Ontology based document annotation: trends and open research problems

Oscar Corcho

Intelligent Software Components (iSOCO),
Pedro de Valdivia, 10 – 28006 Madrid, Spain
E-mail: ocorcho@isoco.com

Abstract: Metadata is used to describe documents and applications, improving information seeking and retrieval and its understanding and use. Metadata can be expressed in a wide variety of vocabularies and languages, and can be created and maintained with a variety of tools. Ontology based annotation refers to the process of creating metadata using ontologies as their vocabularies. We present similarities and differences with respect to other approaches for metadata creation, and describe languages and tools that can be used to implement these annotations.

Keywords: ontology; metadata; annotation.

Reference to this paper should be made as follows: Corcho, O. (2006) 'Ontology based document annotation: trends and open research problems', *Int. J. Metadata, Semantics and Ontologies*, Vol. 1, No. 1, pp.47–57.

Biographical notes: Oscar Corcho is working as a Marie Curie Fellow at the Information Management Group, University of Manchester. Previously, he has worked at iSOCO as a Research Manager and at the Ontological Engineering Group of Universidad Politécnica de Madrid (UPM). He graduated in Computer Science from UPM in 2000, and received the third Spanish award in computer science from the Spanish Government. He obtained his MSc in Software Engineering from UPM in 2001, and his PhD in Artificial Intelligence in 2004. His research activities include ontology languages and tools, the Semantic Web and the Semantic Grid.

1 Introduction

Metadata is usually defined as 'data about data', which aims at expressing the 'semantics' of information, hence improving information seeking, retrieval, understanding and use.

Metadata can be attached to a wide range of documents. These documents may be available electronically in the form of HTML, PDF, Latex, etc., in the Web or in our hard disks or on paper in a library, among others. Not only can metadata be applied to documents, but also to applications running in our computers or available in the web in the form of web services.

Metadata can be expressed in a wide range of languages (from natural to formal ones) and with a wide range of vocabularies (from simple ones, based on a set of agreed keywords, to complex ones, with agreed taxonomies and formal axioms). It may be available in different formats: electronically or even physically (written down in the margins of a textbook). And it can be created and maintained, using different types of tools (from text editors to metadata generation tools), either manually or automatically.

In this paper we will only deal with the management of metadata attached to electronic documents, expressed with formal languages and using ontologies as vocabularies. We will neither deal with the management of metadata for applications, nor with the creation of metadata based on other types of vocabularies. We will describe the advantages and disadvantages of using ontologies as the vocabularies on which the metadata is based (Section 2); we will describe some of the formal languages that can be used to express metadata (Section 3); and we will describe the tools currently available for ontology based document annotation (Section 4). Finally, we will present the conclusions to this paper and some open research problems in ontology based document annotation.

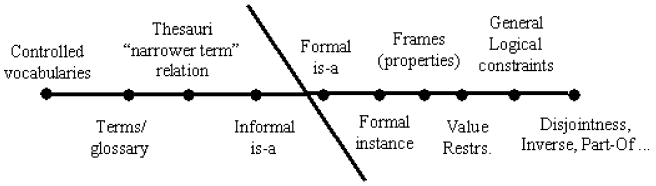
2 Ontologies as vocabularies for metadata annotation

Ontologies appeared first as the backbone of document metadata annotation in preSemantic Web applications like the SHOE project (Luke et al., 1997), the $(KA)^2$ initiative (Benjamins et al., 1999), and the Planet-Onto project (Domingue and Motta, 2000), among others. With the emergence of the Semantic Web, ontology based document annotation has been the focus of many projects and applications, since the availability of annotated content is one of the key challenges to overcome in order to make the

semantic web a reality (Benjamins et al., 2002). Among these projects and applications, we can cite the EU projects Esperonto (http://www.esperonto.net/) and Acemedia (http://www.acemedia.org/) the EU network of excellence SCHEMA (http://www.schema-ist.org/) or the US MindSwap (http://www.mindswap.org/) project all of them have in common the fact that they aim to provide tools or frameworks for annotating different types of content (HTML, databases, multimedia) and with different degrees of automation. A good URL where annotation projects and tools are compiled is the following: http://annotation.semanticweb.org/.

Ontologies are normally defined as "formal, explicit specifications of shared conceptualisations" (Studer et al., 1998). However, neither do all ontologies have the same degree of formality, nor do they include all the components that could be expressed with formal languages such as concept taxonomies, formal axioms, disjoint and exhaustive decompositions of concepts, etc. Given this fact, the ontology community usually distinguishes between lightweight and heavyweight ontologies (Studer et al., 1998). Lassila and McGuinness proposed the classification presented in Figure 1, which shows the different types of ontologies that can be defined in a continuous line from the very lightweight, even informal ontologies to heavyweight ontologies with a large number of formal axioms and constraints.

Figure 1 Lightweight vs. heavyweight ontologies and their relationship with Lassila and McGuinness categorisation



Source: Lassila and McGuinness (2001)

Many of these types of ontologies have been used for annotating metadata in documents and general web resources. Let us see some examples of widely used applications of metadata annotation:

Thesauri and controlled vocabularies. Terms from a thesaurus or from a controlled vocabulary can be used to annotate documents. Since these vocabularies are not completely formal (for instance, the relationships between the terms they include do not have a clear semantics), the annotations are normally pointers to those terms in the vocabulary, which can be used to improve search, for instance. Examples of such vocabularies are MeSH, (Medical Subject Headings (http://www.nlm.nih.gov/mesh/meshhome.html) TGN, (http://www.getty.edu/research/tools/vocabulary/tgn/index.html) etc.

Dublin Core (http://www.dublincore.org/) is an example of a lightweight ontology that is being widely used to specify the characteristics of electronic documents. It specifies a predefined set of document features such as *creator*, *date*, *contributor*, *description*, *format*, etc. Dublin Core annotations can be implemented in languages like RDF and XML. For the RDF annotations, it specifies a RDF Schema with one class and a set of properties for such class, without adding formal constraints on their expected values or to the relationship between them (that is the reason why we can consider it as a lightweight ontology). For instance, the *coverage* property specifies a spatial location, temporal period or jurisdiction, and recommends using a term from an existing thesaurus, but it does not impose the value to be an instance of an actual location, period or jurisdiction, as proposed in its description.

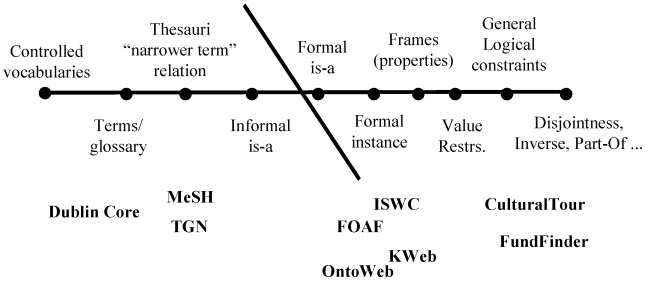
Friend of a Friend (http://www.foaf-project.org/) (FOAF). This initiative aims at creating an annotated Web of homepages for people, groups, companies, etc. It specifies a lightweight ontology that contains some basic classes such as Agent, Person, Organisation, Group, Project, Document, Image, etc., and some basic properties to describe the instances of these classes. This ontology is implemented in RDF Schema.

The OntoWeb (http://www.ontoweb.org/) and KnowledgeWeb (http://knowledgeweb.semanticweb.org/) ontologies, and the publication description ontology (http://annotation.semanticweb.org/iswc/iswc.owl). Similar to the FOAF initiative, these ontologies describe persons, organisations, projects, publications, etc. They are used to describe people and organisations inside those EU networks (OntoWeb and KnowledgeWeb) and to describe the publications of several international conferences and workshops, such as the ISWC series, the SemAnnot workshop series, etc.

The Esperonto (http://www.esperonto.net/) Cultural Tour and Fund Finder applications. These applications show the benefits of upgrading current Web content to the Semantic Web in two domains: culture and funding opportunities. In these applications, documents in both domains are annotated according to corresponding heavyweight ontologies.

Figure 2 shows the relationship between the previous approaches and the classification shown in Figure 1, where two groups can be clearly distinguished.

Figure 2 Annotation approaches and their relationship with Lassila and McGuinness categorisation



2.1 Annotation approaches: examples

To better understand what different annotation approaches consist in, in this section we illustrate how some of them

could be applied to a sample HTML page. Let us suppose that the HTML page shown in Figure 3 belongs to the website of a travel agency and summarises the information about a flight from Madrid to Seattle on 8th February 2003.[1]

Let us see how to apply Dublin Core, a thesaurus about geographical information, and an ontology in the travelling domain to annotate this document. This will provide more details about the main similarities and differences between these approaches.

Figure 3 HTML document that describes the details of a flight

2.1.1 Sample usage of Dublin Core for document annotation

If we annotated this HTML page with Dublin Core, we would include information like the following:

the *contributor* and *creator* is the flight booking service 'www.flightbookings.com'

the *date* would be 1st January 2003, in case that the HTML page has been generated on that specific date

the *description* would be something like 'flight details for a travel between Madrid and Seattle via Chicago on February 8th, 2003'

the document *format* is 'HTML'

the document *language* is 'en', which stands for English, etc.

2.1.2 Sample usage of thesauri for document annotation

Let us suppose that we want to annotate the document with a thesaurus like the Getty Thesaurus of Geographic Names (TGN). In this case we would include information like the following:

Madrid is a reference to the term with ID 7010413 in the thesaurus, which refers to the city of Madrid in Spain

Spain is a reference to the term with ID 1000095, which refers to the Kingdom of Spain in Europe

Chicago is a reference to the term with ID 7013596, which refers to the city of Chicago in Illinois, USA

United States of America is a reference to the term 'United States' with ID 7012149, which refers to the US nation

Seattle is a reference to the term with ID 7014494, which refers to the city of Seattle in Washington, USA.

This is not the only thesaurus that we can use to annotate the document. We could also use the IATA (http://www.iata.org/index.htm) codes to refer to the different airports that appear in the document:

Barajas is a reference to the IATA code MAD

O'Hare International is a reference to the IATA code CHI

Seattle/Tacoma International is a reference to the IATA code SEA.

We could use other thesauri to refer to airline names, plane models, etc.

2.1.3 Sample usage of ontologies for document annotation

Let us suppose that there is a ontology in the travelling domain that we want to use, to annotate the document in Figure 3. This ontology will contain concepts like Flight, Location, Airport, etc., and properties like departure and arrival place, ticket price, etc. Ontology based document annotations usually contain three types of information:

Concept instances relate a part of the document to one or several concepts in an ontology. For example, 'Flight details' may represent an instance of the concept Flight, and can be named as AA7615_Feb08_2003, although concept instances do not necessarily have a name.

Attribute values relate a concept instance with part of the document, which is the value of one of its attributes. For example, 'American Airlines' can be the value of the attribute companyName.

Relation instances that relate two concept instances by some domain specific relation. For example, the flight AA7615_Feb08_2003 and the location Madrid can be connected by the relation departurePlace.

2.2 Relationships between different annotation approaches

As shown in the previous examples, there are some similarities and differences between the different groups of annotations. We detail below some of these relationships:

Dublin Core annotations mainly describe properties of the document itself without providing too many details about its content (only some keywords and natural language descriptions in properties like *subject* or *description*). Ontology based annotations are instead devoted to describe the content of the document, and not its general properties. Finally, thesauri and controlled vocabularies can be used in both approaches to provide agreed terms in specific domains. Consequently, all the approaches complement each other.

In general, Dublin Core annotations are more ambiguous than annotations based on a thesaurus or controlled vocabulary, and these are also more ambiguous, in general, than the annotations based on ontologies. For instance, Dublin Core recommends best practices (nonnormative) for most of the values to be used when describing documents; annotations based on thesauri give clear guidelines on the terms to be used; and finally ontology based annotations normally include relation instances that give 'refer to clear', while in ontology based approaches some of these values will be references to other instances in the ontology.

Finally, the more heavyweight an ontology is, the easier it will be to check constraints in its related document annotations, since heavyweight ontologies define more restrictions on the allowed values of the annotations, on their relationships, etc.

3 Ontology languages for metadata annotation

As commented in the introduction, metadata can be expressed in many different languages, from natural to formal ones. In this section we will focus on those formal languages used so far to annotate metadata based on ontologies.

In the preSemantic Web approaches followed by the $(KA)^2$ initiative and by the SHOE project, the languages used to express metadata were HTML and SHOE (Luke and Heflin, 2000) respectively. In its turn, SHOE used HTML first and XML (Bray et al., 2000) later as their underlying syntax. Let us see some examples based on the example presented in Section 2.

$(KA)^2$ proposed to use an extension of HTML to insert ontology based annotations in Web pages. As described in (Benjamins et al., 1999), this extension was to be understood by agents aware of such extended language, like Ontobroker (Fensel et al., 1999). However, this approach does not specify the language in which the referred ontology must be implemented. Below we present an example of the kind of annotation proposed in $(KA)^2$, applied to the description of our motivating example, where we say that we are describing an instance of the class AA7462, that its departure date is 8th February 2003, and that the arrival place is Seattle.

```
<html>
<head>
<TITLE>Flight Details</TITLE>
<a ONTO="flight:AA7462"/>
</head>
<body>
on Saturday <a ONTO="flight[departureDate=body]">08
February 2003</a> at <b>11:50</b>
Arriving in <a ONTO="flight[arrivalPlace=body]">Seattle
</a> – Seattle/Tacoma International
</body>
</html>
```

The SHOE approach is similar to $(KA)^2$. It consists in an extension of HTML that can be used to describe Web resources. Instead of using the ONTO property inside the A tag for expressing annotations, SHOE proposes to use a set of predefined tags like INSTANCE, CATEGORY, RELATION, etc., which are inserted inside the HTML code of the Web page. Below we show the same example used for illustrating $(KA)^2$ using the HTML version of SHOE. The code presented should be inserted in the source code of the Web page. This approach imposes to use ontologies implemented also in the SHOE language.

```
<INSTANCE KEY="AA7462-Feb08-2003">
 <USE-ONTOLOGY ID="Travel-Ontology"
  URL="http://delicias.dia.fi.upm.es/SHOE/travel.html"
  VERSION="1.0" PREFIX="travel">
 <CATEGORY NAME="travel.AA7462">
 <RELATION NAME="travel.departureDate">
  <ARG POS=1 VALUE="me">
  <ARG POS=2 VALUE="Feb8-2003">
 </RELATION>
 <RELATION NAME="travel.arrivalPlace">
  <ARG POS=1 VALUE="me">
  <ARG POS=2 VALUE="Seattle">
 </RELATION>
</INSTANCE>
```

Currently these approaches have been abandoned and there are only a few tools that can be used to create and process these kinds of annotations, as shown in Section 4. Currently, Ontology based annotations are now implemented using the RDF language (Lassila and Swick, 1999). These annotations can be based on ontologies implemented in RDF Schema (Brickley and Guha, 2004) or OWL (Dean and Schreiber, 2004). Earlier, they could be based on ontologies implemented in OIL (Horrocks et al., 2000) or DAML+OIL (Horrocks and van Harmelen, 2001), the predecessors of OWL.

RDF, RDF Schema and OWL are recommendations of the W3C, and hence they have had a wide acceptance for the implementation of ontologies and of their annotations. Below we show the same example in RDF. This RDF code could refer either to a RDF Schema or to an OWL ontology, and must be inserted also inside the HTML code of the Web page or in a different Web resource that refers to the one being annotated.

```
<AA7462 rdf:ID="AA7462Feb082003">
  <departureDate rdf:datatype="&xsd;date">
    2003-02-08
  </departureDate>
  <arrivalPlace rdf:resource="#Seattle"/>
</AA7462>
```

Though we have presented the most widely adopted approach for ontology based annotation, this does not mean that other ontology languages could be also used to express them. For instance, OCML (Motta, 1999) was used in the Planet-Onto approach and can be generated by the MnM tool, as will be shown in the next section. This language belongs to the so called traditional ontology language group and although it cannot be easily embedded in HTML code, it can be stored in ontology servers and be retrieved from them when needed during the annotation consumption process.

4 Ontology based metadata annotation tools

Ontology based annotation tools, aka ontology based annotators, are primarily designed to allow inserting and maintaining ontology based markups in Web pages. Most of these tools have appeared recently with the emergence of the Semantic Web. Annotators were first conceived as tools that could be used to alleviate the burden of including ontology based annotations manually into Web pages. Since then, many of them have evolved into more complete environments that use Information Extraction (IE) and Machine Learning (ML) techniques to propose semiautomatic annotations for Web documents.

In this section we present the following annotation tools or environments: MnM, OntoMat Annotizer, ONTO-H, SHOE Knowledge Annotator, and UBOT AeroSWARM. This is not an exhaustive list of annotation tools, but rather a selection of some relevant tools with features that differ from each other. We will suppose that this page is not generated dynamically, but that it is static HTML, as these tools have not been designed to annotate content to be generated dynamically.

4.1 MnM

MnM (http://kmi.open.ac.uk/projects/akt/MnM/index.html) (Vargas-Vera et al., 2002) is a standalone application that integrates a Web browser and an ontology viewer and that permits annotating documents manually, semi-automatically, and automatically. It has been developed by the Knowledge Media Institute at the Open University (UK), in the context of the AKT(http://www.aktors.org/) Interdisciplinary Research Collaboration.

MnM is an extensible Java application, based on a plug-in architecture, available for download from the aforementioned URL. For the time being it can load ontologies stored in a WebOnto server or stored in files or URLs in any of the following ontology languages: RDF(S), OWL, and OCML. Similarly, the annotations created with this tool can be used to populate existing ontologies or be attached to the original document (XML format, where the tag names are the names of the concepts, of its attributes, and of its relations).

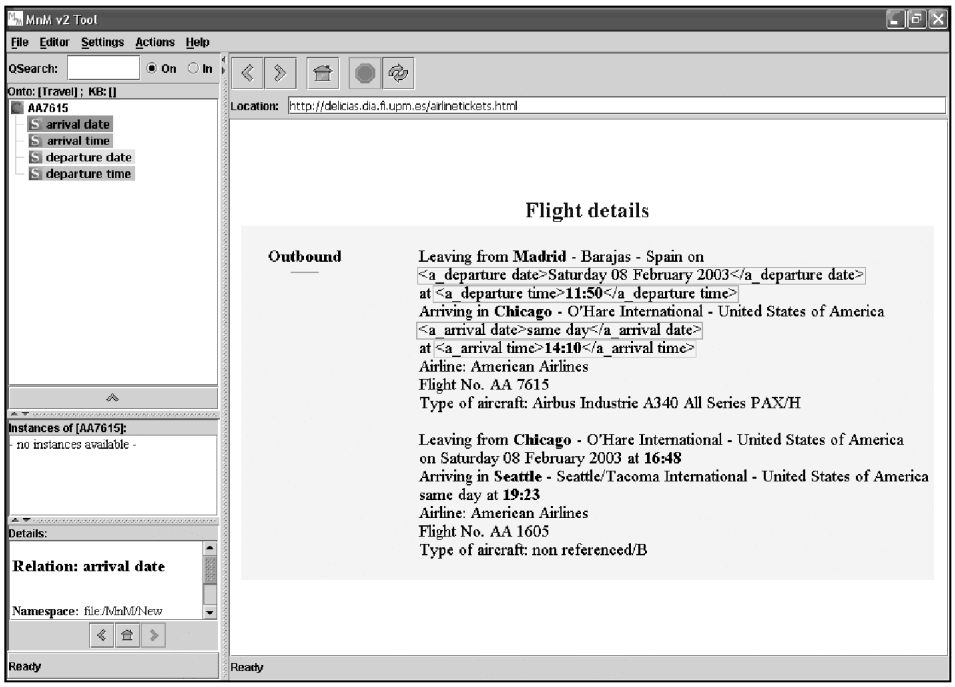
Figure 4 shows our HTML document annotated manually with the instance AA7615_Feb08_2003 of our Web page. We have selected the text that represents this instance in the browser window and the concept instance of which this is an instance. As we can see in the figure, we can add the instance name and the values and target concepts for its attributes and relations respectively.

Concerning the automatic annotation of documents, MnM uses information extraction engines to detect concept instances appearing in documents. These engines must be trained with a set of text and HTML annotated documents so that they generate the rules used to extract information from other documents. When the module is trained, it can be used to detect concept instances, attribute values, and relation instances in other documents. Users may decide to edit the annotations performed by the information extraction module or to leave them as they are generated.

A plugin for the information extraction engine Amilcare (Ciravegna, 2001) is included in the standard distribution. Other information extraction engines could be added as plugins, too.

The annotations generated by this tool can be used in different environments. MnM stores instances in various formats: OCML (so it can be used by any OCML aware tool or application such as WebOnto, Planet-Onto, etc.), RDF, OWL, and XML.

Figure 4 Annotation of the instance AA7615_Feb08_2003 with MnM



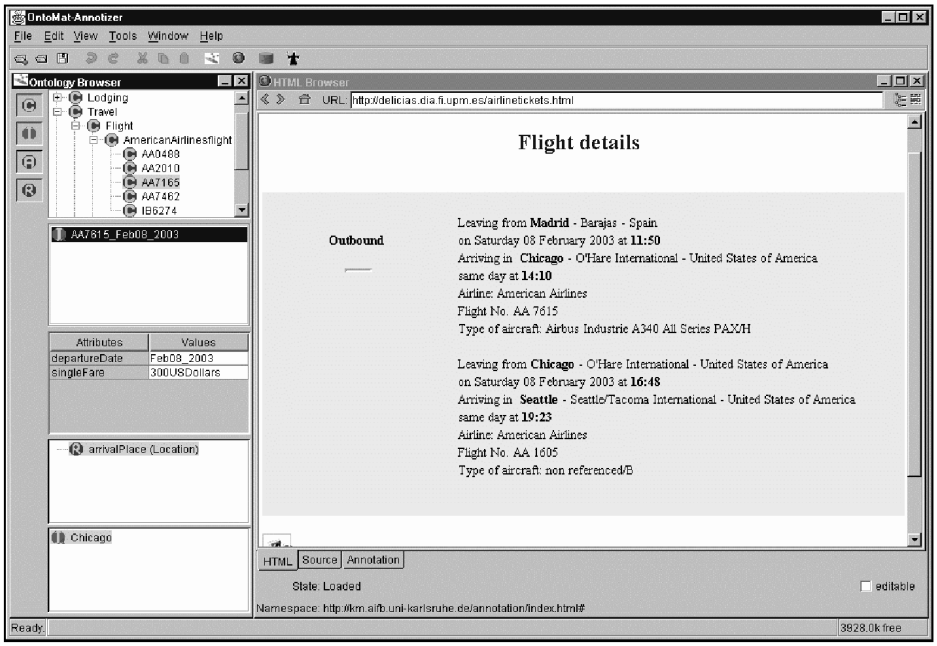
4.2 OntoMat-Annotizer

OntoMat-Annotizer (http://annotation.semanticweb.org/ontomat/index.html) (Handschuh et al., 2001) is a tool for creating, manually, OWL annotations. It is being developed by the Institute AIFB at the University of Karlsruhe.

Like MnM, OntoMat-Annotizer is a Java standalone application with a plugin interface for extensions. It includes an ontology browser to explore ontology concepts and instances, and a HTML browser to display documents and its annotated parts. This tool permits dragging and dropping parts of the text into the annotations being created. In the version 0.8 of this tool, the annotation process is fully manual and does not have any automated support for text annotation.

With this tool, users can create concept instances, with their attributes, and relation instances, as shown in Figure 5. On the left part of the user interface, we can see the attributes and relations of the selected instance that can be filled. In the case of the relations, the tool also presents the instances that can be related to the selected instance with that relation.

Figure 5 Annotation of the instance AA7615_Feb08_2003 with OntoMat-Annotizer



OntoMat-Annotizer loads OWL ontologies. Annotations created with this tool are stored in OWL, either as separate files or embedded in the HTML documents annotated. These annotations can be used by a wide range of applications in the semantic web.

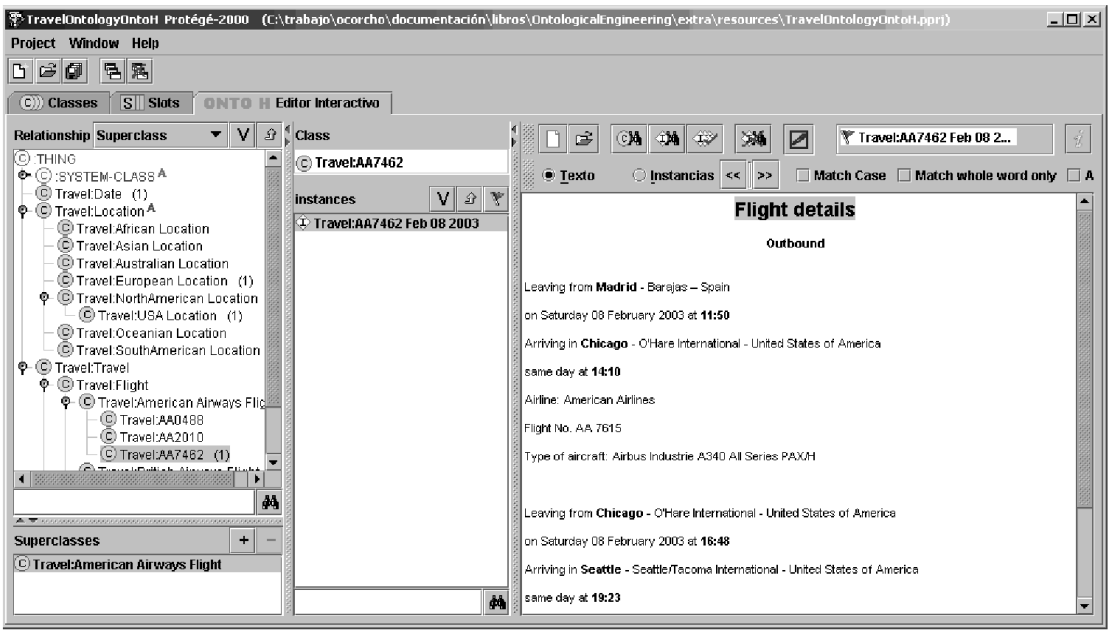
4.3 ONTO-H

ONTO-H (Benjamins et al., 2004) is a tab plugin of the Protégé ontology editor that allows creating annotations of RTF documents. It has been developed by iSOCO (http://www.isoco.com/) in the context of the EU Esperonto project.

Since ONTO-H is integrated in the Protégé editor, it can reuse many of its features, such as the ontology browser, which is similar to the *Classes&Instances* tab that is provided in the Protégé default distribution. Besides, ONTO-H users can reuse all the functionalities provided by the Protégé editor, such as the ontology editing and browsing functions, ontology visualisation, merge, etc., and more important, all the import and export functions of the editor, which give great flexibility with respect to the formats in which the annotations will be stored.

Figure 6 shows the user interface of this annotation tool while annotating our document with the flight details, which has been previously converted to RTF format. In the screenshot we can see that the 'Flight details' part of the document has been selected and dragged&dropped to the instances pane, giving as a result the creation of an instance of the flight *AA7462*.

Figure 6 Annotation of the instance AA7615_Feb08_2003 with ONTO-H



Besides the drag&drop functions for creating annotations manually, the editor also gives suggestions for the annotation of parts of the text, by recognising named entities, annotations already existing with the same name or with a synonym, etc. In this sense, ONTO-H is a tool that can be mainly used for supervised annotation, rather than for a completely manual annotation process.

Finally, ONTO-H allows using declarative rules implemented in the DROOLS (http://www.drools.org/) rule language. These rules are used to prompt the user automatically, with instance edition forms that allow creating new related instances to the one that has just been dropped onto the instance pane. This function has proven to be very useful in the cultural domain, where instances of a piece of work done by an artist are, most of the time, related to instances of expressions and manifestations of such work.

4.4 SHOE knowledge annotator

The SHOE Knowledge Annotator (http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html) (Heflin and Hendler, 2001) is a tool for creating manual annotations in HTML pages with the SHOE language. It was developed by the Parallel Understanding Systems Group, at the Department of Computer Science, University of Maryland. This tool has been the basis for the creation of SMORE, (http://www.mindswap.org/~aditkal/editor.shtml) a more complex tool.

The SHOE Knowledge Annotator is available as a Java applet and as a standalone Java application. Both of them have the same functionalities. Annotations can refer to concepts and relations from one or several ontologies implemented in SHOE, which means that this tool creates annotations of instances of concepts, of their attribute values, and instances of relations.

Figures 7 and 8 show the user interface of the standalone application. Unlike other tools, the HTML document is not browsed as in common Web browsers: only its source code can be accessed. The upper left window of both figures contains the concept instances. When one of these instances is selected, the upper right and lower windows are updated with information related to it. The upper right window contains the names of the ontologies that the instance uses. The lower window contains the claims made by this instance. Two types of claims can be made here:

Figure 7 Edition of the instance AA7615_Feb08_2002 with SHOE Knowledge Annotator

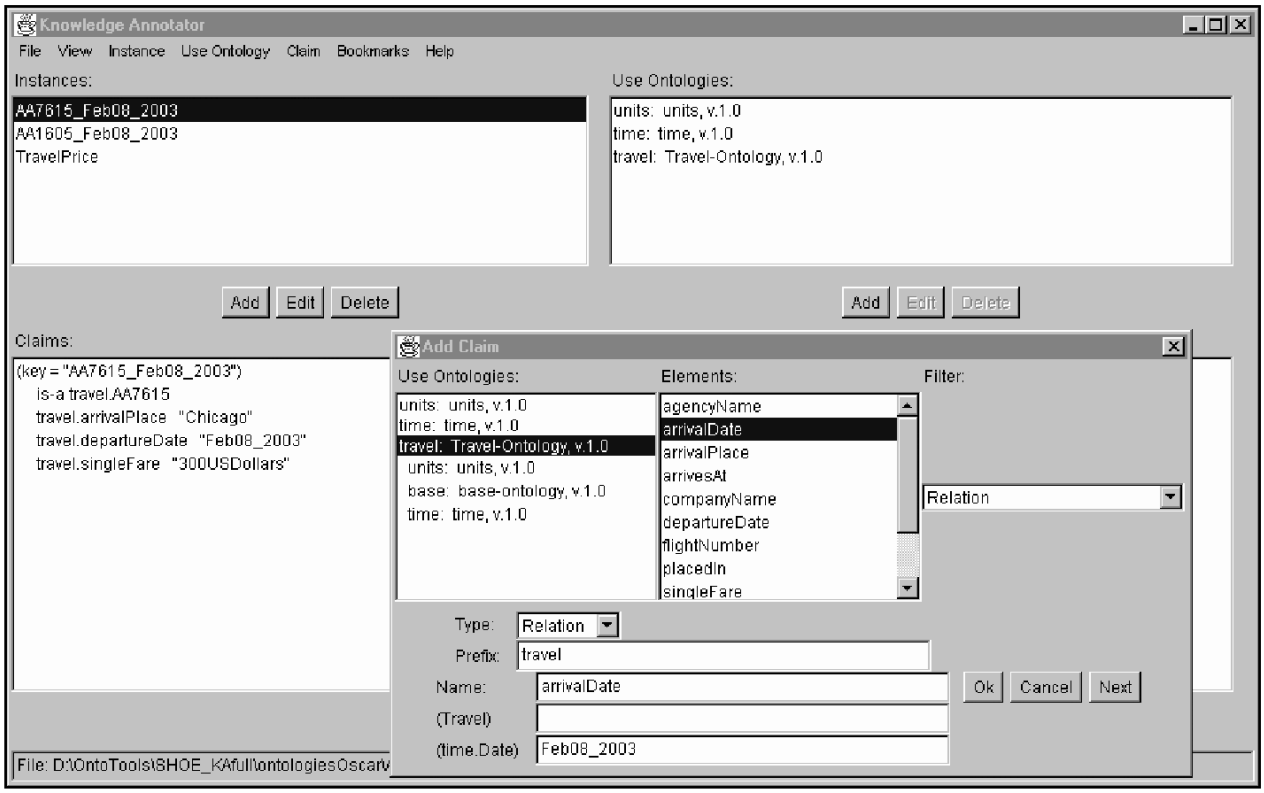
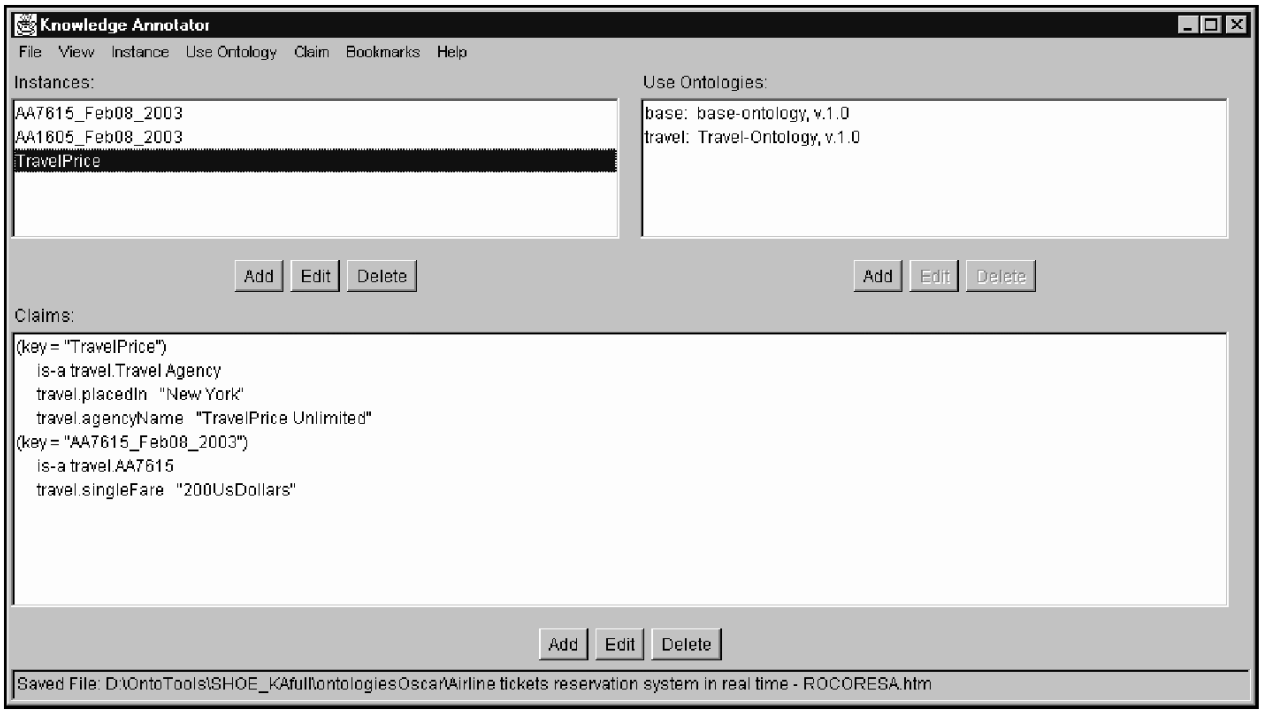


Figure 8 Edition of the instance TravelPrice and some claims about the instance AA7615_Feb08_2003 with SHOE Knowledge Annotator



Claims of information about the instance. In Figure 7, the instance AA7615_Feb08_2003 claims that it is an instance of the class AA7615, that it arrives at Chicago, that its departure date is February 8, 2003, and that its single fare costs 300$.

Claims of information about other instances. In 8th Figure, the instance TravelPrice not only claims that it is an instance of the class Travel Agency, which is located in New York and whose name is 'TravelPrice Unlimited', but it also claims that the single fare for the instance AA7615_Feb08_2003 is 200$ (there is a cheaper negotiated price between this travel agency and the American Airlines flight company).

The SHOE code corresponding to these annotations is embedded in the original HTML document.

4.5 UBOT AeroSWARM

AeroSWARM (http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html) (Kogut and Holmes, 2001) generates, automatically, RDF annotations from text documents. It was developed by Lockheed Martin Corporation as part of the UBOT (UML Based Ontology Toolset) project.

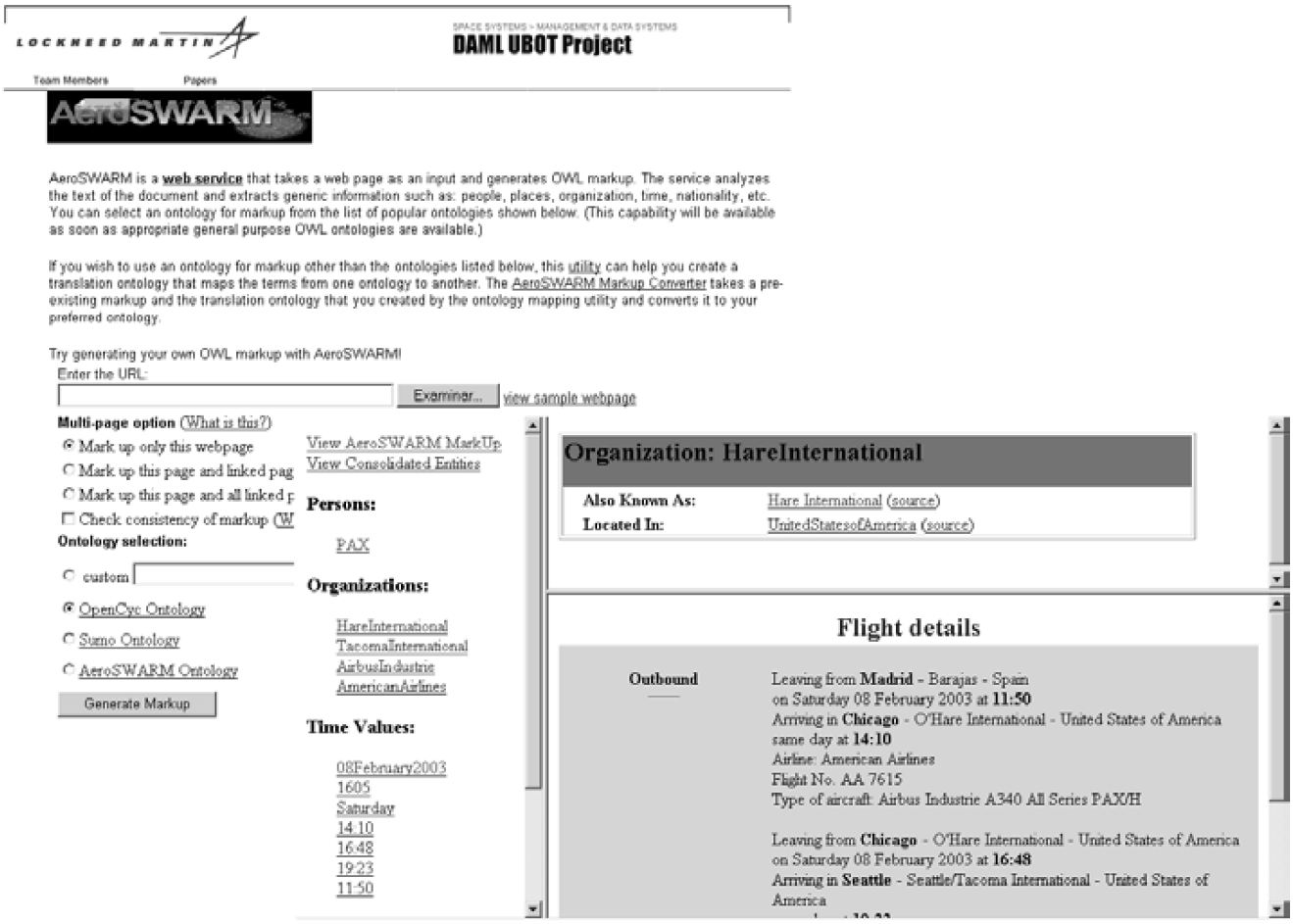
AeroSWARM is available both as a Web form and as a standalone application. In the Web version, shown in Figure 9, users send a text file and AeroSWARM sends back the RDF annotations for that text. These annotations are created according to the OWL versions of OpenCyc, (http://www.cyc.com/2003/04/01/cyc) SUMO, (http://reliant.teknowledge.com/DAML/SUMO.owl) and AeroSWARM (http://ubot.lockheedmartin.com/ubot/2004/04/aeroswarmOntology.owl).

The automatic annotation feature of AeroSWARM is supported by the text mining system, AeroText. This system parses natural language text and extracts those items that have any correspondence with the underlying ontology used. The default extraction rules of this text mining system can also be modified.

AeroSWARM generates instances of concepts (proper nouns, common nouns, dates, currency quantities, etc.), attribute values, and instances of properties (a person belongs to an organisation, an organisation is based in a location, etc.).

Since the annotations created by AeroSWARM are provided in RDF, any RDF aware tool can use them as long as they are appended to the corresponding web page. AeroSWARM could also be used as an automatic annotation service to provide RDF annotations online.

Figure 9 UBOT AeroSWARM annotation web server



5 Conclusion and open research problems

In this paper we have described the most widely used document annotation approaches. We have shown the similarities and differences between the use of Dublin Core for annotating the properties of the document itself and the use of thesauri, controlled vocabularies and ontologies for annotating the document contents. All these approaches can be characterised according to a continuous line between the highly informal, lightweight vocabularies to very formal heavyweight ontologies.

Then we have described formal languages used both in the past, and currently, for annotating web resources, in the context of the semantic web, and tools that allow creating ontology based annotations. During the last years, especially with the emergence of the semantic web, many advances in document annotation have seen the light. However, there are still many open issues (with different degrees of maturity) to be solved. In this paper we have not pretended to present an exhaustive state of the art on document annotation: in fact, we have focused in Sections 3 and 4 in ontology based document annotation, and we have not covered all the existing approaches, but only some of the most relevant ones.

One of these open issues is maybe one of the most important aspects to be considered in order to make the upgrade of current web content to the semantic web a reality. The set of tools presented in Section 4 are mainly manual or semi-automatic annotation tools, the latter based on information extraction and/or machine learning techniques. The manual annotation of documents is a high cost and error prone task, as has been proven by preSemantic Web initiatives. To alleviate this task, an important effort is currently being made in the automation of document annotations, and the result is some degree of automation as shown in some of the descriptions provided in Section 4. However, there is still some work to do to achieve a complete automation of the annotation process.

Finally, it is important to note that there are many other aspects of document annotation that could have been described in this paper, such as the quality of document annotations, the management of inconsistencies in distributed annotated data, the lifecycle of annotations and their related vocabularies (e.g., the management of the evolution of the vocabularies in which the annotations are based), the existence of annotation management systems for querying, storage, reasoning, etc. Clearly all these aspects would deserve future special issues, since much research is being done in all these areas.

Acknowledgement

This work has been supported by the EU project Esperonto (IST-2001-34373).

References

Benjamins, V.R., Contreras, J., Blázquez, M., Dodero, J.M., García, A., Navas, E., Hernández, F. and Wert, C. (2004) 'Cultural heritage and the semantic web', in Bussler, C., Davies, J., Fensel, D. and Studer, R. (Eds.): *The Semantic Web: Research and Applications, First European Semantic Web Symposium (ESWS2004)*, Springer-Verlag, pp.433–444.

Benjamins, V.R., Contreras, J., Corcho, O. and Gómez-Pérez, A. (2002) 'Six challenges for the semantic web', in Cristani, M. (Ed.): *KR2002 Workshop on the Semantic Web*, Toulouse, France.

Benjamins, V.R., Fensel, D., Decker, S and Gómez-Pérez, A. (1999) '$(KA)^2$: building ontologies for the internet: a mid term report', *International Journal of Human Computer Studies*, Vol. 51, pp.687–712.

Bray, T., Paoli, J., Sperberg-McQueen, C.M. and Maler, E. (2000) *Extensible Markup Language (XML) 1.0*, W3C Recommendation, http://www.w3.org/TR/REC-xml.

Brickley, D. and Guha, R.V. (2004) *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation, http://www.w3.org/TR/PR-rdf-schema.

Ciravegna, F. (2001) 'Adaptive information extraction from text by rule induction and generalisation', in Nebel, B. (Ed.): *17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle, Washington, Morgan Kauffmann Publishers, San Francisco, California, pp.1251–1256.

Dean, M. and Schreiber, G. (2004) *OWL Web Ontology Language Reference*, W3C Recommendation, http://www.w3.org/TR/owl-ref/.

Domingue, J. and Motta, E. (2000) 'PlanetOnto: from news publishing to integrated knowledge management support', *IEEE Intelligent Systems and their Applications*, Vol. 15, No. 3, pp.26–32.

Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.P., Staab, S., Studer, R. and Witt, A. (1999) 'On2broker: semantic-based access to information sources at the www', in de Bra, P. and Leggett, J. (Eds.): *World Conference on the WWW and Internet (WebNet'99)*, Honolulu, Hawaii, pp.1366–1371.

Gómez-Pérez, A., Fernández-López, M. and Corcho, O. (2003) *Ontological Engineering: with Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web*, Springer-Verlag, London.

Handschuh, S., Staab, S. and Mäedche, A. (2001) 'CREAM – creating relational metadata with a component-based, ontology-driven annotation framework', in Gil, Y., Musen, M. and Shavlik, J. (Eds.): *First International Conference on Knowledge Capture (KCAP'01)*, ACM Press, Victoria, Canada, 1-58113-380-4, New York, pp.76–83.

Heflin, J.D. and Hendler, J.A. (2001) 'A portrait of the semantic web in action', *IEEE Intelligent Systems and their Applications*, Vol. 16, No. 2, pp.54–59.

Horrocks, I., Fensel, D., Harmelen, F., Decker, S., Erdmann, M. and Klein, M. (2000) 'OIL in a nutshell', in Dieng, R. and Corby, O. (Eds.): *12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00)*, Juan-Les-Pins, France, *Lecture Notes in Artificial Intelligence LNAI 1937*, Springer-Verlag, Berlin, Germany, pp.1–16.

Horrocks, I. and van Harmelen, F. (Eds.) (2001) *Reference Description of the DAML+OIL (March 2001) Ontology Markup Language*, Technical report. http://www.daml.org/2001/03/reference.html.

Kogut, P. and Holmes, W. (2001) 'AeroDAML: applying information extraction to generate daml annotation from web pages', in Handschuh, S., Dieng, R. and Staab, S. (Eds): *KCAP'01 Workshop on Semantic Markup and Annotation*, Victoria, Canada.

Lassila, O. and McGuinness, D. (2001) *The Role of Frame-Based Representation on the Semantic Web*, Technical Report KSL-01-02, Knowledge Systems Laboratory, Stanford University, Stanford, California.

Lassila, O. and Swick, R. (1999) *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, http://www.w3.org/TR/REC-rdf-syntax/.

Luke, S. and Heflin, J.D. (2000) *SHOE 1.01. Proposed Specification*, Technical Report, Parallel Understanding Systems Group, Department of Computer Science, University of Maryland, http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm.

Luke, S., Spector, L., Rager, D. and Hendler, J.A. (1997) *Ontology-based Web Agents*, *First International Conference on Autonomous Agents (AA'97)*, Johnson, W.L. and Jennings, N (Ed.): Marina del Rey, California, ACM Press, New York, pp.59–66.

Motta, E. (1999) *Reusable Components for Knowledge Modelling: Principles and Case Studies in Parametric Design*, IOS Press, Amsterdam, The Netherlands.

Studer, R., Benjamins, V.R. and Fensel, D. (1998) 'Knowledge engineering: principles and methods', *IEEE Transactions on Data and Knowledge Engineering*, Vol. 25, Nos. 1–2, pp.161–197.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. and Ciravegna, F. (2002) 'MnM: ontology driven semi-automatic and automatic support for semantic markup', in Gómez-Pérez, A. and Benjamins, V.R. (Eds.): *13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, Springer Verlag, pp.379–391.

Websites

Advanced Knowledge Technologies: http://www.aktors.org/.

http://annotation.semanticweb.org/iswc/iswc.owl.

http://annotation.semanticweb.org/ontomat/index.html.

http://kmi.open.ac.uk/projects/akt/MnM/index.html.

http://knowledgeweb.semanticweb.org/.

http://reliant.teknowledge.com/DAML/SUMO.owl.

http://ubot.lockheedmartin.com/ubot/2004/04/aeroswarmOntology.owl.

http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html.

http://www.acemedia.org/.

http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html.

http://www.cyc.com/2003/04/01/cyc.

http://www.drools.org/.

http://www.esperonto.net/.

http://www.dublincore.org/.

http://www.foaf-project.org/.

http://www.iata.org/index.htm.

http://www.isoco.com/.

http://www.mindswap.org/.

http://www.mindswap.org/~aditkal/editor.shtml.

http://www.ontoweb.org/.

http://www.schema-ist.org/.

(http://www.nlm.nih.gov/mesh/meshhome.html).

(http://www.getty.edu/research/tools/vocabulary/tgn/index.html).

Note

[1]This example is used in (Gómez-Pérez et al., 2003) to show how different ontology-based annotation tools can be applied, and will be also used in Section 4 of this paper.