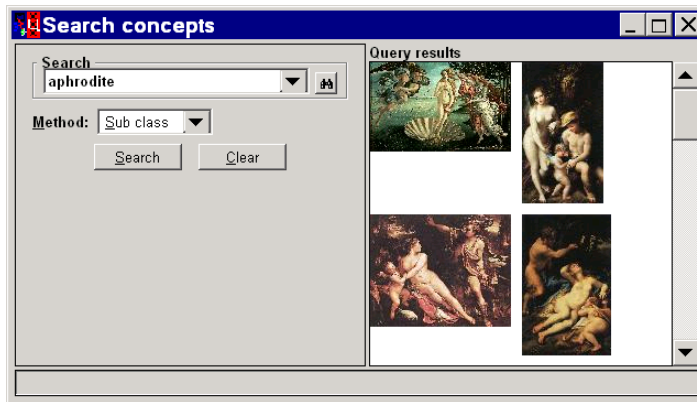


Figure 9.12: Example of concept search. The query “Aphrodite” will retrieve all images for which we can derive a semantic match with the concept **Aphrodite**. This includes all images annotated with the concept **Venus** (as a Roman deity). Only a small fragment of the search results is depicted

General concept search retrieves images which match the query in some part of the annotation. The second search option allows the user to exploit the annotation template for search proposes. An example of this is shown in figure 9.13. Here, the user is searching for images in which the slot culture matches **Netherlandish**. This query retrieves all images with a semantic match for this slot. This includes images of **Dutch** and **Flemish** paintings, as these are sub concepts of **Netherlandish**.

9.2.5 Discussion

This article gives some indication on how a Semantic Web for images might work. Semantic annotation allows us to make use of concept search instead of pure syntactic search. It paves also the way for more advanced search strategies. For example, users may be specialising or generalising a query with the help of the concept hierarchy when too many or too few hits are found.

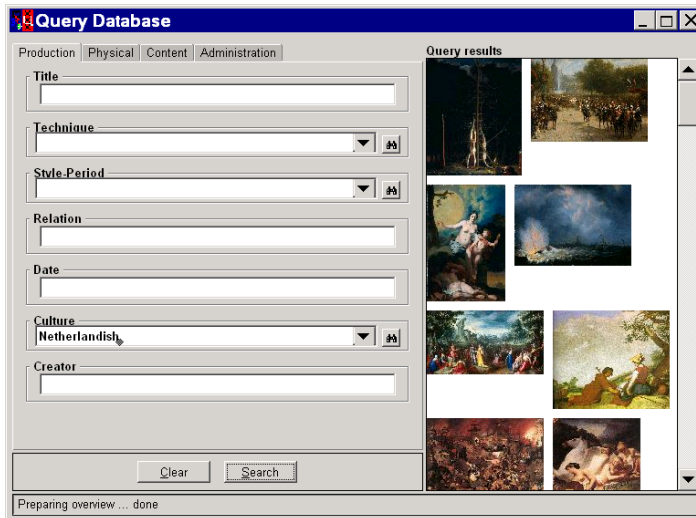


Figure 9.13: Search using the annotation template. The query “Netherlandish” for the slot **culture** retrieve all images with a semantic match for this slot. This includes images of **Dutch** and **Flemish** paintings, as these are sub concepts of **Netherlandish**

In a previous study on a collection of ape photographs (section 9, Schreiber et al. 2001) we did some qualitative analysis on the added value with respect to keyword search. The provisional conclusion was that for some queries (e.g., “ape”) keyword search does reasonably well, but for other slightly different queries (e.g., “great ape”) the results are suddenly poor. This is exactly where semantic annotation could help.

Although our approach relies to some extent on manual annotation (as we did in this study), it should be possible to generate partial semantic annotations from existing annotations (which vary from free text to structured database entries). This would speed up the annotation process considerably. We are currently starting a follow-up study with a collection of 4,000 images of paintings for which we want to extract data such as title, creator, year and location from the existing annotation.

Our experiences with RDF Schema were generally positive. We made heavy use of the metamodeling facilities of RDF Schema (which allows one to treat classes as instances of other classes) for defining and manipulating the meta models of the different thesauri. In our experience this feature is in particular needed in cases where one has to work with existing representations of large ontologies. This is a typical feature of a Semantic Web: you have to work with existing ontologies to get anywhere (we’re not going to rebuild AAT), even if one disagrees with some of the design principles of the ontology.

For our purposes RDF Schema has some limitations in expressivity. We especially needed a notion of property cardinality and of equivalence between resources (classes, instances, properties). For this reason we plan to move at some near point in the future to OWL, the Web Ontology Language currently under development at w3c (W3C Web Ontology Working Group 2001).

A major challenge from a knowledge-engineering perspective is to provide principled methods for adding ontology “glue”. The equivalence relations mentioned in section 9.2.3.1 are one example of this. In many cases existing ontologies can be augmented with additional background knowledge. For example, if we know the creator is André Derain (an artist in ULAN), we could suggest that the style of the painting is “Fauve” (a style in AAT). But for this to work, we need to add knowledge linking ULAN and AAT. In this area much more work needs to be done to see what is feasible and what is not,

9.3 Lessons learned

This chapter describes an application developed before the material described in part I of this thesis (except for chapter 5) was available. The overall aim of this research was to test the hypothesis that using background knowledge in the form of ontologies improves the annotation and search process for multi-media objects (images) compared to using simple keywords. We used software prototype development to explore architectural opportunities provided by the emerging Semantic Web. In particular, based on our experience from Shelley and the CommonKADS Workbench (see chapter 1), using the RDF triple model as the core storage formalism for applications was expected to be a good design. Exploiting the dynamic nature of XPCE/Prolog and in particular its automatic layout capabilities, we generated the user interface as much as possible from the RDF-Schema for an annotation. The schema provides the attributes, their hierarchical organisation (modelled as `rdfs:subPropertyOf` and mapped to *tabs*) and a value set (`rdfs:range` of a property) that can be used for menus (small sets) or autocompletion (large sets). The annotation ontology thus provided all information for the GUI, except for ordering of tabs and properties. Intuitive ordering was defined with additional (Prolog) rules.

In our first experiment (section 9.1) we used small hand crafted ontologies and could afford to store the RDF simply as a dynamic predicate `rdf(Subject, Predicate, Object)`. The second experiment (section 9.2) used much larger triple sets from existing background knowledge which also introduced the problems of integrating these ontologies into a coherent metadata vocabulary, which was resolved using `rdfs:subPropertyOf` mapping rules. The scalability was improved by introducing a formal API for modifying the triple set that managed additional indexing tables, still providing the core query API as `rdf/3`. (Parsia 2001).

Based on this experience and considering our future research plans which involved annotation using terms from large existing ontologies, we decided to invest in a new generation infrastructure to fulfil our requirements:

- *Scalability and expressivity*

We anticipated on the need to store up to 10 million triples in main memory on commodity hardware in the near future. As we intended to map different annotation vocabularies using `rdfs:subPropertyOf`, we required efficient support reasoning with these mapping relations. These two requirements triggered the development of RDF-DB (chapter 3, Wielemaker et al. 2003b).

- *Modular handling of ontologies*

When using externally provided large ontologies, it is of course desirable to keep these separated. Still, the developer of mapping relations and (small) additional ontologies wants an integrated overview of all ontologies. Protégé was not capable to deal with modular ontologies, nor with the scale of data we anticipated. This triggered the design of Triple20 (chapter 2, Wielemaker et al. 2005).

- *Reusable GUI framework*

Still planning for a local GUI implementation for the next generation of search a annotation tools, we assembled Triple20 from reusable components (see section 2.4.1).

Acknowledgments This work was supported by the ICES-KIS project “Multimedia Information Analysis” funded by the Dutch government. We gratefully acknowledge the work of Maurice de Mare on the annotation of the images.

