Ontology-Based Photo Annotation

A. Th. (Guus) Schreiber, Barbara Dubbeldam, Jan Wielemaker, and Bob Wielinga,
University of Amsterdam

While technology enables the storage and distribution of photographic images on an unprecedented scale, finding what you want can be like finding the proverbial "needle in a haystack." The authors describe their approach and the tool they developed to make annotating photos and searching for specific images more intelligent.

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*[1] and emerging standards (such as the resource description framework (RDF)[2]) 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. We also compare our approach's performance with two existing Web-based search engine options. The article concludes with observations regarding the standards and tools we used in this annotation study.

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 and create our test case, we obtained three CDs with collections of animal photo samples. The CDs contained about 3,000 photos, but we used a subset of approximately 100 photos of apes for our annotation study.

Figure 1 shows the general architecture used in our annotation study. We specified all ontologies in RDF Schema (RDFS)[2] using the Protégé-2000[3] 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[4] 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 1 is in the same spirit as the one Yves Lafon and Bert Bos described.[5] However, we place more emphasis on the nature of the ontologies, the subject matter description, and the explicit link to a domain ontology.

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 describing 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.

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,[6] 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 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.

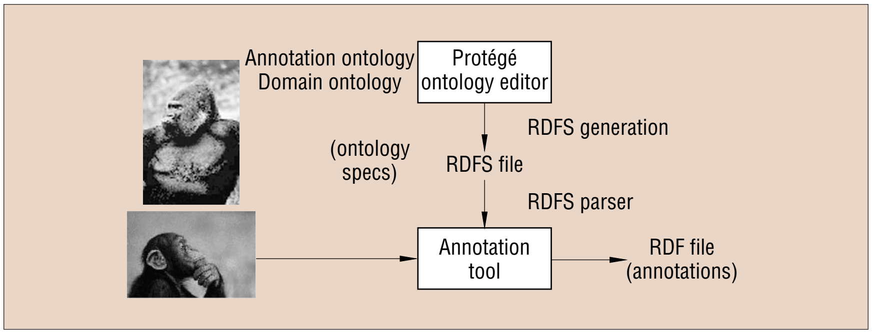


Figure 1. Overview of the approach used in this study. We used the Protégé ontology editor to construct ontologies and store them in RDFS format.

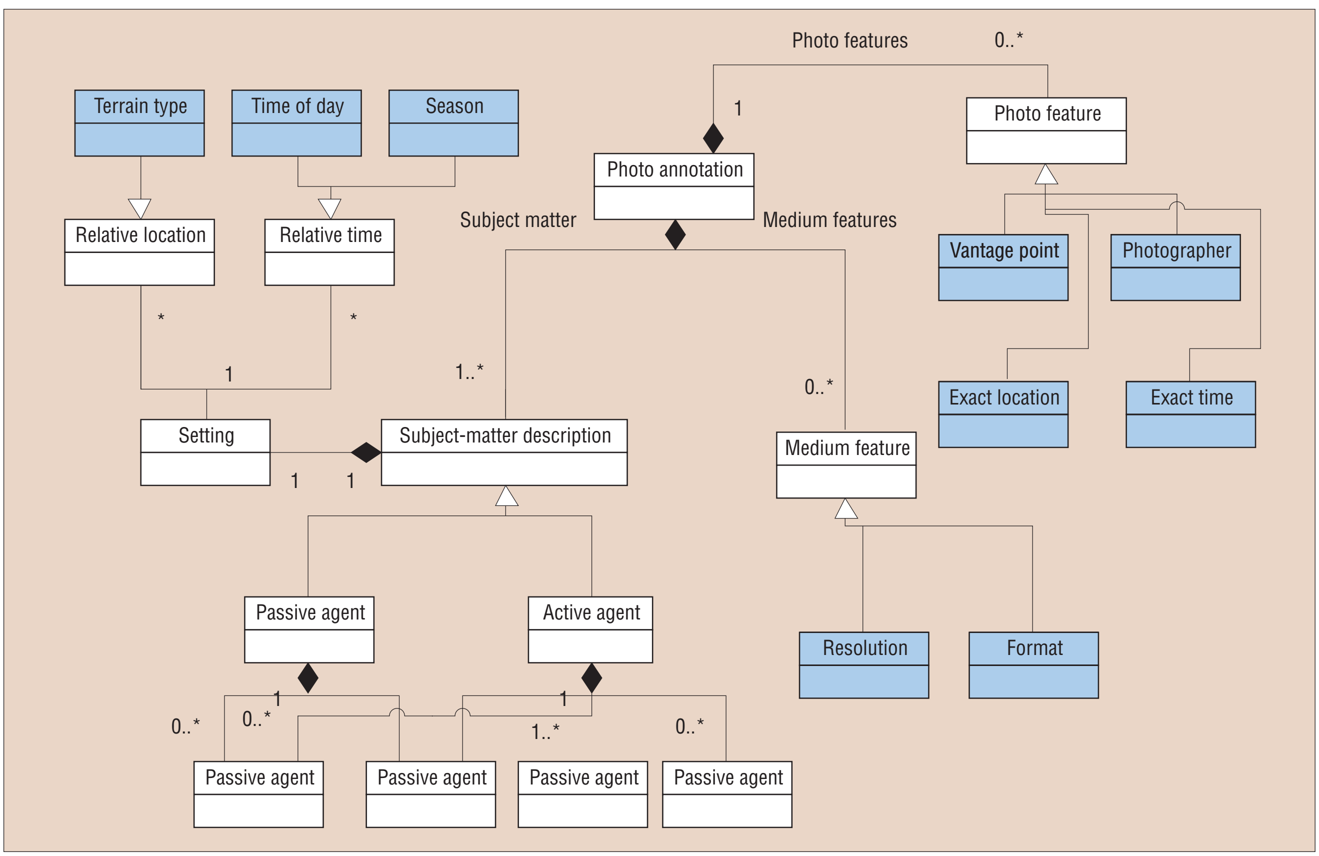


Figure 2. Structure of the photo annotation ontology represented as a UML class diagram.

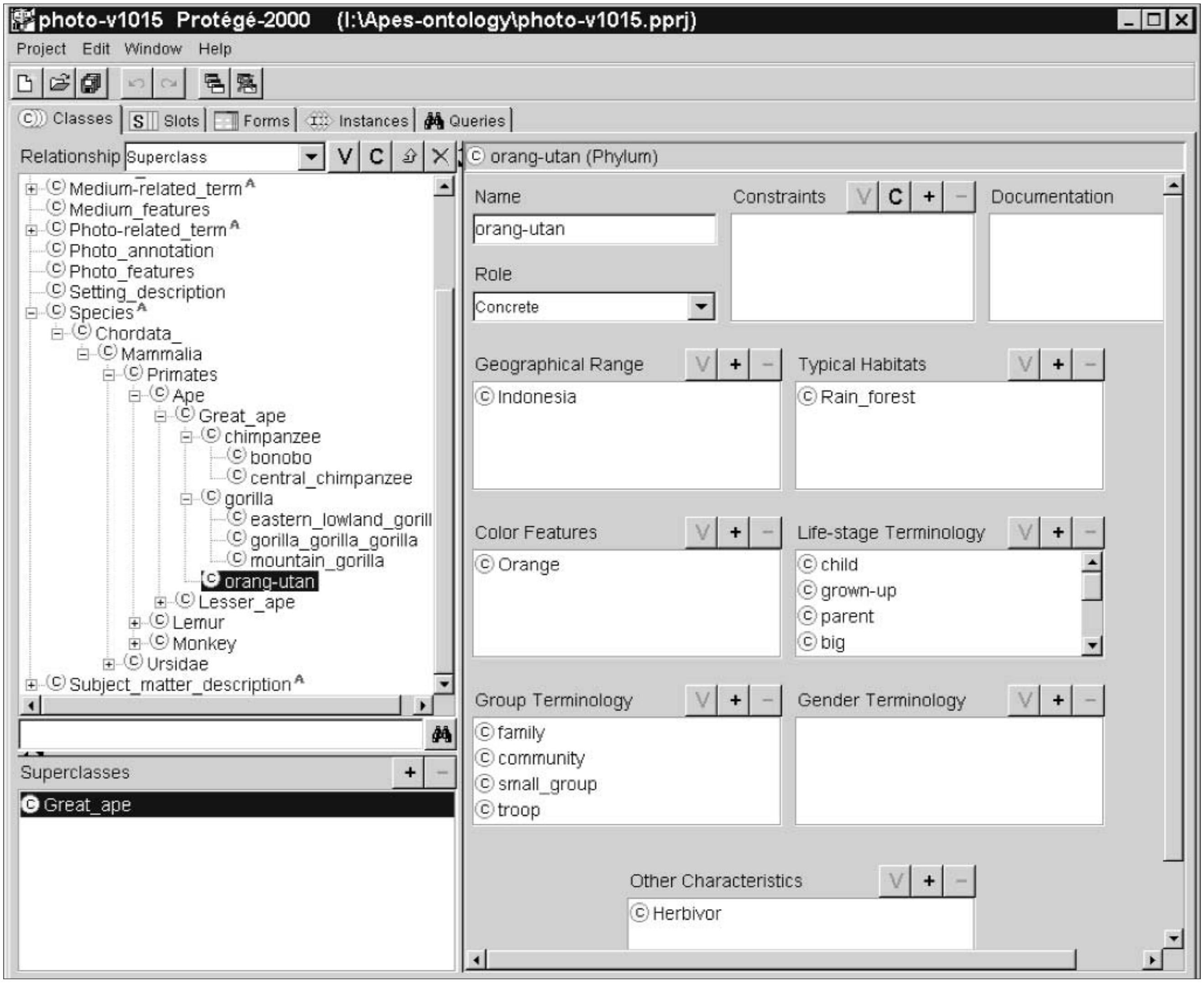


Figure 3. Snapshot of the Protégé-2000 ontology editor showing part of the domain ontology.

When constructing an ontology, we often needed to incorporate definitions already available in other corpora. For example, to define a *color* feature, we don't want to type in all the possible values for "color." A resource such as WordNet[7] 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.

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* Audrey Tam and Clement Leung described.[8] They propose a description template consisting of four elements:

1. An *agent*, for example, "an ape." An agent can have modifiers such as "color = orange."
2. An *action*, for example, "eating."
3. An *object*, for example, "a banana." Objects can also have modifiers (color = "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 see why it is useful to be able to link to existing corpora so we used parts of the WordNet vocabulary.

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 1 shows the properties we currently use.

A particular species represents a class that is an instance of metaclass phylum. This organization of the ontology lets us define

Table 1. Features of animal species defined in the domain ontology.

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."
Color features	Description of general colors ("orange") or color patterns ("striped").
Other characteristics	A catch-all category for characteristics such as "captive animal" and "domestic animal."

instance-type features of species—for example, that an orangutan has an "orange" color 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 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).

There is no way the domain ontology can be complete—it will not include everything a user might want to say about a photograph. Therefore, any annotation mechanism should also allow additional unstructured annotations (consequently requiring syntactic search methods to access them). We concentrated on what we thought were the main discriminating characteristics for our case study. Because this is a crucial choice, a full implementation requires a more detailed analysis and should incorporate cognitive science research on what kind of things people notice when they look at photographs.[9]

Also, not all elements of the subject matter description may have a counterpart in the domain ontology. For example, for the active agent template, there are no domain ontology constructs for the object, because it is impossible to constrain this part of the annotation's value. It could be a banana or something completely different.

General terminology

Both the annotation ontology and the domain ontology use general terminology. Instead of defining this ourselves, we used parts of WordNet and ICONCLASS.[10] 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 color aspects, we want to use both general colors ("orange") as well as animal-specific color 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, ICONCLASS, and possibly many other sources. This means we need a structured approach for linking domain vocabularies.

Figure 4 shows a graphical overview of the ontologies and vocabularies using the UML package notation. The links represent UML dependencies: "<source> depends on <destination>." Figure 4 shows the conceptual distinction we would have liked to make but could not. We confronted two problems. First, the Protégé tool does not support ontology modularization—we can include an ontology, 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 Global Retrieval, Access and Information System for Property items project (see http://arttic.com/GRASP), a Corba-based ontology server directly links descriptions of art objects to elements of the AAT.

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 prior ontology to the phylum hierarchy in the later ontology. Figure 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

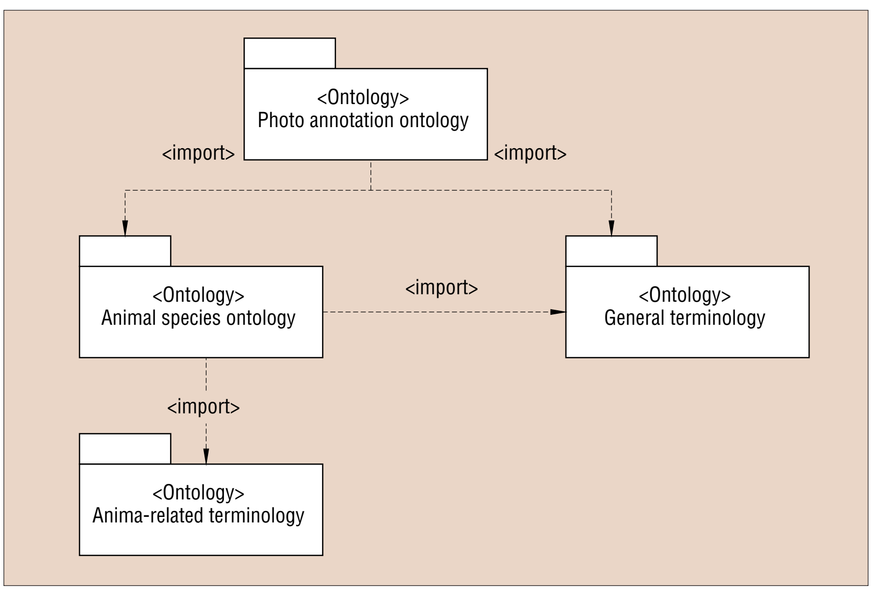


Figure 4. Import relations between the ontologies for animal photograph annotations using the UML package notation.

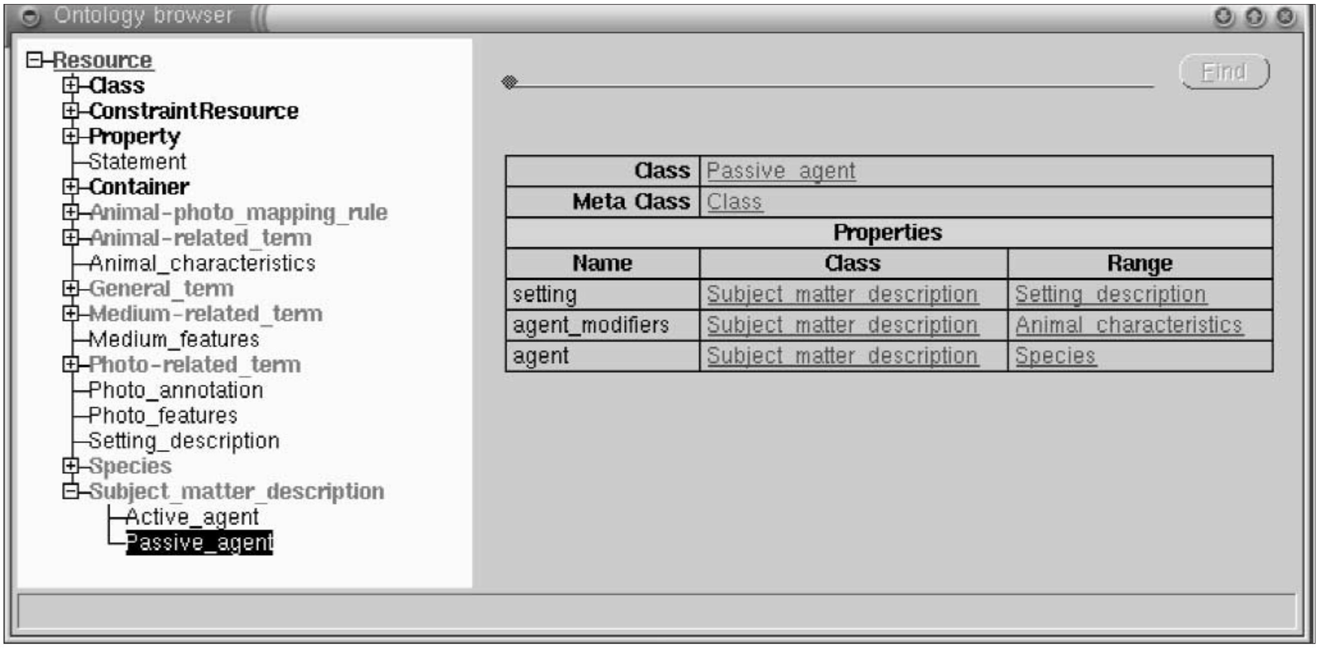


Figure 5. Snapshot of the tool's RDFS browser. Two properties of the subject matter description link to the animal domain ontology (agent → species; agent modifier → animal characteristic).

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.

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 is required.

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,[2] 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 ranges can be disjunctive. In our opinion, this limitation should be removed in the final RDFS standard.

From the RDFS specifications, the tool generates a user interface for annotating photos. Figure 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."

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 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* and might not be intuitive for users. For practical applications, the user must provide groupings, ordering and labels to the UI generator.

Querying photographs using our multimedia information analysis tool

We constructed a query tool to test the annotations. This tool uses the same interface as the annotation tool to create an RDF description of the target image. The tool searches the database for annotations that have all properties specified in the target description filled with values that are equal - to or specializations of the value in the target. The tool uses the subsumption hierarchy as represented in the ontology. A value is a

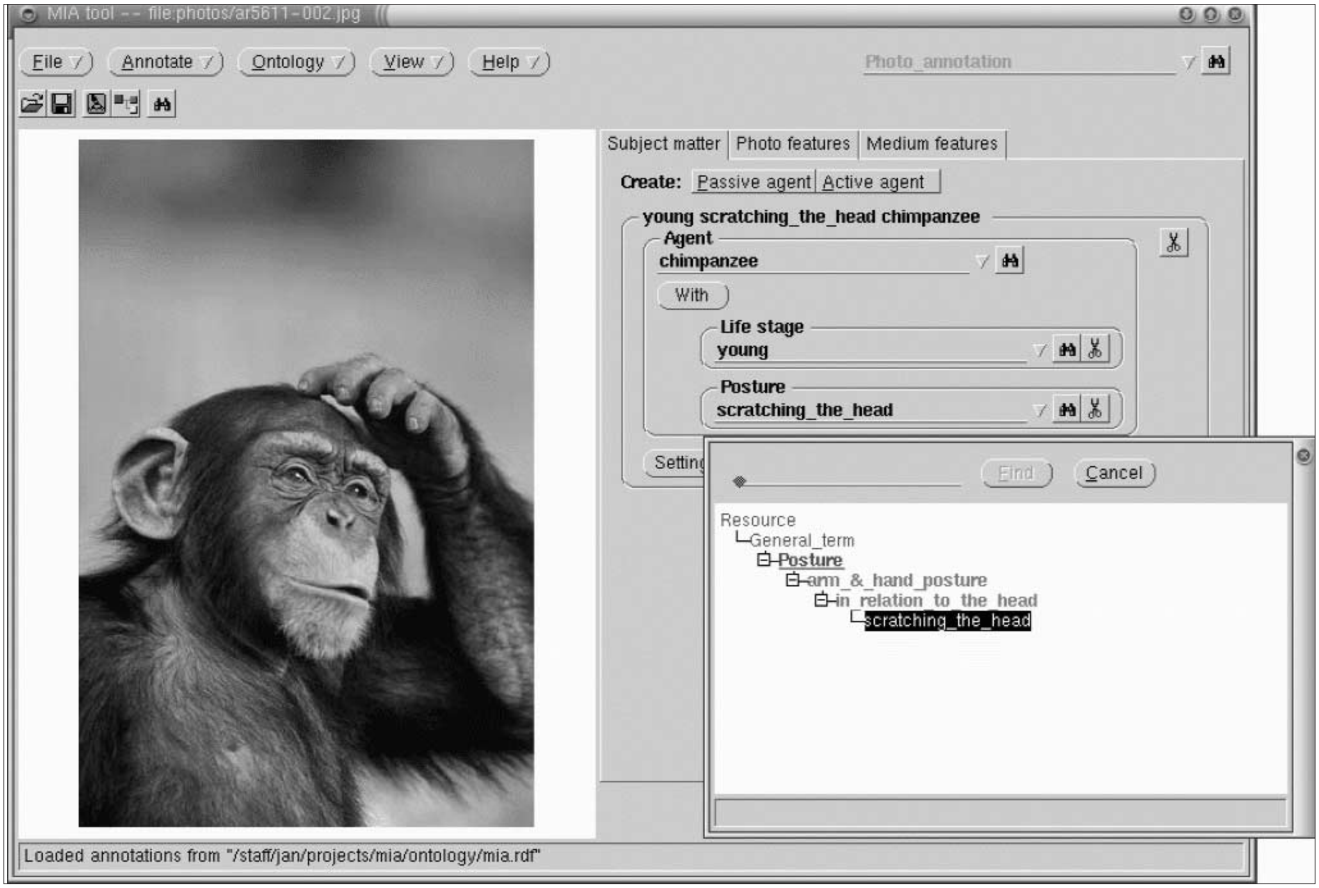


Figure 6. A snapshot of the annotation interface. The user has selected "chimpanzee" as the animal species. Two values for agent modifiers are defined: "color" and "posture." At the lower right, the part of the domain ontology is shown containing fillers for the "posture" feature.

specialization if the domain of the related property is an RDFS class, and the value in the annotation is a subclass of the value in the target description. For example, the posture property value "in relation to the head" will match more specific values such as "peeping between the fingers." Parts of a description (for example, the action or object in an active agent) can be left open and act as wildcards. So, it is possible to search for an eating ape without specifying what the ape is eating. Because one image can have multiple annotations, various queries search for a particular image. For example, an image can be indexed as an active agent "with mouth wide open" and as a passive agent "showing teeth." Currently, the images aren't ranked. Figure 7 shows the tool in action.

Our annotations contain *default information* (such as "an orangutan lives in Indonesia"), represented as attributes of the RDFS class orangutan. This type of knowledge requires interpretation outside the scope of plain RDFS, which allows for subclassing and refining class "class" before making "orangutan" and instance of this refined class but does not attach semantics to this. We anticipate using this type of knowledge for reasoning schemas such as "Because orangutans are normally orange, we may include them in a query for orange-colored animals."

Alternatively, "orange" could be a range restriction to the orangutan's color property. Such a description is only possible in more

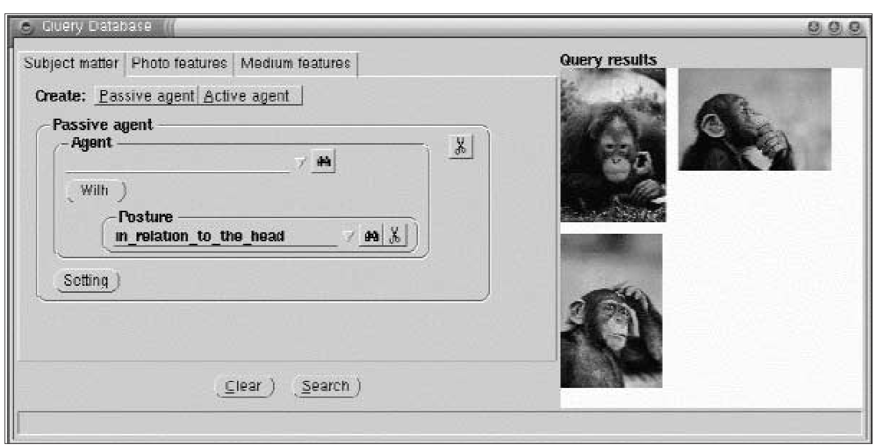


Figure 7. A snapshot of the annotation tool's query interface. The user specified a search for a photo with an animal doing something with its head. The user has not specified a particular species (*agent*), so all photos in the database are considered.

sophisticated languages such as OIL,[11] however, photo annotation and querying is not a very formal domain. For example, we want to deal with an "albino orangutan" individual simply by annotating the color as "white"—not by introducing all the knowledge representation complexities of exceptions. It is likely that querying photos benefits more from heuristic reasoning than from strict formal reasoning.

Default information such as "an orangutan normally has color orange" can include the orangutans in the query results for "an orange ape." Similarly, a query for "all orangutans" might include all "orange apes" to deal with omissions in the annotation process. Default information can suggest new queries using additional or alternative properties, while inheritance can widen or refine the scope guided by the hierarchy.

Evaluation

Although the application described in this article is still a prototype, we have attempted to gather some data on its performance. We used two methods of evaluation: qualitative comparison with other search engines available on the World Wide Web and usability analysis performed by a small set of users.

Qualitative comparison with other search engines

In the first part of the evaluation, we compared search results obtained with our tool with searches done with Alta Vista and a search engine connected to a number of photo archives.

If we consider simple queries such as "find images of a great ape," it is clear that our application will have 100 percent recall and precision, if an annotator with sufficient knowledge of great apes has done the annotation. The Alta Vista image finder searching the Web for "great + ape" finds 13 hits of which six are images of real apes. Clearly, recall is poor—there are at least thousands of images of great apes on the Web—precision around 50 percent is not very good either. The query "great ape" produces over 45,000 hits. It is difficult, if not impossible, to estimate the recall value for this search result because we have no idea of the number of relevant images on the Web. We estimated the precision of Alta Vista's retrieval process by inspecting the first 200 images presented by the search engine. Of these 200 images, only 64 were images of great apes—a precision of 32 percent. Comparing the results of a search engine such as Alta Vista with our tool could be considered unfair, because Alta Vista uses machine-generated indexes and we indexed our data by hand. However, the comparison shows that current search technology on the Web is either too specific (low recall, reasonable precision) or too general (high recall, low precision).

We also compared the performance of our tool with a search engine for a collection of photo archives, namely gettyone.com. Generally, humans index these photo archives using an archive-specific style of describing images in terms of a set of keywords. Sometimes, but not always, closed vocabularies are used in the indexing process. The gettyone search engine appears to use some knowledge to map the query keywords onto the keywords used in a particular archive. For example the query "ape" not only retrieves images indexed by the keyword "ape," but also finds images indexed by the terms "gorilla," "chimpanzee," and "orangutan." However, the term "great ape" is not part of its knowledge base, and thus no images resulted from this query. The query "ape" produced 521 hits out of an estimated 350,000 images in the database. Out of this set, 10 images are not of living apes (they include drawings, maps, or humans dressed up as apes). Of the remaining 511 images, 64 are images of other primates resulting in a precision of 86 percent. Confusing the concepts "ape" and "monkey" explains why these images were misclassified. Both keywords were included in the index for some of the images. Perhaps the person indexing the images was not a native English speaker because many languages do not make the distinction between "ape" and "monkey." Searching for the three species that belong to the great ape family, we retrieved 424 images, 19 less than in the relevant "ape" set. If we use more specific terms, the recall goes down to 95 percent, and precision goes up to 98 percent.

> Comparison shows that current search technology on the Web is either too specific (low recall, reasonable precision) or too general (high recall, low precision).

Clearly, image archives indexed by hand show a greater performance than machine-indexed databases. Compared to our tool, gettyone produces almost similar results for specific categories, but fails if generalized terms are used (compare "great ape").

We also investigated queries not just based on categories but also on additional features of the photograph (the modifiers in our structured annotations). We tested the query that produced the result in Figure 7 with the gettyone search engine. We used the image in Figure 6—which represents a chimpanzee scratching its head—to compare our tool to the gettyone engine. Using "chimpanzee" and "scratching" as keywords, we find the image depicted in Figure 6 as the only result. The fact that the animal is scratching its head rather than some other body part is difficult to represent in the keyword approach. The query "chimpanzee AND hand AND head" resulted in two hits, not including the one in Figure 6. Searching for great apes with their hands near their heads required keywords such as "head in hands," "animal head," and "animal hand." These keywords only came to mind after browsing a substantial part of the data. Recall was very unpredictable with these more specialized keywords. Many images showed apes with their hands in close relation to their head, but the terminology used in indexing varied widely. In a similar vein, we attempted a search for a great ape with its mouth open showing its teeth. The gettyone engine required terms like "yawning," "snarling," "teeth," "calling," and "danger" to retrieve images with apes with a mouth wide open showing teeth. In our application, the subsumption hierarchy of attributes provides a much more flexible indexing scheme.

Usability

Six users evaluated the system, using it to annotate a number of images and to retrieve other images. The users completed a questionnaire based on an in-house evaluation framework. Here, we focus on the evaluation of the system's conceptual aspects: the annotation process, the retrieval process, and the match between the users mental model and the ontology.

Annotation process evaluation revealed problems with the subject matter description's structure. In the case of images showing more than one agent, users wanted to represent the fact that multiple agents were present. This is currently not possible. The ontology contains a modifier *group*, but this is not always a natural way of expressing the cardinality of animals. The best solution would probably be to define a third annotation template for *interacting agents*, allowing different modifier values for different agents.

The use of default knowledge appeared to confuse users. They disagreed with some of the default knowledge and changed it. Also, some default knowledge was irrelevant for image description and played no part in the retrieval process. Certain aspects of the image couldn't be expressed—for example, a neutral facial expression was "lips pressed together." The relation between emotion and facial expressions was unclear, and users did not realize they could use multiple annotations for one image. This caused problems in the retrieval process—for example, there was no consensus whether an animal with an open mouth should be modeled as an active or passive agent. Unless the user enters two descriptions, the retrieval process can fail in the case of different views on the image. Also, the users had problems understanding the system's terminology. Annotation—and, to a lesser extent, retrieval—requires some domain knowledge. The users gave positive comments related to the efficiency and effectiveness of the system.

In summary, our annotation tool requires further development and refinement to be useful for large-scale deployment.

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

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. Still, there is a long way to go to actually prove that ontology-based search is better (in some respects) than keyword search.

Architecture feasibility

The architecture presented in this article provides a 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 RDFS using a set of classes with properties represents 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 very large databases (greater than 100,000 triples). Discussions on how to solve this problem are currently underway (see the RDF mailing list's archive: rdf-interest@w3c.org).

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 realized 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 project), 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 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.

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,[6] 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 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[2] 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 prin-*

Table 2. Correspondence between features of the annotation ontology and Dublin Core element.

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 color	medium feature	Format

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

Support tools

Support tools are crucial for making the architecture sketched in Figure 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 problem we had with Protégé was that it does not support multiple ontologies with import relations. This 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).

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-MIA project 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.

References

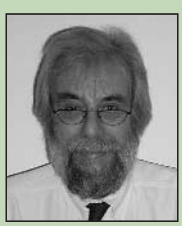
1. T. Berners-Lee, *Weaving the Web*, Orion Business, London, 1999.
2. D. Brickley and R.V. Guha, *Resource Description Framework (RDF) Schema Specification 1.0*, candidate recommendation, W3C Consortium, 27 Mar. 2000; www.w3.org/TR/2000/CR-rdf-schema-20000327 (current 7 May 2001).
3. N. Fridman Noy, R.W. Fergerson, and M.A. Musen, "The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility," *Knowledge Engineering and Knowledge Management: 12th Int'l Conf. EKAW 2000*, *Lecture Notes in Artificial Intelligence*, vol. 1937, Springer-Verlag, Berlin, 2000, pp. 17–32.
4. J. Wielemaker, *SWI-Prolog RDF Parser*, SWI, Univ. of Amsterdam, 2000; http://swi.psy.uva.nl/projects/SWI-Prolog/packages/sgml/rdf2pl.html (current 7 May 2001).
5. Y. Lafon and B. Bos, "Describing and Retrieving Photographs Using RDF and HTTP," W3C Consortium, 28 Sept. 2000, www.w3.org/TR/2000/NOTE-photo-rdf-20000928 (current 7 May 2001).
6. *Dublin Core Metadata Element Set Version 1.1: Reference Description*, Dublin Core Metadata Initiative, July 1999, http://dublincore.org/documents/1999/07/02/dces (current 7 May 2001).
7. G. Miller, "WordNet: A Lexical Database for English," *Comm. ACM*, vol. 38, no. 11, Nov. 1995.
8. A.M. Tam and C.H.C. Leung, "Structured Natural-Language Description for Semantic Content Retrieval," to be published in *J. American Soc. Information Science*, to be published Sept. 2001.
9. C. Jörgensen, "Indexing Images: Testing an Image Description Template," *Proc. ASUS 1996 Ann. Conf.*, Baltimore, Md., 21-24 Oct. 21-24 1996, pp. 209–213.
10. H. van der Waal, *ICONCLASS: An Inconographic Classification System*, Royal Dutch Academy of Sciences (KNAW), 1988.
11. D. Fensel et al., "OIL in a Nutshell," *Knowledge Engineering and Knowledge Management: 12th Int'l Conf. EKAW 2000, Lecture Notes in Artificial Intelligence*, vol. 1937, Springer-Verlag, Berlin, 2000, pp. 1–16.

The Authors

A. Th. (Guus) Schreiber is an associate professor of social science informatics at the University of Amsterdam. His research interests include knowledge engineering and knowledge-system development. Contact him at Social Science Informatics, Univ. of Amsterdam, Roetersstraat 15 NL-1018 WB Amsterdam, The Netherlands; schreiber@swi.psy.uva.nl

Barbara Dubbelddam is an MSc student in social science informatics at the University of Amsterdam. She works as a research assistant on the Multimedia Information Analysis project funded by the Dutch government's ICES program. Her focus is ontology development and evaluation issues. Contact her at Social Science Informatics, Univ. of Amsterdam Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands; dubbeldam@swi.psy.uva.nl.

Jan Wielemaker is a senior research scientist at SWI. His current focus includes tools for ontology management, annotation, and querying based on Web standards. Contact him at Social Science Informatics, Univ. of Amsterdam Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands; jan@swi.psy.uva.nl.



Bob Wielinga is a professor of social science informatics in the psychology department at the University of Amsterdam. His research focus has been the methodology of knowledge-based system design and knowledge acquisition. He received his PhD degree *cum laude* from the University of Amsterdam for a thesis in nuclear physics. Contact him at Social Science Informatics, Univ. of Amsterdam Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands; wielinga@swi.psy.uval.nl.