

## Foreword

Capturing meaning and knowledge by using markup techniques and by supporting semantic annotations is a major technique for creating semantic metadata. It is beneficial in a wide range of content-oriented intelligent applications. One important application is the Semantic Web. The research about the WWW currently strives to augment syntactic information already present in the Web by semantic metadata in order to achieve a Semantic Web that both human and software agents can access, understand and further process. Here, one of the most urgent challenges is a meaning and knowledge-capturing problem, i.e. how one may turn existing syntactic resources into semantic and knowledge structures. A solution is to markup web document in order to create semantic metadata on the web or to author new documents in a way that they contain semantic markup directly. Since the Web is increasingly containing (multilingual) multimedia material, an important application within the Semantic Web consists in the content indexing and searching of multimedia (and multilingual) data. It is difficult to completely process the content of multimedia data, even with technologies based on natural language processing, image processing, machine vision and speech recognition. Therefore, Semantic annotation is one of the promising methodologies to define semantic structures on the content of multimedia material, also present in the web.

This workshop continues the discussion started at the Workshop of Knowledge Markup and Semantic Annotation (SemAnnot2001), which was held at Victoria, B.C, Canada, October 21, 2001 as a workshop of the First International Conference on Knowledge Capture (K-Cap 2001) and is now for the first time organized as a satellite event of the International Semantic Web Conference.

The workshop is bringing together researchers and practitioners from such research areas as the Semantic Web, knowledge acquisition, language technology, multimedia processing, information science etc. The contributions discuss various aspects of knowledge markup and semantic annotation in an interdisciplinary way.

The workshop proposes three types of presentations: long oral, short oral (position papers) and posters. All the presentations contribute in an equal way to the discussion and differ only in the presentation format.

As mentioned above, the workshop does not address only the semantic annotation of textual documents (to be) published on the web, but also the increasing number of multimedia material that is being made available on the Internet. In most of the papers, a large place is made to the role of ontologies (tools), as the main encoding of semantic and knowledge to be used in annotation, indexing and searching.

More specially, papers are addressing a wide variety of topic, all related to the main topics of the workshop: Semantic Annotation and Knowledge markup:

What are they, what are the basic components coming into play, how can we (automatically or with tools) generate semantic annotation, how can they be used for improving semantic web applications.

So contributions are dealing with ontology detection and selection, (e.g. Buitelaar), the creation of semantic metadata (Mori et al), the building of corpora for SW application (Möller et al), the use of semantic web authoring tools (McGregor et al), multimedia in the SW, including semantic annotation schemes and ontology framework for multimedia applications (Feinberg, Hollink, Feinberg and Shaw, Tummarello et al., Dasiopoulou et al), the issue of distributed knowledge (Nickles et al.), a discussion of SW-based real world applications (De Blasio et al, Tijerino), the semantic annotation of databases (Hyvoenen et al), the semantic annotation of Web Pages, including the use of human language technology: (Witbrock, Black et al), the management of semantic annotation and knowledge markup (Kawazoe et al.), ontology extraction and bootstrapping (Tijerino), presentation and usability issues for the Semantic Web, for example for visually impaired (Harper and Bechhofer), the annotation of the ontologies themselves (Parsia and Kalyanpur). Last but not least, in some of the contributions, the main annotation strategy for the Semantic Web, with the use of ontologies, are seen with a critical eye. There might be better annotation (and presentation strategies, for example for the visually impaired), and there might be other strategies for introducing intelligence in the WWW, as the one defined by the Semantic Web initiative. Halpin and Thomson, as well as Harper and Bechhofer, among others are proposing an interesting discussion on those issues.

We wish to express our appreciation to all the authors of submitted papers and to the members of the program committee for making the workshop a valuable contribution to the vision of the Semantic Web.

November 2004

Siegfried Handschuh, Therry Declerck (co-chairs)  
Marja-Riitta Koivunen, Rose Dieng  
Richard Benjamins, Steffen Staab

Acknowledgment: This workshop was partially funded by the 5th Framework EU project "Esperanto" (IST-2001-34373), the EU IST project "DOT.KOM", the 6th Framework EU IST project "aceMedia", and the DAPRA DAML programme.

# Organization

The Workshop of Knowledge Markup and Semantic Annotation (SemAnnot2004) was organized as a workshop within the Third International Semantic Web Conference (ISWC 2004). It was held on November 8, 2004, in Hiroshima, Japan.

## Organization Committee

Siegfried Handschuh (co-chair)  
Institute AIFB  
<http://www.aifb.uni-karlsruhe.de/WBS/sha>  
[handschuh@aifb.uni-karlsruhe.de](mailto:handschuh@aifb.uni-karlsruhe.de)

Thierry Declerck (co-chair)  
Saarland University & DFKI GmbH  
<http://www.dfki.de/~declerck/>  
[declerck@dfki.de](mailto:declerck@dfki.de)

Marja-Riitta Koivunen  
Annotea project  
<http://www.w3.org/2001/Annotea/>  
<http://www.annotea.org/mozilla/ubi.html>  
[marja@annotea.org](mailto:marja@annotea.org)

Rose Dieng  
The french National Institute for  
Research in Computer Science and Control (INRIA)  
<http://www.inria.fr/Rose.Dieng>  
[Rose.Dieng@inria.fr](mailto:Rose.Dieng@inria.fr)

Richard Benjamins  
iSOCO  
<http://www.isoco.com>  
[richard.benjamins@isoco.com](mailto:richard.benjamins@isoco.com)

Steffen Staab  
Institute for Informatics  
University of Koblenz-Landau  
<http://www.uni-koblenz.de/FB4>  
[staab@uni-koblenz.de](mailto:staab@uni-koblenz.de)

## Program Committee

Ana Belen Benitez  
(*Columbia University*)

Paul Buitelaar  
(*DFKI*)

Philipp Cimiano  
(*AIFB*)

Nigel Collier  
(*National Institute of Informatics*)

Olivier Corby  
(*INRIA*)

Grit Denker  
(*SRI International*)

Martin Frank  
(*ISI*)

Fabien Gandon  
(*CMU*)

Carole Goble  
(*University of Manchester*)

Harry N. Keeling  
(*Howard University*)

Libby Miller  
(*ILRT*)

Guenter Neumann  
(*DFKI*)

Sofia Pinto  
(*Instituto Superior Tecnico*)

Alun Preece  
(*University of Aberdeen*)

Guus Schreiber  
(*Free University Amsterdam*)

Sylvie Szulman  
(*LIPN, University Paris-Nord*)

Hideaki Takeda  
(*National Institute of Informatics*)

Maria Vargas-Vera  
(*Open University*)

Martin Wolpers  
(*Learning Lab Lower Saxony*)

Yiannis Kompatsiaris  
(*CERTH/ITI*)

Ebroul Izquierdo  
(*Queen Mary College*)

Joachim Koehler  
(*Fraunhofer Institute*)

Copyright remains with the authors, and permission to reproduce material printed here should be sought from them. Similarly, pursuing copyright infringements, plagiarism, etc. remains the responsibility of authors.





# Table of Contents

## Full Papers

An Ontology Framework For Knowledge-Assisted Semantic Video Analysis and Annotation . . . . .	1
<i>S. Dasiopoulou, V. K. Papastathis, V. Mezaris, I. Kompatsiaris, M. G. Strintzis</i>	
Catalog Search Engine: Semantics applied to products search . . . . .	11
<i>Jacques-Albert De Blasio, Takahiro Kawamura, and Tetsuo Hasegawa</i>	
Social Annotation of Semantically Heterogeneous Knowledge . . . . .	21
<i>Matthias Nickles, Tina Froehner, Gerhard Weiss</i>	
Adding Spatial Semantics to Image Annotations . . . . .	31
<i>Laura Hollink, Giang Nguyen, Guus Schreiber, Jan Wielemaker, Bob Wielinga, Marcel Worring</i>	
Ontology-enablement of a system for semantic annotation of digital documents . . . . .	41
<i>William J Black, Simon Jowett, Thomas Mavroudakakis, John McNaught, Babis Theodoulidis, Argyrios Vasilakopoulos, Gian-Piero Zarri, Kalliopi Zervanou</i>	
Keyword Extraction from the Web for Personal Metadata Annotation . . . .	51
<i>Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, Boi Faltings</i>	
Annotation of Heterogeneous Database Content for the Semantic Web . . . .	61
<i>Eero Hyvönen, Mirva Salminen, and Mikka Junnila</i>	
Automated OWL Annotation Assisted by a Large Knowledge Base . . . . .	71
<i>Michael Witbrock, Kathy Panton, Stephen L. Reed, Dave Schneider, Bjørn Aldag, Mike Reimers, Stefano Bertolo</i>	
Multimedia Distributed Knowledge Management in MIAKT . . . . .	81
<i>David Dupplaw, Srinandan Dasmahapatra, Bo Hu, Paul Lewis, Nigel Shadbolt</i>	
Low Cost Mark-Up for Lightweight Semantics . . . . .	91
<i>Simon Harper and Sean Bechhofer</i>	

## Short Papers/Poster

Managing the semantics of coreference relations with Open Ontology Forge	103
<i>Ai Kawazoe, Asanobu Kitamoto, Nigel Collier</i>	

MetaDesk: A Semantic Web Desktop Manager . . . . .	107
<i>Robert MacGregor, Sameer Maggon, Baoshi Yan</i>	
MPEG7ADB: Automatic RDF annotation of audio files from low level low level MPEG-7 metadata . . . . .	111
<i>Giovanni Tummarello, Christian Morbidoni, Francesco Piazza, Paolo Puliti</i>	
Action: A Framework for Semantic Annotation of Events in Video . . . . .	115
<i>Melanie Feinberg, Ryan Shaw</i>	
Integrating Event Frame Annotation into the Open Ontology Forge An- notation Tool . . . . .	119
<i>Tuangthong Wattarujeeekrit, Nigel Collier</i>	
<b>Position Papers</b>	
Annotating OWL Ontologies . . . . .	125
<i>Bijan Parsia, Aditya Kalyanpur</i>	
OntoSelect: Towards the Integration of an Ontology Library, Ontology Selection and Knowledge Markup . . . . .	127
<i>Paul Buitelaar</i>	
Bootstrapping Domain Ontologies for Rapid Semantic Annotation of User- Friendly Semantic Web Content . . . . .	129
<i>Yuri A. Tijerino</i>	
Towards an Integrated Corpus for the Evaluation of Named Entity Recog- nition and Object Consolidation . . . . .	131
<i>Knud Möller, Alexander Schutz, Stefan Decker</i>	

# An Ontology Framework For Knowledge-Assisted Semantic Video Analysis and Annotation

S. Dasiopoulou<sup>1,2</sup>, V. K. Papastathis<sup>2</sup>, V. Mezaris<sup>1,2</sup>, I. Kompatsiaris<sup>2</sup> and  
M. G. Strintzis<sup>1,2</sup> \*

<sup>1</sup> Information Processing Laboratory, Electrical and Computer Engineering  
Department, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece

<sup>2</sup> Informatics and Telematics Institute (ITI)/ Centre for Research and Technology  
Hellas (CERTH), 1st Km Thermi-Panorama Rd, Thessaloniki 57001, Greece  
email: strintzi@iti.gr

**Abstract.** An approach for knowledge assisted semantic analysis and annotation of video content, based on an ontology infrastructure is presented. Semantic concepts in the context of the examined domain are defined in an ontology, enriched with qualitative attributes of the semantic objects (e.g. color homogeneity), multimedia processing methods (color clustering, respectively), and numerical data or low-level features generated via training (e.g. color models, also defined in the ontology). Semantic Web technologies are used for knowledge representation in RDF/RDFS language. Rules in F-logic are defined to describe how tools for multimedia analysis should be applied according to different object attributes and low-level features, aiming at the detection of video objects corresponding to the semantic concepts defined in the ontology. This supports flexible and managed execution of various application and domain independent multimedia analysis tasks. This ontology-based approach provides the means of generating semantic metadata and as a consequence Semantic Web services and applications have a greater chance of discovering and exploiting the information and knowledge in multimedia data. The proposed approach is demonstrated in the Formula One and Football domains and shows promising results.

## 1 Introduction

As a result of recent progress in hardware and telecommunication technologies, multimedia has become a major source of content on the World Wide Web, used in a wide range of applications in areas such as content production and distribution, telemedicine, digital libraries, distance learning, tourism, distributed CAD/CAM, GIS, etc. The usefulness of all these applications is largely determined by their accessibility and portability and as such, multimedia data sets

---

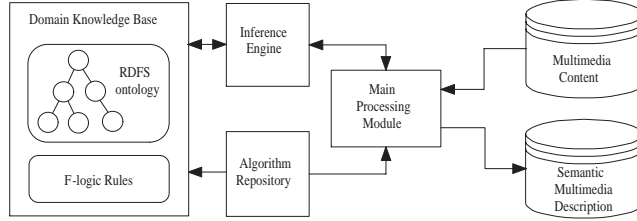
\* This work was supported by the European Commission under contracts FP6-001765 aceMedia and FP6-507482 KnowledgeWeb.

present a great challenge in terms of storing, querying, indexing and retrieval. In addition, the rapid increase of the available amount of multimedia information has revealed an urgent need for developing intelligent methods for understanding and managing the conveyed information. To face such challenges developing faster hardware or more sophisticated algorithms has become insufficient. Rather, a deeper understanding of the information at the semantic level is required [1]. This results in a growing demand for efficient methods for extracting semantic information from such content, since this is the key enabling factor for the management and exploitation of multimedia content.

Although new multimedia standards, such as MPEG-4 and MPEG-7 [2], provide the needed functionalities in order to manipulate and transmit objects and metadata, their extraction, and that most importantly at a semantic level, is out of the scope of the standards and is left to the content developer. Extraction of features and object recognition are important phases in developing general purpose multimedia database management systems [3]. Significant results have been reported in the literature for the last two decades, with successful implementation of several prototypes [4]. However, the lack of precise models and formats for object and system representation and the high complexity of multimedia processing algorithms make the development of fully automatic semantic multimedia analysis and management systems a challenging task.

This is due to the difficulty, often mentioned as the *semantic gap*, in capturing concepts mapped into a set of image and/or spatiotemporal features that can be automatically extracted from video data without human intervention [5]. The use of domain knowledge is probably the only way by which higher level semantics can be incorporated into techniques that capture the semantics through automatic parsing. Such techniques are turning to knowledge management approaches, including Semantic Web technologies to solve this problem [6]. A priori knowledge representation models are used as a knowledge base that assists semantic-based classification and clustering [7, 8]. In [9] and [10] automatic associations between media content and formal conceptualizations are performed based on the similarity of visual features extracted from a set of pre-annotated media objects and the examined media objects. In [11], semantic entities, in the context of the MPEG-7 standard, are used for knowledge-assisted video analysis and object detection, thus allowing for semantic level indexing. In [12], the problem of bridging the gap between low-level representation and high-level semantics is formulated as a probabilistic pattern recognition problem. In [13], an object ontology, coupled with a relevance feedback mechanism, is introduced to facilitate the mapping of low-level to high-level features and allow the definition of relationships between pieces of multimedia information.

In this paper, an approach for knowledge assisted semantic content analysis and annotation, based on a multimedia ontology infrastructure, is presented. Content-based analysis of multimedia requires methods which will automatically segment video sequences and key frames into image areas corresponding to salient objects, track these objects in time, and provide a flexible framework for object recognition, indexing, retrieval and for further analysis of their relative



**Fig. 1.** Overall system architecture.

motion and interactions. This problem can be viewed as relating symbolic terms to visual information by utilizing syntactic and semantic structure in a manner related to approaches in speech and language processing [14]. In the proposed approach, semantic and low-level attributes of the objects to be detected in combination with appropriately defined rules determine the set of algorithms and parameters required for the objects detection. Semantic concepts within the context of the examined domain are defined in an ontology, enriched with qualitative attributes of the semantic objects, multimedia processing methods, and numerical data or low-level features generated via training. Semantic Web technologies are used for knowledge representation in RDF/RDFS language. Processing may then be performed by using the necessary processing tools and by relating high-level symbolic representations to extracted features in the signal (image and temporal feature) domain. F-logic rules are defined to describe how tools for multimedia analysis should be applied according to different object attributes and low-level features, aiming at the detection of video objects corresponding to the semantic concepts defined in the ontology. The proposed approach, by exploiting the domain knowledge modelled in the ontology, enables the recognition of the underlying semantics of the examined video, providing a first level semantic annotation. The general system architecture is shown in Fig. 1

Following this approach, the multimedia analysis and annotation process largely depends on the knowledge base of the system and as a result the method can easily be applied to different domains provided that the knowledge base is enriched with the respective domain ontology. Extending the knowledge base with spatial and temporal objects interrelations would be an important step towards the detection of semantically important events for the particular domain, achieving thus a finer, high-level semantic annotation. In addition, the ontology-based approach also ensures that semantic web services and applications have a greater chance of discovering and exploiting the information and knowledge in multimedia data.

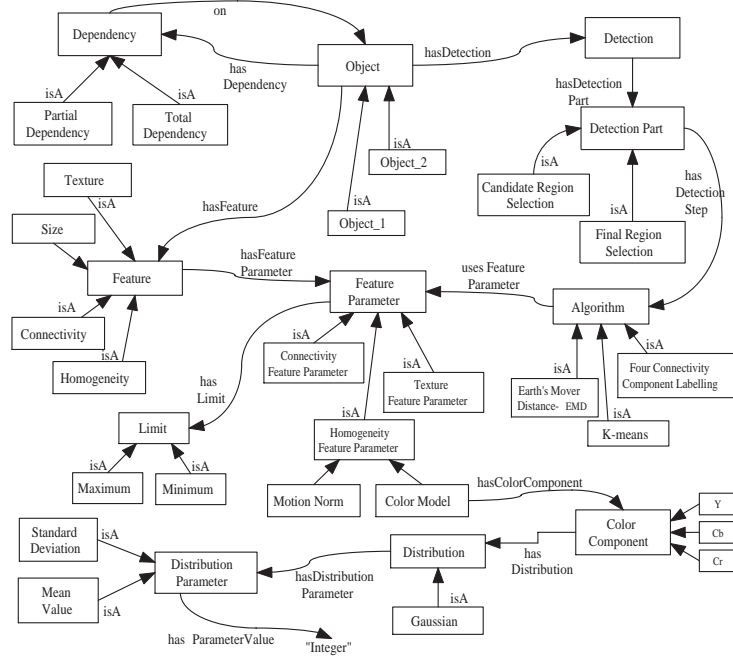
The remainder of the paper is organized as follows: section 2 a detailed description of the ontology and rules developed is given, while in section 3, its application to the Formula One domain is described. Experimental results are presented in section 4. Finally, conclusions are drawn in section 5.

## 2 Multimedia Analysis Ontology Development and Rule Construction

In order to realize the knowledge-assisted multimedia content semantic analysis and annotation technique explained in the previous section, an analysis and a domain ontology are constructed. The *multimedia analysis ontology* is used to support the detection process of the corresponding domain specific objects. Knowledge about the domain under discourse is also represented in the form of an ontology, namely the *domain specific ontology*. The domain-independent, primitive classes comprising the analysis ontology serve as attachment points allowing the integration of the two ontologies. Practically, each domain ontology comprises a specific instantiation of the multimedia analysis ontology providing the corresponding color models, restrictions e.t.c as will be demonstrated in more detail in section 3.

Object detection in general considers the exploitation of objects characteristic features in order to apply the most appropriate detection steps for the analysis process in the form of algorithms and numerical data generated off-line by training (e.g. color models). Consequently, the development of the proposed analysis ontology deals with the following concepts (RDFS classes) and their corresponding properties, as illustrated in Fig. 2:

- Class **Object**: the superclass of all video objects to be detected through the analysis process. Each object instance is related to appropriate feature instances by the **hasFeature** property and to one or more other objects through a set of appropriately defined spatial properties.
- Class **Feature**: the superclass of multimedia low-level features associated with each object.
- Class **Feature Parameter** which denotes the actual qualitative descriptions of each corresponding feature. It is subclassed according to the defined features, i.e. to **Connectivity Feature Parameter**, **Homogeneity Feature Parameter** e.t.c.
- Class **Limit**: it is subclassed to **Minimum** and **Maximum** and allows the definition of value restrictions to the various feature parameters.
- The **Color Model** and **Color Component** classes are used for the representation of the color information, encoded in the form of the Y, Cb, Cr components of the MPEG color space.
- Class **Distribution** and **Distribution Parameter** represent information regarding the defined **Feature Parameter** models.
- Class **Motion Norm**: used to represent information regarding the object motion.
- Class **Algorithm**: the superclass of the available processing algorithms ( $A_1, A_2, \dots, A_n$ ) to be used during the analysis procedure. This class is linked to the **FeatureParameter** class through the *usesFeatureParameter* property in order to represent the potential argument list for each algorithm.
- Class **Detection**: used to model the detection process, which in our framework consists of two stages. The **CandidateRegionSelection** involves finding a set of regions which are potential matches for the object to be detected,



**Fig. 2.** Multimedia analysis ontology.

while **FinalRegionSelection** leads to the selection of only one region that best matches the criteria predefined for this object (e.g. size specifications).

- Class **Dependency**: this concept addresses the possibility that the detection of one object may depend on the detection of another, due to possible spatial or temporal interrelations between the two objects. For example in the Formula One domain, the detection of the car could be assisted and improved if the more dominant and characteristic region of road is detected first. In order to differentiate between the case where the detection of object  $O_1$  requires the detection of the candidate regions of object  $O_2$  and the case where the entire final region of object  $O_2$  is required, **PartialDependency** and **TotalDependency** are introduced.

As mentioned before, the choice of algorithms employed for the detection of each object is directly dependent on its available characteristic features. This association is determined by a set of properly defined rules represented in F-logic. F-logic is a language that enables both ontology representation and reasoning about concepts, relations and instances [15, 16].

The rules required for the presented approach are: rules to define the mapping between algorithms and features (which implicitly define the object detection steps), rules to determine algorithms input parameters, if any, and rules to deal



with object interdependencies as explained above. The rules defined for each category have the following form:

- “IF an object  $O$  has features  $F_1 \cap F_2 \cap \dots F_n$  as part of its qualitative description THEN algorithm  $A_1$  is a step for the detection of  $O$ .”
- “IF an object  $O$  has feature  $F$  AND  $O$  has algorithm  $A$  as detection step AND  $A$  uses feature  $F$  THEN  $A$  has as input the parameter values of  $F$ .”
- “IF an object  $O_1$  has partial dependency on object  $O_2$  AND object  $O_2$  has as **CandidateRegionSelection** part the set  $S = \{A_1, A_2, \dots, A_m\}$  THEN execute the set of algorithms included in  $S$  before proceeding with the detection of  $O_1$ .”
- IF an object  $O_1$  is totally dependent on object  $O_2$  THEN execute all detection steps for  $O_2$  before proceeding with the execution of  $O_1$  detection.”

In order for the described multimedia analysis ontology to be applied, a domain specific ontology is needed. This ontology provides the vocabulary and background knowledge of the domain i.e. the semantically significant concepts and the properties among them. In the context of video understanding it maps to the important objects, their qualitative and quantitative attributes and their interrelations.

### 3 Domain Knowledge Ontology

As previously mentioned, for the demonstration of the proposed approach the Formula One and Football domains were used. The detection of semantically significant objects, such as the road area and the cars in racing video for example, is an important step towards understanding and extracting the semantics of a temporal segment of the video by efficiently modelling the events captured in it. The set of features associated with each object comprises their definitions in terms of low-level features as used in the context of video analysis. The selection of the attributes to be included is based on their ability to act as distinctive features for the analysis to follow, i.e. the differences in their definitions indicate the different processing methods that should be employed for their identification. As a consequence, the definitions used for the Formula One domain are:

- **Car**: a motion homogeneous (i.e. comprising elementary parts characterized by similar motion), fully connected region whose motion norm must be above a minimum value and whose size can not exceed a predefined maximum value.
- **Road**: a color homogeneous, fully connected region, whose size has to exceed a predefined minimum value and additionally to be the largest such region in the video.
- **Grass**: a color homogeneous, partly connected region with the requirement that each of its components has a minimum predefined size.
- **Sand**: a color homogeneous, partly connected region with the requirement that each of its components has a size exceeding a predefined minimum.

In a similar fashion, the corresponding definitions for the Football domain include the concepts **Player**, **Field** and **Spectators** and their respective visual descriptions. As can be seen, the developed domain ontologies focus mainly on the representation of the object attributes and positional relations and in the current version does not include event definitions. For the same object, multiple instances of the **Color Model** class are supported, since the use of more than one color models for a single object may be advantageous in some cases.

### 3.1 Compressed-domain Video Processing and Rules

The proposed knowledge-based approach is applied to MPEG-2 compressed streams. The information used by the proposed algorithms is extracted from MPEG sequences during the decoding process. Specifically, the extracted color information is restricted to the DC coefficients of the macroblocks of I-frames, corresponding to the Y, Cb and Cr components of the MPEG color space. Additionally, motion vectors are extracted for the P-frames and are used for generating motion information for the I-frames via interpolation. P-frame motion vectors are also necessary for the temporal tracking in P-frames, of the objects detected in the I-frames [17].

The procedure for detecting the desired objects starts by performing a set of initial clusterings, using up to eight dominant colors in each frame to initialize a K-means algorithm. From the resulting mask, which contains a number of non-connected color-homogeneous regions, the non-connected semantic objects can be identified by color-model based selection. The application of a four connectivity component labelling algorithm results in a new mask featuring connected color-homogeneous components. The color-model-based selection of an area corresponding to a color-homogeneous semantic object is performed using a suitable mask and the Earth Movers Distance (EMD). EMD computes the distance between two distributions represented as signatures and is defined as the minimum amount of work needed to change one signature into the other. Additional requirements as imposed by the models represented in the ontology, are checked to lead to the desired object detection. For motion-homogeneous objects a similar process is followed. At first, a mask containing motion-homogeneous regions is generated. Subsequently, the model-based selection depends on the information contained in the ontology (e.g. size restrictions, motion requirements).

The construction of the domain specific rules derives directly from the aforementioned video processing methodology. For example, since color clustering is the first step for the detection of any of the three objects, a rule of the first category without any feature matching condition is used to add the k-means algorithm as the first detection step to all objects. A set of different algorithms could have been used as long as the respective instantiations are defined.

## 4 Experimental results

The proposed approach was tested in two different domains: the Formula One and the Football domain. In both cases, the exploitation of the knowledge con-

tained in the respective system ontology and the associated rules resulted to the application of the appropriate analysis algorithms using suitable parameter values, for the detection of the domain specific objects. For ontology creation the OntoEdit ontology engineering environment [18] was used, having F-logic as the output language. A variety of MPEG-2 videos of  $720 \times 576$  pixels were used for testing and evaluation of the knowledge assisted semantic annotation system.

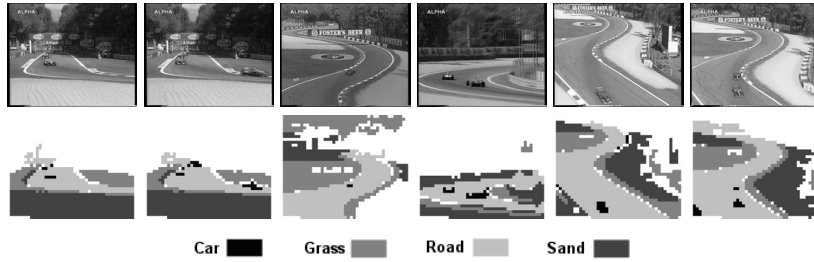
For the Formula One domain our approach was tested on a one-hour video. As was discussed in section 3, four objects were defined for this domain. For those objects whose homogeneity attribute is described in the ontology by the **Color Homogeneity** class, the corresponding color models were extracted from a training set of approximately 5 minutes of manually annotated Formula One video. Since we assume the model to be a Gaussian distribution for each one of the three components of the color space, the color models were calculated from the annotated regions of the training set accordingly. Results for the Formula One domain are presented both in terms of sample segmentation masks showing the different objects detected in the corresponding frames (Fig. 3) as well as numerical evaluation of the results over a ten-minute segment of the test set (Table. 1). For the Football domain, the proposed semantic analysis and annotation framework was tested on a half-hour video, following a procedure similar to the one illustrated for the Formula One domain. Segmentation masks for this domain are shown in Fig. 4, while numerical evaluation of the results over a ten-minute segment of the test set for this domain are given in Table. 1.

For the numerical evaluation, the semantic objects appearing on each I-frame were manually annotated and compared with the results produced by the proposed system. It is important to note that the regions depicted in the generated segmentation masks correspond to semantic concepts and this mapping is defined according to the domain specific knowledge (i.e. object models) provided in the ontology.

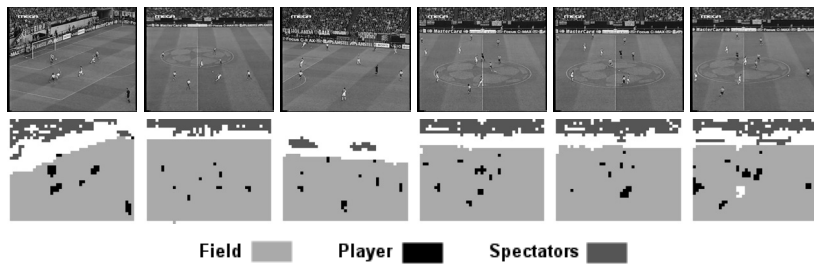
## 5 Conclusions

In this paper we have presented an ontology-based approach for knowledge assisted domain-specific semantic video analysis. Knowledge involves qualitative object attributes, quantitative low-level features generated by training as well as multimedia processing methods. The proposed approach aims at formulating a domain specific analysis model with the additional information provided by rules, appropriately defined to address the inherent algorithmic issues.

Future work includes the enhancement of the domain ontology with more complex model representations, including spatial and temporal relationships, and the definition of semantically important events in the domain of discourse. Further exploration of low-level multimedia features (e.g. use of the MPEG-7 standardized descriptors) is expected to lead to more accurate and thus efficient representations of semantic content. The above mentioned enhancements will allow more meaningful reasoning, thus improving the efficiency of multimedia content understanding. Another possibility under consideration is the use of a



**Fig. 3.** Results of road, car, grass and sand detection for Formula One video. Macroblocks identified as belonging to no one of these four classes are shown in white.



**Fig. 4.** Results of field, player, and spectators detection for Football video. Macroblocks identified as belonging to no one of these three classes are shown in white.

more expressive language, e.g. OWL, in order to capture a more realistic model of the specific domain semantics.

## References

1. S.-F. Chang. The holy grail of content-based media analysis. *IEEE Multimedia*, 9(2):6–10, Apr.–Jun. 2002.
2. S.-F. Chang, T. Sikora, and A. Puri. Overview of the MPEG-7 standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.
3. A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, Jan/Feb 1999.
4. P. Salembier and F. Marques. Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services. *IEEE Trans. Circuits and Systems for Video Technology*, 9(8):1147–1169, December 1999.
5. W. Al-Khatib, Y.F. Day, A. Ghafoor, and P.B. Berra. Semantic modeling and knowledge representation in multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):64–80, Jan/Feb 1999.
6. S. Little J. Hunter, J. Drennan. Realizing the hydrogen economy through semantic web technologies. *IEEE Intelligent Systems Journal - Special Issue on eScience*, 19:40–47, 2004.

**Table 1.** Semantic analysis results for the Formula One and Football domains

Object	correct detections	false detections	missed
Road	97%	2%	1%
Grass	87%	8%	5%
Sand	87%	9%	4%
Car	66%	27%	7%
Field	100%	0%	0%
Player	76%	5%	19%
Spectators	70%	2%	28%

7. A. Yoshitaka, S. Kishida, M. Hirakawa, and T. Ichikawa. Knowledge-assisted content-based retrieval for multimedia databases. *IEEE Multimedia*, 1(4):12–21, Winter 1994.
8. V. Mezaris, I. Kompatsiaris, and M.G. Strintzis. An Ontology Approach to Object-based Image Retrieval. In *Proc. IEEE Int. Conf. on Image Processing (ICIP03)*, Barcelona, Spain, Sept. 2003.
9. A.B. Benitez and S.F. Chang. Image Classification Using Multimedia Knowledge Networks. In *Proc. IEEE Int. Conf. on Image Processing (ICIP03)*, Barcelona, Spain, Sept. 2003.
10. R. Tansley, C. Bird, W. Hall, P. Lewis, and M. Weal. Automating the linking of content and concept. In *Proc. ACM Int. Multimedia Conf. and Exhibition (ACM MM-2000)*, Oct./Nov. 2000.
11. G. Tsechpenakis, G. Akrivas, G. Andreou, G. Stamou, and S.D. Kollias. Knowledge-Assisted Video Analysis and Object Detection. In *Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (Eunite02)*, Algarve, Portugal, September 2002.
12. M. Ramesh Naphade, I.V. Kozintsev, and T.S. Huang. A factor graph framework for semantic video indexing. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(1):40–52, Jan. 2002.
13. I. Kompatsiaris, V. Mezaris, and M. G. Strintzis. *Multimedia content indexing and retrieval using an object ontology*. Multimedia Content and Semantic Web - Methods, Standards and Tools, Editor G.Stamou, Wiley, New York, NY, 2004.
14. C. Town and D. Sinclair. A self-referential perceptual inference framework for video interpretation. In *Proceedings of the International Conference on Vision Systems*, volume 2626, pages 54–67, 2003.
15. J. Angele and G. Lausen. *Ontologies in F-logic*. International Handbooks on Information Systems. Springer, 2004.
16. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, 1995.
17. V. Mezaris, I. Kompatsiaris, N.V. Boulgouris, and M.G. Strintzis. Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(5):606–621, May 2004.
18. Y. Sure, J. Angele, and S. Staab. *OntoEdit: Guiding Ontology Development by Methodology and Inferencing*. Springer-Verlag, 2002.

# Catalog Search Engine: Semantics applied to products search

Jacques-Albert De Blasio, Takahiro Kawamura, and Tetsuo Hasegawa

Research and Development Center, Toshiba Corp.

**Abstract.** The Semantic Web introduces the need for semantic search engines. In this paper, we explain our vision of a catalog search engine for semantically defined products. With our prototype, we address the problem of products' information retrieval over the Internet and their semantic enrichment through the mixed usage of thesauruses and ontologies. We show how we automatically build a repository of instances of ontology classes, and how we dynamically prioritize the search variables of our engine. We then introduce our prototype which, through the use of all those concepts, improves the user experience.

## 1 Introduction

The Semantic Web introduces the need for semantic search engines. Although semantic search is already available in a variety of forms such as SHOE[1] or Ask Jeeves[2], semantic search for products sold on the Internet is rarely available. With the system we developed, we strived to fill this gap.

Products catalogs available on the Internet all have limitations of several types. They either provide a wide range of products but have a poor search engine in terms of precision, or offer a limited range of products with a powerful but too specialized (in terms of search variables) search engine. Whichever the catalog, the user can easily get frustrated by their poor ability to supply him/her precise results and an extensive selection of products at the same time.

One of the challenges of the semantic web is to transform the already available information into more meaningful, more usable data. A lot of the available literature tackles this problem and agrees on a fundamental problem: most of the difficulties come from the ambiguity of the human language, and from the fact that nearly all this information has been created by humans for human consumption. Products information, on the other hand, has the advantage of being, in most cases, based on an agreed vocabulary. However, the problem with products sold on the Internet is that the quality of their descriptions (in terms of availability), as well as the presentation of those descriptions (in structured tables, simple paragraphs, etc) varies greatly from a web site to another. Nonetheless, considering that the Internet is the biggest products database available, it becomes obvious that it should be the source of any search engine aspiring to be as complete and accurate as possible.

Our vision of a catalog search engine includes three distinct goals. The catalog must contain as much products as possible, its search engine must be the most

accurate, and the user must be given enough tools to let him/her search efficiently through the catalog. In this paper, we show that we answered to those three needs with the following strategies. The wideness of the catalog is insured by the gathering of existing products' information all over the Internet through the usage of dedicated parsers. The accuracy of the search engine is reached by a combination of semantic enrichment of the previously fetched information, and the automatic conversion into logic facts of every characteristic of the products. Eventually, the user's search efficiency is enhanced by the usage of algorithms dynamically computing the usefulness of each products' characteristic. Moreover, the usage of a thesaurus during the query phase allows the user not to worry about the exactitude of the content of his/her query.

In the following sections, we first introduce the architecture of our system. Then, we focus on its usage and explain the details of its features. We continue with a short demonstration of our prototype and follow with a discussion about decisions we took during the design phase. Eventually, we take a brief look at the existing work tackling the problem of catalog search engines and conclude.

## 2 Architecture of the Catalog Search Engine

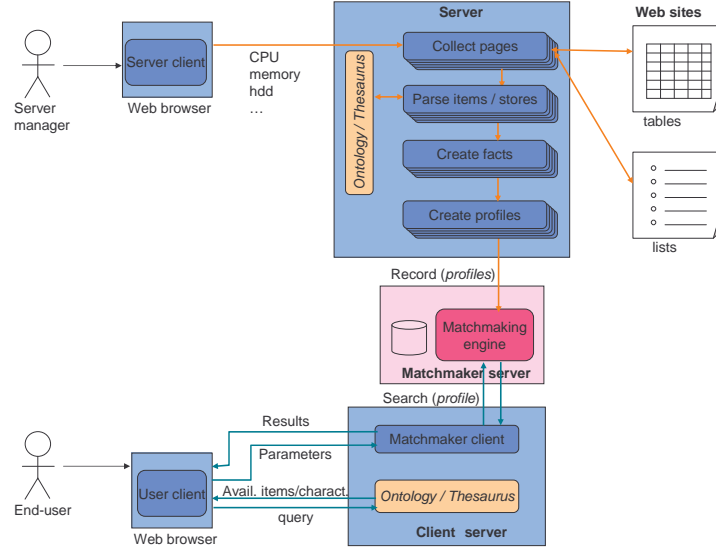
### 2.1 Overall Architecture

The architecture of the catalog search engine is shown in Fig. 1. The system we suggest is a complete solution, from the server fetching data from the Internet, to the client that will let the end-user carry out his/her requests. The server is made of 4 main components; the web page fetcher, the parser, the facts creator and the profiles creator. The client is separated into 2 main components; a GUI with which the end-user will communicate, and a proxy which will be in charge of the client-Matchmaker communication. In between the server and the client lies the Matchmaker server which provides the search capabilities.

The main idea of this system consists in fetching information about products sold on the Internet, and publishing this information on the Matchmaker server. This information will be enriched with semantics using ontologies. On the other side, the end-user will be able to express requests to the Matchmaker. The latter will browse through all the available advertisements, try to find those which are the most closely related to the request and eventually return them to the user. We will describe the usage in the next section.

At the heart of our prototype lies the Semantic Service Matchmaker, a service search engine based on the LARKS[3] algorithm. It adopts the filtering approach which uses sophisticated information retrieval mechanisms and ontology-based subsumption mechanisms to match requests against advertisements. This engine has already proven to be efficient in regard to web services matchmaking[4][5].

Ideally, when the requester looks for a product, the Matchmaker will retrieve a product that matches exactly the expected one. In practice, if the exact product is not available, the Matchmaker will retrieve one which capabilities are similar to those expected by the requester. Ultimately, the matching process is the result of the interaction of the products available, and the requirements of the requester.



**Fig. 1.** Architecture and flow of the system. Note that the “ontology / thesaurus” denotes the same instance on the server and client sides

Although the Matchmaker originally provides a set of 5 filters, our prototype uses only two of them. The *type* filter applies a set of subtype inferencing rules mainly based on structural algorithm to determine whether an ontology class is a subsumption of another. The *constraint* filter has the responsibility to verify whether the subsumption relationship for each of the constraints are logically valid. The Matchmaker computes the logical implication among constraints by using polynomial subsumption checking for Horn clauses. More details about the Matchmaker’s filters are provided in [4].

## 2.2 Usage scenario

### Server side

1. The server is initialized with a file which contains information concerning the web pages to be fetched and parsed. This file connects each type of products with a list of web pages (e.g `http://somewhere/hddidetosell.html`  $\Rightarrow$  products type “HDD IDE”).
2. Next, web pages are fetched from selected web sites. Once done, a parser detects relevant information from those web pages. If a new product is detected, the server automatically creates a new instance of the ontology class which describes the type of the product. If a new characteristic is detected, the server updates the list of properties associated with each class of product.
3. Then, the server automatically creates a file containing a list of facts written in RDF-RuleML[8][9]. Each fact corresponds to a characteristic of a product (see section 2.4).



4. Eventually, the server creates an “advertisement” profile for each product. A profile is the semantic description of a product (see section 2.5). Once all the profiles for all the products have been created, they are registered in the Matchmaker server.

### Client side

1. First the user inputs a query. This query is parsed and its content is compared with the thesaurus’ words, as well as with the name of the instances of the products’ types (created at step 2 of the server part).

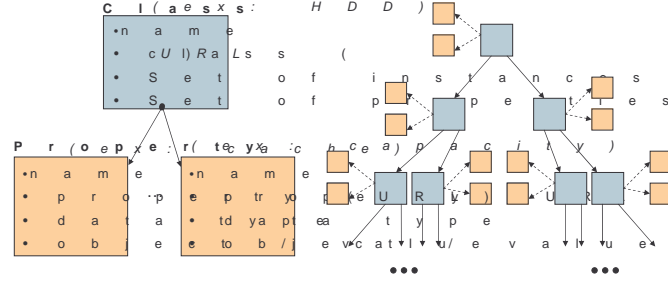
2. The answer to the query is either a list of types of products having the best matching terms to the query, or a list of instances, or both. If the answer is a list of instances, the user can click on one of them in order to display the details of the chosen product. If the answer is a list of types of products, the user can click on one of them in order to display a list of characteristics of the chosen type. If the user wants to carry out a fine search, he/she must input some values for the characteristics with which he/she wants the result to be relevant. When the user eventually clicks on the “finer search” button, a RDF-RuleML file containing those characteristics translated into facts is automatically created.

3. The system automatically creates a “request” profile. This profile is then submitted to the Matchmaker, which tries to match this “request” to the “advertisements” contained in its database. If one or more matching profiles are found, they are sent back to the user, who sees them as individual products. He/she can then click on one of the available link to different shops selling the product.

## 2.3 Usage of thesauruses and ontologies

Thesauruses are needed for any search engine of which search mechanism is not exclusively based on the query’s keywords. For the sake of simplicity, we built our own thesaurus for our prototype. While the goal is, of course, to use a rather complete thesaurus such as WordNet[6], we wanted our thesaurus to be multilingual, which WordNet is not. In our thesaurus,  $n$  terms can be synonyms of  $n$  other terms, and each term is translated into  $m$  languages. As the user can choose in which language he/she would like to interact with our prototype, it will set the language of the thesaurus. In the future we want to improve this by letting the user type the request in any language and let the search engine browse through the entire thesaurus, without any preference for the language.

Our ontologies are written in OWL[7]. An OWL class corresponds to a product type which characteristics are described using the OWL properties. Each property’s *range* is either an object (e.g. the type of interface of a hard disk) or a value (e.g. the capacity of a hard disk, in gigabytes). The Fig. 2 shows our internal representation of an ontology. The “datatype” attribute is needed in order to make sure that the value of a product’s characteristic taken from a web site (during the parsing) has the same type as the one expected.



**Fig. 2.** On the left: internal representation of the ontology. On the right: architecture of our ontologies (note that the relationships are not necessarily binaries).

In our system, we have a general ontology composed of classes corresponding to various types of products (e.g. “hdd ide”, “scanner”, etc). We assume that this ontology has been created prior to the launch of the system. When the server of the system browses and parses the information found on various web sites, it automatically fills in each instances slot for each class. For example, all the products found as being of type “hdd ide” will be automatically added to the instances slot of the product type (the class) “hdd ide”. The instances growth is non-monotonic. It makes sense in that products may disappear if the catalog of the sellers is updated. In our prototype, we track each product’s availability, so that if all the web sites of which we get information do not offer a product anymore, its instance is automatically deleted.

A problem we encountered was related to the names of the products. A product has, in general, the same name wherever it is sold. Even so, some web sites on the Internet mix the name and the characteristics of the products in the same string. The instances we record being based on the name, we had to insure that, even though the name may be different for a same product, new instances might not be created. To solve this problem, we first check the manufacturers of the products. If they match, we use a parser which is able to correct suspicious products’ names through string comparisons. Also, as some products may have different names in different countries, we use a thesaurus of products’ names to make sure that the products are the same.

## 2.4 Characteristics of the products

A product is distinguished by its characteristics. A hard disk has a certain seek time, capacity, interface, etc. Translating products’ characteristics into facts is quite natural, as each characteristic can be thought as a *truth* about the product it describes. The facts our system creates are all of the form  $P(x, y)$  where  $P$  is a predicate, and  $x$  and  $y$  two terms. The facts written by the server use two kinds of predicates, “equal” when the term  $y$  is a numerical value, and “is” when the term  $y$  is a string of characters. The facts written by the client use the predicates “is less than or equal to” and “is more than or equal to” when  $y$  is a numerical value, “is” when  $y$  is a string of characters. Each property of

the products' classes of our main ontology are converted into OWL classes and used as the term  $x$  of the facts (for a discussion about this conversion, refer to section 4). The facts concerning the manufacturer, the seller and the price of a product are considered the minimum information required for a product to be taken into account.

In the table 1, we show an example of a product's characteristics converted into facts on the server and the client side. On the server side, the characteristics of a hard disk fetched from the Internet are converted into facts. On the client side, characteristics which values have been input by the user are also converted into facts. "MANUFACTURER", "COST", "SELLER" and "CAPACITY" were OWL properties converted into classes. The predicates are classes of an ontology describing predicates. We use our own thesaurus to make the connections between the words used in the description of a characteristic (e.g. "sold at") and the corresponding ontology class (e.g. "COST"). See the code at section 2.5 for an example of a fact.

Server side	
Char. fetched from web sites	Facts
sold at <b>45,000</b> Yen	<i>equal</i> (COST, 45,000)
manufactured by <b>Toshiba</b>	<i>is</i> ( MANUFACTURER, Toshiba)
sold by <b>anotherMart</b>	<i>is</i> (SELLER, anotherMart)
a capacity of <b>80</b> Gb	<i>equal</i> (CAPACITY, 80)
Client side	
Char. with values	Facts
price $\leq$ <b>50,000</b>	<i>is less than or equal to</i> (PRICE, 50,000)
capacity $\geq$ <b>60</b>	<i>is more than or equal to</i> (CAPACITY, 60)

**Table 1.** Characteristics translated into facts - Server side

Once the client has transmitted the request profile (which contains links to the facts created by the client) to the Matchmaker, the latter will use its inference engine (called the *constraints* filter) to match the facts of the advertisements to the facts of the request.

## 2.5 Profiles

A profile is an OWL file containing a semantic description of a product, as well as a list of links to each fact present in the facts files related to this same product. For each shop selling the product, a profile is created (i.e. a product being sold by 10 shops will have 10 different profiles). The information stored in those profiles are the ontology class of the product's type, the name of the product (only if the profile is an advertisement), the list of facts and the URL to the shop selling the product. Once advertisements profiles have been registered to the Matchmaker, if a request profile is submitted the Matchmaker applies a matching using its *type* filter on the ontology class of the product's type and its *constraint* filter on all the facts. The following code shows an example of a profile and a fact.

```

<product:description rdf:id="Kakaku_CPU_Athlon_64_2800_Socket754_5">
  <product:name>ATHLON 64 2800 Socket754_5</productName>
  <product:restrictedTo rdf:resource="http://somewhere/onto.owl#cpu" />
  <product:constraint rdf:resource="http://somewhere/facts.rdf#clockspeed" />
  <product:constraint rdf:resource="http://somewhere/facts.rdf#cost" />
  <product:constraint rdf:resource="http://somewhere/facts.rdf#manufacturer" />
  ...
  <product:shopURL> http://www.aShopURL.com</product:shopURL>
</product:description>

<ruleml:Fact ruleml:label="cost">
  <ruleml:head>
    <ruleml:Atom ruleml:rel="http://somewhere/predicates.owl#numericallyEqual">
      <ruleml:args>
        <rdf:Seq>
          <rdf:li>
            <ruleml:Var ruleml:name="http://somewhere/store.owl#COST" />
          </rdf:li>
          <rdf:li>
            <ruleml:Ind ruleml:name="20990" />
          </rdf:li>
        </rdf:Seq>
      </ruleml:args>
    </ruleml:Atom>
  </ruleml:head>
</ruleml:Fact>

```

## 2.6 Dynamic update and prioritization of the characteristics

When the user searches for a given type of product, its related characteristics are displayed. If the user wants to carry out a fine search, he/she can insert some values to the characteristics he/she wants to be respected. The list of available characteristics is updated on the server side, when fetching and parsing information from various web sites. However, the priority in which those characteristics are shown to the user is dependant of each characteristic's associated weight. The weights are updated as follows.

- the more a characteristic is available for a given type of product, the greater its weight will be,
- the more a characteristic has possible values, the greater its weight will be (e.g. the size of a screen can be 15", 17", 19", 21", etc),
- the more a characteristic is chosen by the user, the greater its weight will be.

Other conditions come also into play to determine the position of a characteristic. As products' types are ontologically defined, we rely on the parent-child relationships to tell whether a product's characteristic should be shown before another. For instance, as "computer" is the parent class of "notebook computer", if the user searches for "notebook computers" its characteristics should be displayed prior to those of "computer".

## 3 The Prototype

We introduce here the prototype of the client application. Data from the Internet has already been fetched from Kakaku[10], a Japanese catalog web site, and parsed on the server side.

In Fig. 3(a), the user entered “hard disk” as a query. With the thesaurus, the client found out that “hard disk” is the equivalent to “hdd”. Moreover, using the ontology, the client proposes not only “hdd” but also “hdd ide” and “hdd scsi”, two subclasses of “hdd”. In Fig. 3(b), the user selected “hdd ide”. The client now proposes a list of characteristics of the “hdd ide”. The three first are always present for each product. The next ones are the characteristics available for “hdd ide”, as well as “hdd”, as well as any other parent class of the “hdd”, back up to the root of the ontology. The user decided to input values for two characteristics, the cost and the capacity. Once done, he/she gets the result shown in Fig. 3(c). The values corresponding to the chosen characteristics in the previous step are shown in bold. A link is provided to shops selling the products.

## 4 Discussion

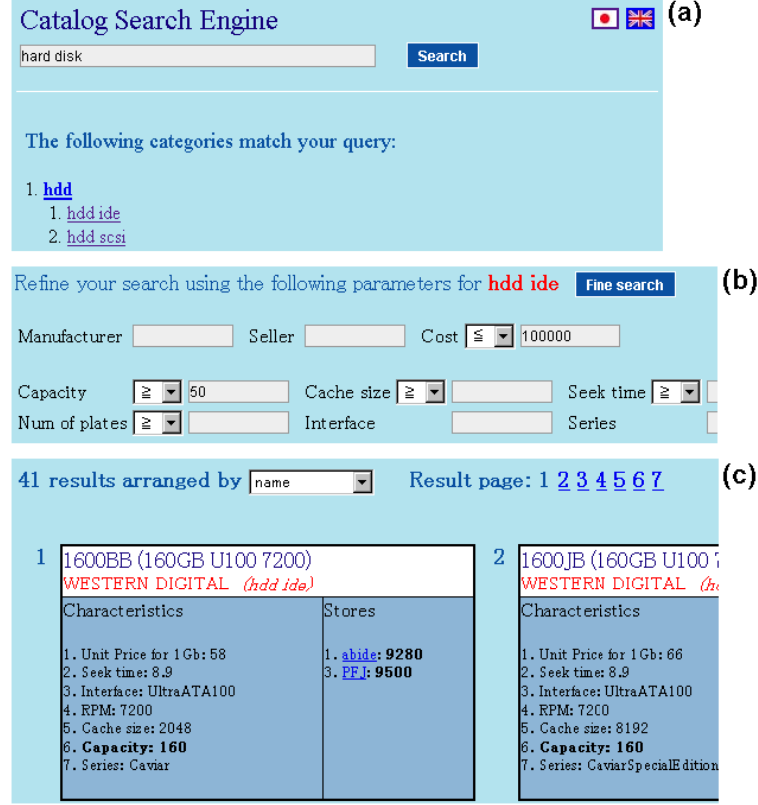
Our intention was not to produce a very efficient catalog search engine in terms of speed, but rather in terms of relevance of the results. In this regard, we reached the three goals cited in the introduction of this paper. As our system gathers data from various web sites, we get more details about the products than if we simply relied on specific vendors and thus insure the wideness of the catalog. Accuracy of the search engine is provided by the combined usage of thesauruses and ontologies, allowing the system to return very precise results even if the query of the user is relatively vague. Eventually, search efficiency is attained by handling the feedback of the user regarding the characteristics of the products. As a consequence of all this, we observed that a user needs about half the number of clicks than usually needed when accessing the same information about products on other web sites such as Kakaku.

However, as our system is still at the stage of a prototype, it is not without flaws. In fact, the parser approach to the problem of information fetching on various web sites can prove to be quite weak in the long term. It requires much more advanced techniques to be able to fetch facts or rules from web pages such as Amazon or Yahoo Shopping as those web sites do not always display information about products in a very formal way.

The reader may wonder why we chose to create facts using RDF-RuleML to describe the characteristics of each products, instead of directly using the properties of each products’ classes of our ontology. The reason is that we intend to create a much more powerful search engine, which does more than giving the possibility to enter a value for each characteristic. The goal is to use a better Natural Language Processing tool during the parsing phase on the server side, so that the system becomes able to create rules such as “*if the credit card is Visa or American Express, the customer can have a 5% discount*”. This kind of rule can not be expressed with OWL’s classes and properties.

Alternatively, we thought about expressing the facts and rules in SWRL[11] instead of RDF-RuleML. The advantage is that SWRL allows the use of properties which have been created in the ontology, and thus avoids redundancy. However, the terms of an atom in SWRL must be either variables, OWL indi-

viduals or OWL data values. Unfortunately, as individuals lack any subsumption relationship, the constraint filter of the Matchmaker would not work efficiently.



**(a) Catalog Search Engine**

Search input:

The following categories match your query:

- [hdd](#)
  - [hdd ide](#)
  - [hdd scsi](#)

**(b) Refine your search using the following parameters for **hdd ide****

Manufacturer:  Seller:  Cost:

Capacity:   Cache size:   Seek time:

Num of plates:   Interface:  Series:

**(c) 41 results arranged by  Result page: 1 2 3 4 5 6 7**

1	2																								
<b>1600BB (160GB U100 7200)</b> <b>WESTERN DIGITAL (hdd ide)</b>	<b>1600JB (160GB U100 7200)</b> <b>WESTERN DIGITAL (hdd ide)</b>																								
<table border="1"> <thead> <tr> <th>Characteristics</th> <th>Stores</th> </tr> </thead> <tbody> <tr> <td>1. Unit Price for 1 Gb: 58</td> <td>1. <a href="#">ahide</a>: <b>9280</b></td> </tr> <tr> <td>2. Seek time: 8.9</td> <td>3. <a href="#">PEJ</a>: <b>9500</b></td> </tr> <tr> <td>3. Interface: UltraATA100</td> <td></td> </tr> <tr> <td>4. RPM: 7200</td> <td></td> </tr> <tr> <td>5. Cache size: 2048</td> <td></td> </tr> <tr> <td>6. <b>Capacity: 160</b></td> <td></td> </tr> <tr> <td>7. Series: Caviar</td> <td></td> </tr> </tbody> </table>	Characteristics	Stores	1. Unit Price for 1 Gb: 58	1. <a href="#">ahide</a> : <b>9280</b>	2. Seek time: 8.9	3. <a href="#">PEJ</a> : <b>9500</b>	3. Interface: UltraATA100		4. RPM: 7200		5. Cache size: 2048		6. <b>Capacity: 160</b>		7. Series: Caviar		<table border="1"> <thead> <tr> <th>Characteristics</th> </tr> </thead> <tbody> <tr> <td>1. Unit Price for 1 Gb: 66</td> </tr> <tr> <td>2. Seek time: 8.9</td> </tr> <tr> <td>3. Interface: UltraATA100</td> </tr> <tr> <td>4. RPM: 7200</td> </tr> <tr> <td>5. Cache size: 8192</td> </tr> <tr> <td>6. <b>Capacity: 160</b></td> </tr> <tr> <td>7. Series: CaviarSpecialEdition</td> </tr> </tbody> </table>	Characteristics	1. Unit Price for 1 Gb: 66	2. Seek time: 8.9	3. Interface: UltraATA100	4. RPM: 7200	5. Cache size: 8192	6. <b>Capacity: 160</b>	7. Series: CaviarSpecialEdition
Characteristics	Stores																								
1. Unit Price for 1 Gb: 58	1. <a href="#">ahide</a> : <b>9280</b>																								
2. Seek time: 8.9	3. <a href="#">PEJ</a> : <b>9500</b>																								
3. Interface: UltraATA100																									
4. RPM: 7200																									
5. Cache size: 2048																									
6. <b>Capacity: 160</b>																									
7. Series: Caviar																									
Characteristics																									
1. Unit Price for 1 Gb: 66																									
2. Seek time: 8.9																									
3. Interface: UltraATA100																									
4. RPM: 7200																									
5. Cache size: 8192																									
6. <b>Capacity: 160</b>																									
7. Series: CaviarSpecialEdition																									

**Fig. 3.** Prototype screenshots. Three steps to search for products (cropped images)

## 5 Related Work

Froogle, a twin of Google in a shopping search engine point of view, offers a wide catalog and blatant speed, but allows search refinement only through price range. Kakaku gives the possibility to search using the products' characteristics, but the number of the latter is static. Both search engines get the products information directly from the vendors. Although it insures accuracy, this method limits greatly the number of sources of information. Amazon is too restrictive in terms of products, as they propose only those which they sell. To our knowledge, none of the web sites cited above make use of semantics.

The IWebS project[12][13] aims at creating an intelligent yellow pages service with semantically annotated services. Although they share some similarities with

our approach, they introduce the needs for manual annotations which would be intolerable for a database of thousands of different products.

Active Catalog[14] focuses on how retrieved information can be used to engineer parts and physical objects. Its database is entirely built beforehand, that is, there is no dynamic data acquisition. The parts' characteristics are also all predetermined. Eventually, the content as well as the usage makes it usable exclusively to engineers.

## 6 Conclusion

Based on the Matchmaker, we developed a prototype of a catalog search engine which enables users to have more accurate results in regard to their queries. Search parameters are dynamically updated through the analysis of fetched information and the feedback from the users. We showed that the approach of fetching available products data from the Internet, adding semantic to it through the use of ontologies, and efficiently searching through it is feasible. Using rules, our system will be able to give more expressive power to users' queries.

## References

1. Simple HTML Ontology Extensions, <http://www.cs.umd.edu/projects/plus/SHOE/>
2. Ask Jeeves, <http://www.ask.com/>.
3. K. Sycara, S. Widoff, M. Klusch, J. Lu, "LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace", In *Autonomous Agents and Multi-Agent Systems*, Vol.5, pp.173-203, 2002.
4. M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara, "Semantic Matching of Web Services Capabilities", *Proceedings of First International Semantic Web Conference (ISWC 2002)*, IEEE, pp. 333-347, 2002.
5. T. Kawamura, J. D. Blasio, T. Hasegawa, M. Paolucci, K. Sycara, "Public Deployment of Semantic Service Matchmaker with UDDI Business Registry", *Proceedings of 3rd International Semantic Web Conference (ISWC 2004)*, 2004. to appear.
6. WordNet, <http://www.cogsci.princeton.edu/wn/>
7. Web Ontology Language, <http://www.w3.org/TR/owl-ref/>.
8. Resource Description Framework, <http://www.w3.org/RDF/>.
9. RuleML, <http://www.ruleml.org/>.
10. Kakaku, <http://www.kakaku.com>.
11. Semantic Web Rule Language, <http://www.w3.org/Submission/SWRL/>.
12. M. Laukkanen, K. Viljanen, M. Apiola, P. Lindgren, and E. Hyvonen. "Towards Ontology-Based Yellow Page Services". In *Proceedings of the WWW 2004 Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, New York, USA, May 18th 2004.
13. E. Hyvonen, K. Viljanen, A. Hatinen. "Yellow pages on the semantic Web". Towards the Semantic Web and Web services. In *Proceedings of XML Conference*, Finland 2002.
14. S.R. Ling, J. Kim, P. Will, and P. Luo, "Active Catalog: Searching and Using Catalog Information in Internet-Based Design," *Proceedings of DETC '97 - 1997 ASME Design Engineering Technical Conferences*, Sacramento, California, September 14-17, 1997.

# Social Annotation of Semantically Heterogeneous Knowledge

Matthias Nickles<sup>1</sup>, Tina Froehner<sup>2</sup>, and Gerhard Weiss<sup>1</sup>

<sup>1</sup> Artificial Intelligence/Cognition Group, Department of Computer Science,  
Technical University Munich (TUM)  
D-85748 Garching b. München, Germany, {nickles, weissg}@cs.tum.edu

<sup>2</sup> Research Center Knowledge Management (RCKM),  
University of Applied Sciences Cologne  
D-50678 Köln, Germany, tina.froehner@fh-koeln.de

**Abstract.** An important kind of tacit knowledge in the context of the Semantic Web are the *social communication structures* among heterogeneous knowledge sources and users. Communication structures heavily influence the way knowledge is generated and used, because in a context of distributed and autonomous information sources like in the Semantic Web, knowledge is constituted and adapted pragmatically through possibly conflictive communication processes. As a way to set social structures in relation to distributively acquired knowledge, this work proposes Open Ontologies and Open Knowledge Bases for the annotation of (first-level) knowledge with emergent social meta-data (*social reification*). Whereas traditional approaches to knowledge and ontology integration emphasize the consensus finding among the participants, Open Ontologies and Open Knowledge Bases explicitly model semantical heterogeneity in multiple levels of complexity reduction, and allow the probabilistic weighting of inconsistent knowledge resulting from their assertive weight in their communicative context.

Keywords: Semantic Web, Semantic Knowledge Annotation, Emergent Semantics, Ontologies, Social Data Mining, Computational Autonomy

## 1 Introduction

The Semantic Web can be seen as the most important effort toward large scale knowledge building and sharing in an open information environment. Decisive for the success of this long-term task is the provision of formalisms and mechanisms for the communication (i.e. symbolic interaction) of a very large number of distributed, autonomous knowledge sources and users. Shared ontologies and knowledge bases play a crucial role in this scenario, since they enable such communication, and knowledge acquisition among autonomous information sources is basically a communicative act.

Traditional approaches to the modeling and acquisition of ontologies and instance knowledge have several shortcomings in this respect as they seldom handle meaning dynamics, they seldom consider knowledge as being contextualized with intentions, processes and effects from the “outside world”, and they usually have no concept for the treatment of semantic heterogeneity (e.g. resulting from contradictions) that does not result



in a loss of information. Whereas approaches like *Emergent Semantics* [1], *Dynamic Ontologies* [2] and semantical ontology merging and alignment have caused significant improvements regarding some of these problems, semantical inconsistencies due to conflicting knowledge sources are almost always still taken for something which either should be avoided, or should be homogenized using, e.g. clustering techniques, or should be filtered out (e.g., using criteria like (dis-)trust or source reputation [5]). In demarcation from such views, it should be recognized, that semantical inconsistencies are not just unfavorable states, but that they are in real-world environments often unpreventable due to stable belief or goal conflicts [3] of knowledge sources, that they can even provide the knowledge user with valuable meta-information about the intentions, goals and social relations among the knowledge sources, and, if they have been made explicit and visible, that they can be prerequisites for a subsequent conflict resolution. In general, in the absence of a normative meaning governance, mechanisms for knowledge integration can only be a preliminary decision about the reasonable modeling of communicated knowledge artifacts, because within a heterogeneous group of autonomous knowledge sources and users, in the end each user can only decide for himself about the relevance and correctness of the given information, which provides a strong argument for the conservation of knowledge heterogeneity while integrating.

With this work we propose *Open Ontologies* and *Open Knowledge Bases* as a general approach to the *social* acquisition and annotation of knowledge for open environments like the Semantic Web (but also, e.g., for open P2P systems and Semantic Grids). It is primarily meant to introduce a fundamentally novel perspective rather than providing technical specifications.

## 2 Towards a Socially-Aware Semantic Web: Knowledge as a result of controversial mass communication

The Semantic Web has several key characteristics that make the acquisition and representation of knowledge complicate in contrast to closed systems and applications:

**Openness** Access, number and contributions of information sources are unrestricted for its major part.

**Opaqueness of knowledge sources** The intentions of knowledge providers are more or less unknown and their trustability and reliability cannot be guaranteed.

**Opaqueness of users** The impact of a knowledge contribution to the Semantic Web on its users is often hard to predict.

**High dynamics and complexity** There are very large, heterogeneous and fluctuating amounts of knowledge sources, knowledge contributions and users.

**Highly controversial** Several domains of web knowledge are highly controversial, e.g. in regard to politics, culture and product assessments by consumers. It seems to be extremely unlikely that such fundamentally divergent world views can be homogenized even in regard to general ontological concepts in the foreseeable future. Thus, semantic inconsistency is a reality knowledge management must cope with.

**No authoritative background knowledge** Decentralized structures and different background knowledge lead to a high diversity of individual knowledge.

**Missing process knowledge** Currently, the representation of machine accessible knowledge focusses on “knowledge end-products”, not on the representation of processes that generate, modify or use knowledge.

These issues have in common that they rise mainly from the *autonomy* and *proactivity* of knowledge sources and users, being black- or gray-box actors with more or less opaque goals they pursue asserting or forming their individual world views. The way such autonomous entities (conceptually captured in the notation of *information agents* in this work) exchange information is *communication*. Although truly intelligent information agents are not expected to be widely spread on the internet in the foreseeable future, web knowledge can already be considered as communicative, because it is generated in order to influence its recipients and its intentionality and reliability is often unknown. This is even true if knowledge is communicated indirectly, tacitly or asynchronously using e.g. static web sites. Web knowledge is also contextualized with other web knowledge, and it can be agreed as well as denied by other knowledge facets (respectively their sources). Therefore, it appears to be reasonable to consider the Semantic Web as a very large, heterogeneous and hybrid system of interacting information agents (including humans), where information provided by humans and computationally generated knowledge co-exist. Due to the highly distributed character and the heterogeneity of this partially “wild grown” multiagent system, besides agreed protocols and formalisms, shared ontologies and knowledge bases are expected to be extremely useful to enable and improve mutual understanding and interactivity. Because knowledge on the Semantic Web is not only required in order to improve communication, but, maybe even more important, is an emergent outcome and constituent of communication, the key properties of communication need to be taken into account when it comes to building such ontologies and knowledge bases. Thus, viewing the Semantic Web as a system of directly or indirectly communicating information agents, we propose a communication-oriented paradigm, which has several implications for the retrieval and modeling of distributed knowledge. Most important, knowledge management for the Semantic Web needs to cope with the fact that the meaning of information on the web can never be determined for sure in general, might change, and might be constituted from the possibly conflicting opinions of large sets of knowledge sources. The primary goal of Open Ontologies and Open Knowledge Bases is to make the knowledge contributions of large, fluctuating and possibly conflicting sets of autonomous sources usable in a computational sense, i.e. to provide computationally accessible meta-data to the users even if such socially accumulated knowledge is inconsistent or unreliable (especially in the absence of trustability). For this purpose, the *social layer of knowledge* on the web needs to be found and made explicit by means of semantic annotation to the web users. In particular, the technical openness of shared knowledge like ontologies and the comparability of distributed, local knowledge needs to be improved, knowledge artifacts need to be interpretable as parts of *communication processes* (with induced relationships like assertion, agreement, contradiction, request, revision, specialization, generalization...), and the complexity of socially accumulated knowledge needs to be reduced *without* the need to come to a consent among the participants and with as less loss of information about social heterogeneity as possible. Largely neglecting these aspects, most of the current efforts in order to build the Se-

semantic Web concentrate on the specification of languages and tools for the modeling of agreed, homogeneous knowledge, and research is just beginning to take into consideration phenomena like the social (i.e. communicative) impact of resource descriptions, conflicting opinions, information biased by e.g. competing commercial or political interests, and inconsistent or intentionally incorrect information. Bringing information (e.g. via web sites or web services) into the web is in fact a social act, and the relationship between informational artifacts on the web is communicative (i.e. specifying, agreeing, contradicting...). This can of course produce intentional and unavoidable inconsistencies (e.g. company interests versus customer interests or various conceptualizations due to differences in culture). If these are ignored, or filtered out, ranked/recommended or homogenized too early (e.g. applying trust), important information for the user or the application might be lost. In order to make this important information available, we propose the following:

- Knowledge facets on the web like meta-data annotating web pages must be seen as *subjective belief assertions* of rational intelligent black-box agents (artificial agents as well as human users). They are created with certain intentions which are more or less hidden and are situated within action processes in order to make the successful assertion of this particular “truth” more likely (with advertisement as the most usual case, but also e.g. user recommendations regarding products and political statements, and even lexicon entries).
- Knowledge heterogeneity needs to be made *explicit*. Since knowledge sources are more or less opaque with hidden belief and goals, the need for instruments that enable the comparison of different standpoints becomes more important for knowledge users.
- Knowledge heterogeneity needs to be *explained*. Publication of knowledge on the web is an assertive act that is embedded within a pragmatical context of reasons and implications. In fact, the meaning of knowledge cannot be determined without considering this pragmatical context [8].
- The representation of web knowledge has to comprise *uncertainty* on the social level. Knowledge assertions uttered from black- or gray-box agents are basically more or less unreliable, and they might be misleading. One way to ensure reliability is the establishment of trust relationships. But to establish trust, one has to accumulate experiences and weigh different opinions. In addition, heterogeneous knowledge contributions of large numbers of agents need to be generalized using stochastic methods in order to reduce their complexity and to make practical use of them (e.g. to derive average opinions). From the viewpoint of a knowledge consumer, even though someone cannot say how things “are” in reality, a knowledge base must provide an approximate value for her decision finding.

Whereas it is already widely agreed that the statements of human individuals can only be transferred to machine understandability with a more or less degree of uncertainty, the need for the use of probabilistic and approximate representation formalisms in order to model collectively constituted knowledge on the web is still largely neglected.

Figure 1 shows the semantical levels proposed by Tim Berners-Lee for the structure of the forthcoming Semantic Web, with extensions (red/light gray font) we recommend

for some aspects of this concept in response to the mentioned issues. In particular, it appears to be inevitable to us to provide formalisms and calculi that explicitly consider semantically heterogeneous meta-data like resource descriptions and ontologies created from the contributions of multiple sources that compete for the assertion of their individual “truths” and interests. Of course, the Semantic Web is already open, but for a broad acceptance and to provide value to its users, we strongly suppose that communicative (i.e. social) relationships among closed “islands” of knowledge like contradiction or agreement need to be made explicit formally and technically as part of the layers of a “socially-aware” Semantic Web, using a concept called *social reification* (cf. next section). In this regard, the empirical derivation and stochastic modeling of open meta-data seems inevitable if the set of knowledge sources is either very large, or fluctuates, or generates indefinite information.

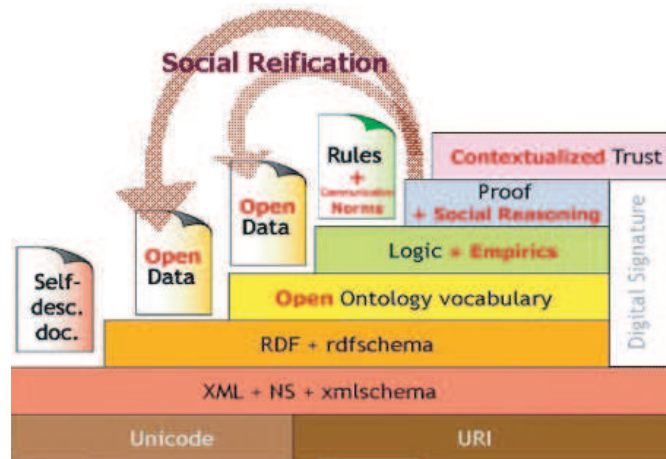


Fig. 1. A socially-aware Semantic Web

### 3 Open Ontologies and Open Knowledge Bases

#### 3.1 Characteristics

Formal ontologies and knowledge bases are traditionally defined as agreed descriptions of certain domains which serve as common ground for distributed tasks like knowledge exchange, modeling and user information. This understanding leads to difficulties if the informational input these media are build from is likely to be intentionally inconsistent, and there either does not yet exist enough meta-knowledge like trust to identify and filter out “inappropriate” or “wrong” data a priori, or there does not even exist a concept of global inappropriateness or correctness at all. On the other hand, sound and agreed ontologies are doubtless an inevitable prerequisite for efficient knowledge creation, representation and exchange, whereby we consider implicit and emerged ontologies and schemata (e.g. in the context of semi-structured data modeling) to be

such ontologies too. Of course, ontological heterogeneity can be overcome by means of techniques like the renaming of inconsistent concepts, and in general, inconsistent knowledge can be made consistent providing appropriate *truth contexts* [10]. However, such solutions often generate redundancy instead of an informational benefit for the knowledge users, or lead to difficulties finding other than trivial annotations like “In the belief of agent x, the following is true:...”. OO&OKB aim at the solution for this dilemma by embedding conceptual knowledge facets gained from a heterogeneous set of self-interested autonomous knowledge sources (e.g. information agents or humans) within contextual information about their communicative (i.e. social) origin, impact, and relationships (e.g., contradiction, approval, revision or specification) to other communicated knowledge facets (which can be communicated by means of formal communication languages, but also be derived from, e.g., structured, semi-structured or natural language documents) and their sources. Doing so, in OO&OKB, knowledge as it can be found in conventional knowledge or ontology bases, is *lifted* to the social level and thus to a level where the sources and the users of the ontology are likely to achieve an agreement with the *social assessments* of possibly inconsistent and uncertain facts (e.g., if *agent*<sub>1</sub> contradicts *agent*<sub>2</sub>, both usually agree that they do so!). The judgement of assessed facts is then a subsequent task based on rich social knowledge instead of binary distinctions like to trust or not to trust particular agents. *OO&OKB are thus dynamic communication media which receive their content from the communication of multiple autonomous information sources and users, and provide a dynamic representation of socially annotated heterogeneous knowledge.*

Communication is here not so much to be understood as the exchange of symbols with a fixed meaning, but the other way round as a means to generate supra-individual meaning from interrelated interactions among black- or gray-box agents (i.e., agents with more or less unknown internal states, cognition and goals). The practical consequences arising from this are that OO&OKB need to be continuously adapted to new information, and the processes of creation, contextualization and interpretation of knowledge are integral aspects of OO&OKB themselves. In addition, communication among multiple agents likely requires mechanisms for the generalization of emergent meaning, since otherwise the complexity would grow too large due to the sheer number of individual knowledge contributions. Generalization is also a way to make OO&OKB look like homogeneous ontologies or knowledge bases if necessary, because at its highest level, generalization causes semantical homogenization among contradicting knowledge sources. Summing it up, Open Ontologies and Open Knowledge Bases have the following characteristics:

**Openness** No (or as few as possible) initial assumptions are made regarding the benevolence, trustworthiness, relevance, informedness and cooperativeness of its sources. Nevertheless, information about e.g. (dis-)trust and knowledge (un-)reliability is likely derivable from Open Ontologies and Open Knowledge Bases, since these are special cases of social structures.

**Dynamical derivation from communication** OO&OKB are emergent from and evolving with ongoing communication (e.g. agent interaction, but also asynchronous, indirect or tacit communication e.g. via the semantically interrelated contents of web sites) of knowledge sources and knowledge users to assert (deny, specify...) information and to express and specify informational needs and expectations. Social

background knowledge (existing social structures like laws) can be included in the derivation process.

**Explicitness and social annotation of semantical heterogeneity** OO&OKB *maintain* semantical inconsistencies arising from contradictions and conflicts, and contain (consistent) annotations of (conceptual or instance) knowledge with meta-information about its *social meaning* within the course of communication.

This concept is related to context logic [10], but in contrast does not aim for the provision of logical truth contexts. Rather, social annotations state the sound social meaning of subjective statements without judging them as true or false.

**Multiple, probabilistically modeled levels of social generalization** They allow multiple, application-dependant levels of generalization of social concepts (like the generalization of single information agents as *agent roles* or groups, allowing to derive “average” or shared group opinions from the communications of multiple knowledge sources), weighting the degree of inconsistency and the degree of details of the annotating meta-information (cf. section 4). Generalization can also help to overcome privacy issues by averaging individual information contributions.

### 3.2 Social Reification

OO&OKB contain as first-order objects knowledge facets that have the form 1st-level knowledge  $\leftarrow$  2nd-level knowledge, where 1st-level knowledge partially describes a domain concept in the same way as within usual ontologies (or instances of such concepts, respectively, for Open Knowledge Bases), but probably in an inconsistent way regarding other 1st-level knowledge in the same ontology. Since Open Ontologies are primarily an abstract meta-concept build upon conventional approaches for the representation of conceptual knowledge, we do not constrain or specify the sort of concrete entities that are to be “wrapped” within an Open Ontology (Open Knowledge Base) or at the content level of agent messages, like first-order logical statements, classes or frames. For the same reason, we do also not make any assumptions relating to ontology domains or concrete areas of application here. In contrast to 1st-level knowledge, 2nd-level knowledge (also called *social knowledge*) depicts the social context of 1st-level knowledge, the latter taken as generated from a communication act of an autonomous source of knowledge. This kind of annotation of 1st-level knowledge with 2nd-level knowledge we call *social reification*. A quite trivial kind of social reification is *quoting* (e.g., ‘Sue says: “...”’), but in general, all kind of information which describes how and to what effect certain data is produced within a process of communication can be informally understood as 2nd-level knowledge (and, of course, we can apply social reification recursively, i.e. annotate 2nd-level knowledge with 3rd-level knowledge as in ‘Sue says: ‘Tom says: “...”’ and so on). The most elementary forms of such social meta-data are considered agent speech act types like assertion, denial or query, inducing relations among single communication like ‘Sue contradicts Tom’s statement saying “...”’ and rich 2nd-level knowledge types such as knowledge source and user profiles and even complex social systems like organizations. In an empirical communication model [8] symbolic communicative acts gain their semantics from their expected effect on the subsequent trajectory of communications, which can be learned empirically from past

interactions (although we recommend empirical semantics to disregard mentalistic details which are unknown for autonomous agents and allow for the handling of uncertain meanings, the usage of such a semantics is not required to define an Open Ontology or an Open Knowledge Base). Because meaning is contextualized by the situation (history) of the respective act occurrence, in general 2nd-level knowledge describes communication processes (this applies even to simple quotations: In Sue says: "...", "Sue" is in fact just an abbreviation for the pragmatic impact utterances from Sue are expected to have. This concept is not meant to be a replacement for the usage of e.g. first-order predicate logic for Web reasoning, but instead as a completion which could be introduced gradually. E.g., the Resource Description Framework *RDF(S)* and *Notation3* already have elementary reification capabilities, which could be used for elementary social annotations (e.g. collective rating of RDF statements) as described in [6, 7], but would require an appropriate specification of this kind of usage. In the following, we will outline a more ambitious approach to this issue.

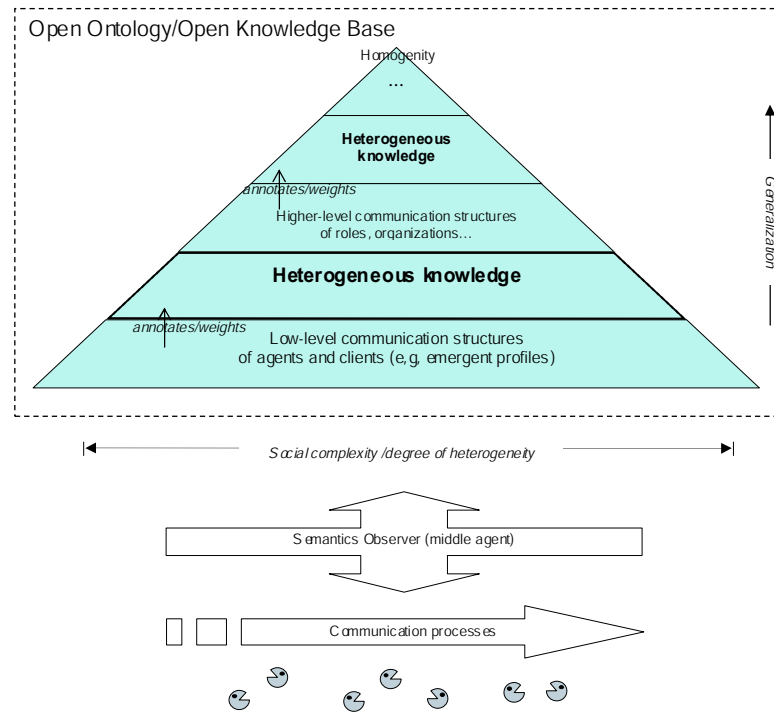
#### 4 Derivation of Open Ontologies and Open Knowledge Bases

Open Ontologies and Open Knowledge Bases need to be learned from the observation of communication processes. The technical requirements for this learning process are:

- information agents or other knowledge sources (e.g. peers in a P2P network, or passive resources like web documents) able to communicate and query 1st-level knowledge facets. In case of software agents, this can be done by means of a formal agent communication language (since OO&OKB do not require agent cooperativeness, speech act performatives used for collaboration like negotiation are not required, although they would be useful).
- a facility for the acquisition of OO&OKB from the observation of above communications, e.g., a dedicated middle agent within the infrastructures of the respective application, called a *semantics observer* (cf. figure 2).
- optionally, a pre-defined content of the Open Ontology or Open Knowledge Base, in order to speed up the learning process of the semantics observer, and to avoid the bootstrapping problem known from e.g. recommender systems, or to set static social structures like norms
- a facility for the low-level storage and querying of persistent knowledge (e.g., a database management system).
- optionally, a facility for the social reasoning upon the 2nd-level knowledge within the Open Ontology or Open Knowledge Base. respectively (to deduce new facts like "Sue is likely to contradict or specify Toms information", but also to derive trust relationships among the participants subsequently. Here, known techniques as described in e.g. [5] can be used).

The acquisition of OO&OKB comprises the following main tasks, which have to be performed in a loop as a continuous, incremental learning process for the whole period of agent communication (please find details in [9]).

1. Observation of communication. In addition, implicit or tacit communication might needs to be made explicit beforehand.



**Fig. 2.** Emergence and Generalization of Open Ontologies and Open Knowledge Bases

2. Derivation and/or adaptation of 2nd-level knowledge according to the respective semantical model (e.g. empirically)
3. Stochastic generalization of 2nd-level knowledge
4. Social reification and generalization of 1st-level knowledge
5. Alignment with given, obligatory 1st-level knowledge (e.g. a normative top-level ontology) or normative 2nd-level knowledge (e.g. laws preventing certain utterance of certain information), if necessary.

As mentioned earlier, OO&OKB also require the generalization of meaning in order to reduce their complexity (cf. figure 2). Generalization as a task in this sense has two steps: 1) the merging of 2nd-level knowledge, 2) the subsequent merging of related 1st-level knowledge facets. Typically, 1) comprises the merging of similar social processes to interactions patterns, and the combination of multiple similar behaving agents to social groups or social roles. After applying such generalization rules to 2nd-level knowledge, the annotated 1st-level knowledge needs to be merged accordingly. If, for example, multiple agents forming a single social group make inconsistent assertions, within the Open Ontology (Open Knowledge Base) each of these assertions obtains a probabilistic weight expressing the degree of expected approval this assertions gets from the role or group as a whole (calculated, e.g., from the frequency this assertion



has been uttered by different agents within this role or group) [7, 6]. We propose the usefulness of a co-presence of multiple levels of generalization, tailored to the desired levels of heterogeneity of the respective Open Ontology or Open Knowledge Base (cf. figure 2). Of course, the concrete representation and degree of heterogeneity that should be maintained strongly depends from application and user needs.

## 5 Conclusion

There is an obvious and rapidly growing need for knowledge-based systems capable of running in open environments like the Semantic Web with autonomous knowledge sources and users, given the increasing inter-operability and inter-connectivity among computing platforms. On the one hand, knowledge bases and ontologies should provide a stable ground for user information, agent and user communication and subsequent knowledge modeling, on the other hand, in open environments concept descriptions tend to be semantically inconsistent, they emerges from a possibly very large number of competing subjective beliefs and goals, and a priori there might be no such thing as a commonly agreed “truth” (in the “real world”, not even a discursive trend towards such a thing can be assumed). To cope with these two contradictory aspects must be a core concern of the communication-oriented paradigm of knowledge modeling and management, and is the basic motivation underlying the work described here. To this end, we have proposed Open Ontologies and Open Knowledge Bases as a fundamental step towards the modeling and representation of socially-induced knowledge heterogeneity for the Semantic Web.

## References

1. A. Maedche, F. Nack, S. Santini, S. Staab, L. Steels. Emergent Semantics. IEEE Intelligent Systems, Trends & Controversies, 17(2), 2002.
2. J. Heflin, J. A. Hendler. Dynamic Ontologies on the Web. Procs. of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, p. 443 - 449, 2000.
3. R. Dieng, H.J. Mueller (Eds.). Conflicts in Artificial Intelligence. Springer, 2000.
4. <http://www.w3.org/2001/sw/meetings/tech-200303/social-meaning/>
5. J. Golbeck, B. Parsia, J. Hendler. Trust Networks on the Semantic Web. Proceedings of Cooperative Intelligent Agents, 2003.
6. M. Nickles, G. Weiss. A framework for the social description of resources in open environments. Procs. of the Seventh International Workshop on Cooperative Information Agents (CIA, pp. 206-221). LNCS Volume 2782. Springer, 2003.
7. M. Nickles, Towards a Multiagent System for Competitive Website Ratings. Research Report FKI-243-01, Technical University Munich, 2001.
8. M. Nickles, M. Rovatsos, G. Weiss. Empirical-Rational Semantics of Agent Communication. Procs. of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04), New York City, 2004.
9. M. Nickles, T. Froehner. Social Reification for the Semantic Web. Research Report FKI-24x-04, Technical University Munich, 2004. To appear.
10. A. Farquhar, A. Dappert, R. Fikes, W. Pratt, Integrating Information Sources using Context Logic. Procs. of the AAAI Spring Symposium on Information Gathering from Distributed Heterogeneous Environments, 1995.

# Adding Spatial Semantics to Image Annotations\*

Laura Hollink<sup>1</sup>, Giang Nguyen<sup>2</sup>, Guus Schreiber<sup>1</sup>, Jan Wielemaker<sup>2</sup>, Bob Wielinga<sup>2</sup>, and Marcel Worring<sup>2</sup>

<sup>1</sup> Free University Amsterdam, Department of Computer Science  
{hollink, schreiber}@cs.vu.nl

<sup>2</sup> University of Amsterdam, Informatics Institute  
{giangnp, worring}@science.uva.nl, {jan, wielinga}@swi.psy.uva.nl

**Abstract.** In this paper we discuss a the support of users in adding spatial information semi-automatically to annotations of images. Descriptions of objects depicted in an image are extended with information about the position of those objects. We distinguish two types of spatial concepts: absolute positions of objects (e.g., east, west) and relative spatial relations between objects (e.g., left, above).

We show the use of a tool for a collection of art paintings with pre-existing RDF annotations, including a list of image objects. First, the tool segments a painting into regions. The user selects regions, and labels these with objects from the existing annotation. Then, the tool computes absolute positions and relative spatial relations of the selected regions, and adds these to the annotation. A small evaluation study is reported in which annotations generated by the tool are compared to manual annotations by ten volunteers.

## 1 Introduction

In this paper we discuss semi-automatic annotation of images with spatial information. In a previous study [6] it was shown that people who describe images often use spatial descriptions like "On the left side" or "Below object x". Spatial information is important for describing the composition of an image, and for the identification of specific objects.

Making a complete and elaborate annotation of the content of an image is a time consuming process. Therefore, the human annotator should be supported in this task as much as possible. In spite of improvements in the field, automatic annotation of images is not feasible at the moment. This is due to the fact that what is depicted in an image is highly subjective. Spatial information, however, is mainly objective. This makes it a good starting point for semi-automatic annotation. This work can be seen as an exploration into bridging the "semantic

---

\* An early version of this paper has been accepted for presentation at the 'Workshop on the Application of Language and Semantic Technologies to support Knowledge Management Processes' at EKAW 2004.

gap” [10], which refers to the cognitive distance between the analysis results delivered by state-of-the-art image-analysis tools and the concepts humans look for in images. In this work we use images from a collection of art paintings that we have used in an earlier study about semantic annotation [4]. The system we propose takes an annotated image as input. It segments the image into regions and allows the user to label the regions with concepts from the annotation. The system computes the position of the concepts and the spatial relations between them, and adds the spatial information to the annotation. A small evaluation is done in which annotations generated by our system are compared to manual annotations by humans.

It should be noted that this is an exploratory study to investigate the potential of content-based techniques for (spatial) image annotation at a conceptual level. As will be seen, we have deliberately “cut some corners” with the intention to show whether the idea could work in principle.

In the next section we discuss the representation of spatial information. In Sect. 3 we give a description of our system. Section 4 contains the results of a small evaluation study. The final section contains a general discussion.

## 2 Representing Spatial Relations

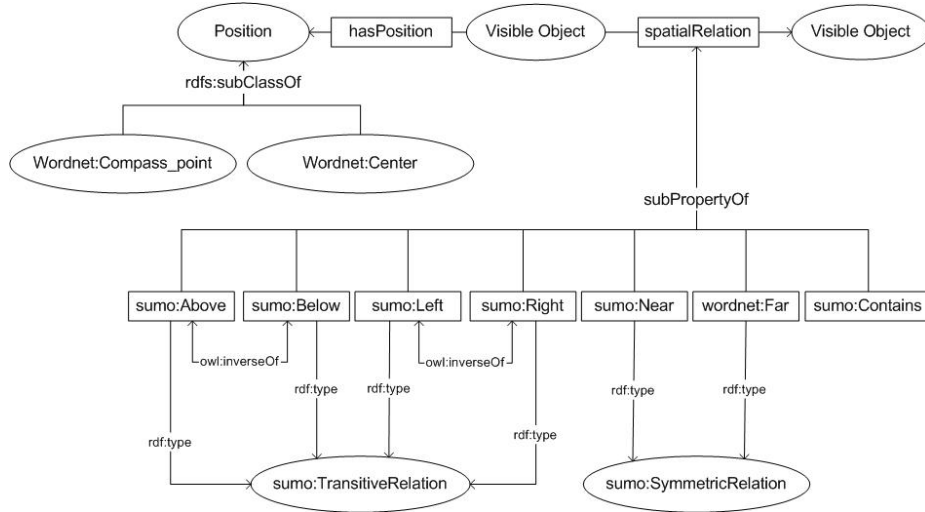
Talmy [12] describes spatial relations in the context of human perception. He conveys that the spatial disposition of an object in a scene is always characterized in terms of another object. The first object, which is called the ‘figure’, is the subject in the expression. The second object, or the ‘ground’, is used as a fixed reference to which the position of the figure is described. Grounds are for example the earth or the body of the speaker. More than one ground object is possible (e.g. “the bike is on the other side of the church”: the bike is the figure, the church is the ground object, the body of the speaker is the second ground object). Another important point is that in human language a finite number of words is used to represent an infinite number of spatial configurations. This means that choices have to be made about which spatial concepts are used in a vocabulary.

Cohn [2] points out that when making a representation of space, questions have to be addressed regarding the kind of spatial entity being used (e.g. regions, points), and the way of describing relationships between these entities (e.g. their topology, size, distance, orientation or shape). For our practical purposes of annotating objects in images, we restricted ourselves to two-dimensional, binary relations between regions. The spatial relations that are included in our vocabulary must be (1) relevant for image annotations, and (2) suitable for automatic detection. This last requirement disqualifies concepts like ‘behind’ and ‘in front of’ since they are very hard to detect.

We distinguish two types of spatial concepts: absolute positions and relative spatial relations. The first are used to describe the position of objects within an image. The image functions here as the ‘ground’ of the expression. A common representation of absolute positions are the compass points North, South, East, West, Northeast, Southeast, Northwest and Southwest. We divided an image

into nine squares where each of the outer squares represents one of the compass points and the middle square represents the center. Relative spatial relations are used to describe positions of objects relative to each other; one object is the ‘figure’, the other is the ‘ground’. The set of relations that we used in this study includes: Right, Left; Above, Below; Near, Far; and Contains. One additional spatial relation can be derived, namely **Next** is either **Left** or **Right**.

In order to add the spatial information to semantic annotations of images, we used concepts from existing ontologies to specify the positions and spatial relations. Spatial relations were taken from SUMO [8]. This is a large, well structured ontology that takes into account Cohn’s ideas about spatial relations.<sup>1</sup> Absolute positions were taken from the general lexical database WordNet [3]. One exception was the spatial relation **Far** that was taken from WordNet since it was not a concept in SUMO (version 1.15).



**Fig. 1.** Spatial concepts (ellipses) and their properties (rectangles) as they are used in our annotation schema.

For each spatial relation that we use we specify whether or not it is a **Symmetric Relation**, or a **Transitive Relation**, and what the **inverse Of** the relation is. RDF Schema is used for the representation of the spatial concepts<sup>2</sup>. Figure 1 depicts an RDF graph of the spatial annotation schema that we use. It shows a **Visible Object** that has a **Position**. The **Position** class has

<sup>1</sup> CVS log for SUO/Merge.txt, <http://ontology.teknowledge.com/cgi-bin/cvsweb.cgi/SUO/Merge.txt>, revision 1.24

<sup>2</sup> One term from OWL was used, `owl:inverseOf`, for there is no notion of opposite properties in RDF.

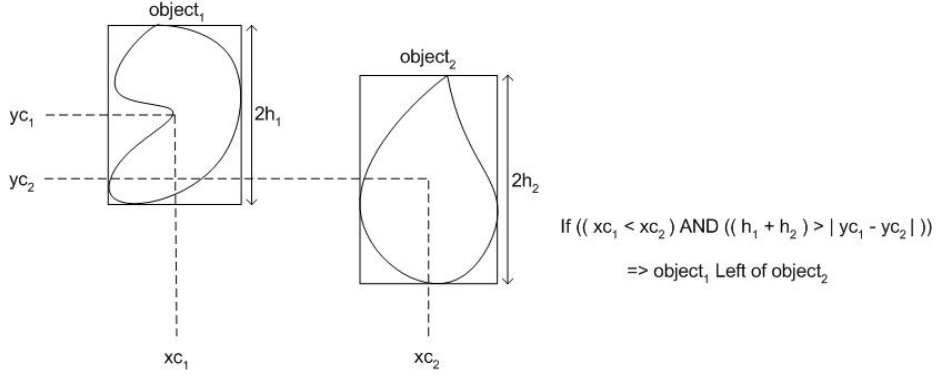
two subclasses, namely the WordNet classes `Compass Point` and `Center`. The `Visible Object` has a spatial relation with another `Visible Object`. We defined the spatial concepts from SUMO as subproperties of the property `spatial Relation`. `Left` and `Right` are each others inverse, just as `Above` and `Below`. All four are `Transitive Relations`. `Far` and `Near` are defined as being `Symmetric Relations`. We disregard Talmy here [12], who points out that near and far are in human language not used as symmetric relations: a bike can be near a house, but nobody will say that the house is near the bike. This has to do with the size and mobility of the objects, which are properties that we do not take into account at this time.

### 3 Spatial Annotation Tool

The system we propose helps the user to add spatial information to image annotations. For this purpose, we use a collection of art paintings that are annotated with the objects that are visible in them. The collection of images is first segmented off-line. For each painting color and texture features are extracted using Gabor filters. Pixels with similarity values above a given threshold are merged into a region. Several segmentations are computed for one painting, using different scales and thresholds.

The interactive annotation process consists of five steps: input, interactive segmentation, annotation, computation of spatial relations, and output. In the *input* step, the user selects a painting from the collection. In the *interactive segmentation* step the relevant objects in the image are identified. In this step we employ the framework described in Nguyen & Worring [7]. The system first offers the user a segmentation of the image using the default set of parameters. The user can now ask for a larger or smaller number of regions, after which the system updates the parameters. This process goes on until the user is satisfied with the segmentation. By allowing the user to give feedback, the resulting segmented image will closely match the user’s expectations. Different purposes require segmentations at different levels.

In the *annotation* step, meaning is added to the relevant objects. The user labels regions in the segmented image with concepts from the annotation. The labelling is done by clicking on a region and clicking on a concept from the annotation. Fig. 3 shows the interface of the system, at the moment that a user is labelling the regions. When the user decides that all relevant regions are labelled, the system continues to the *computation of spatial information* step. In this step, absolute positions and relative spatial relations of the selected regions are computed. Each selected region is represented by a bounding box and the center of the bounding box. Absolute positions are computed by determining in which of nine squares the center is. For the computation of the relative spatial relations we employ the method of Abella & Kender [1]. All relations are computed by comparing the centers and borders of bounding boxes of two objects. In Fig. 2 the definition of `Left` is shown as an example. For details of the other relations we refer to the reference.



**Fig. 2.** Definition of the spatial concept **Left**.

Finally, in the *output* step, the spatial information is written as RDF statements to the original annotation, from where it can be queried by other tools. Fig. 5 depicts a screenshot of the Triple20 toolkit<sup>3</sup>, that can be used to display and query the annotations. The figure shows the graphical output of Triple20 that displays the spatial annotation of the Matisse painting “Conversation” (Fig. 4) as an RDF graph. The annotation includes two objects linked by the SUMO concept **Left**. The position of one of the objects is specified by a WordNet concept with the meaning **East**.

## 4 Preliminary Evaluation

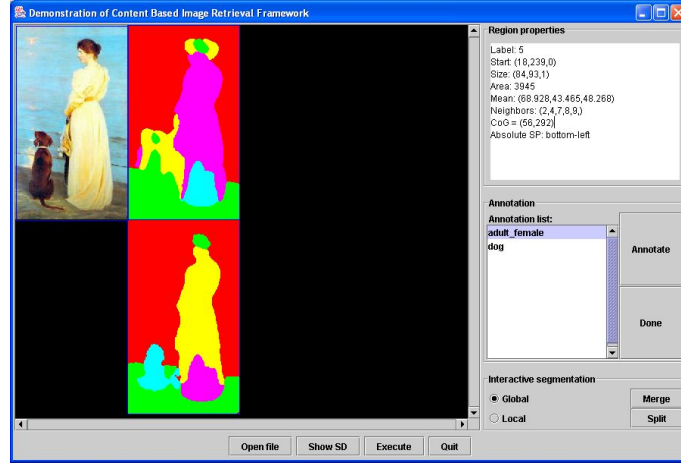
### 4.1 Methods

While designing the tool we have made decisions regarding the choice of concepts that are incorporated, and the definitions of these concepts. In this user study we evaluate these decisions. We asked two questions:

1. Are the spatial concepts that the tool uses the same as the concepts that users would use?
2. Are the definitions of the spatial concepts in accordance with the intuition of users?

Shariff & Egenhofer [9] asked similar questions for relations between lines and regions. They asked human subjects to draw sketches of English-language spatial terms. The sketches were used to map spatial terms onto geometric parameters and their values. One of their results was that topology was more important than metric properties in the selection of spatial terms. We took another approach:

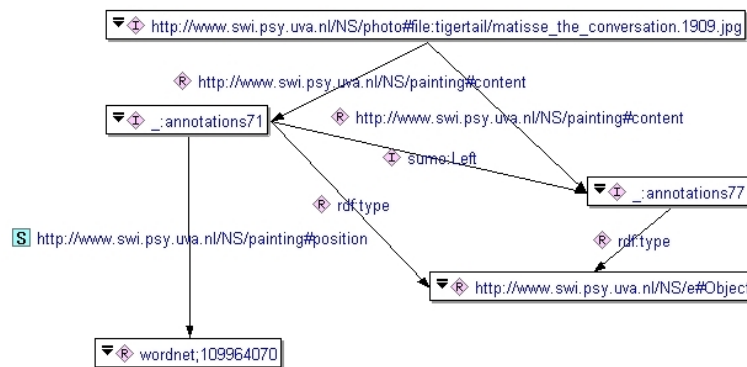
<sup>3</sup> Triple20 is an open-source Prolog-based semantic-web package, see <http://www.swi-prolog.org/packages/Triple20/>.



**Fig. 3.** Screenshot of the spatial annotation tool, showing a painting segmented at two levels. *Region properties* of the selected region are shown in the top right corner. Concepts from the annotation are listed in the *Annotation list*.



**Fig. 4.** “Conversation” by Henri Matisse, 1909



**Fig. 5.** Screenshot of Triple20’s graphical output of a spatial annotation

subjects were asked to select spatial terms when provided with a configuration of objects in an image.

For the study we selected eight paintings that were well segmented by the tool (this seems a legitimate criterium since we are not evaluating the segmentation algorithms). Another criterium was that the paintings had to contain at least two objects. We asked ten PhD students who were familiar with annotation but not in particular with spatial concepts to participate in the study. They were split into two groups of five in order to answer the two evaluation questions.

Group 1 were provided with the eight paintings associated with a list of the objects that were visible on each painting. They were asked to provide statements about the absolute positions and relative spatial relations of these objects. Any number of statements was allowed. Comparing the spatial concepts that were used by Group 1 to the concepts included in the tool, will give an answer to Question 1.

Group 2 was also provided with the eight paintings and a list of objects. They were asked to describe positions and spatial relations using a limited list of spatial concepts. The list contained only the terms that are included in the tool. Again, any number of statements was allowed. Comparison of the statements of Group 2 to the statements of the tool will answer question two. We make the assumption that the spatial concepts that humans select are the correct ones.

## 4.2 Results

**Group 1** In total, 257 statements were written down by Group 1: 129 absolute positions and 128 relative spatial relations (Table 1). 81 Percent of the absolute positions of Group 1 were concepts that were included in the tool. 8 Percent consisted of concepts that were not included in the tool. This were mainly three-dimensional positions such as “background” and “in front”. The remaining 11 percent of the statements of Group 1 were more precise versions of the concepts in the tool. Examples are “almost in the center”, “far right”, “between left and center”.

Of the relative spatial relations only 57 percent of the statements by Group 1 were concepts that were included in the tool. 29 Percent of the descriptions were concepts that were not in the tool; these were mainly three dimensional relations (“behind”, “in front of”), statements about the connectedness of two objects (“connected”, “freestanding”) and “between”. 14 Percent were more precise or less precise versions of concepts in the tool. “Object1 is northwest of Object2” is more precise than the concepts “above” and “left” in the tool, while “Object1 is higher than Object2” is more general than the concept “above” in the tool.

**Group 2** The five subjects of Group 2 produced a total of 234 statements. Together they selected 127 absolute positions of 27 objects (Table 2). Of the 127 positions, 88 (69 %) matched the absolute positions that the tool computed. 39 Positions did not correspond to the computed positions, which seems a high number of mistakes. However, note that the tool cannot match all statements



**Table 1.** Summary of the results for Group 1, divided over absolute positions (AP) and relative spatial relations (SR)

Group 1	AP	SR	Total
Included in the tool	107 (81 %)	70 (57 %)	177 (69 %)
Not included in the tool	11 (8 %)	36 (29 %)	47 (18 %)
Not precise enough in the tool	14 (11 %)	18 (14 %)	32 (13 %)
Total	132 (100 %)	124 (100 %)	256 (100 %)

when the participants disagree about the position of an object. We found that for only seven of the 27 objects a majority of the participants (at least 3) agreed on a position different from the tool’s position. An example of such a mistake by the tool is the window in the Matisse painting *Conversation*. The tool assigned the window the position *North*, while all subjects agreed that it was in the center.

Group 2 produced 107 statements about relative spatial relations. Not all possible relations between two objects were described by the subjects. It appeared that they used the **inverse Of** and **symmetric Relation** properties for the selection of relevant object pairs: when a subject had stated “woman left of man”, he or she would not also state “man right of woman”. To make the statements comparable to the statements of the tool, that did compute relations between each object pair, we added symmetric and inverse relations where necessary. This brought the total number of relative statements of Group 2 to 210 (and the total number of statements of Group 2 to 337). 154 Of these (73 %) were also found by the tool, 56 (27 %) were not.

**Table 2.** Summary of the results for Group 2, divided over absolute positions (AP) and relative spatial relations (SR)

Group 2	AP.	SR.	Total
Found by the tool	88 (69 %)	154 (73 %)	242 (72 %)
Not found by the tool	39 (31 %)	56 (27 %)	95 (28%)
Total	127 (100 %)	210 (100 %)	337 (100 %)

Another evaluation measure is the proportion of statements of the tool that corresponds to statements of the subjects. The tool computed 106 statements. 24 Of these were about an object pair that was not described by any of the participants, which means they cannot be validated. Of the remaining 82 statements, 56 ( 68 %) corresponded to at least one participant. Of the 26 ‘incorrect’ statements of the tool, 18 concerned **far** and **near**. Participants hardly used these concepts.

## 5 Discussion

In this paper we explored the possibility to use a content-based image analysis technique to aid the process of spatial image annotation. The study shows there are indeed some points where the “semantic gap” can be bridged. A number of spatial concepts specified by human annotators were compatible with annotations produced by the tool. The results of the study seem to indicate that the absolute positions in the tool are roughly the same as the concepts that human annotators use. However, a number of relative spatial relations that people tend to use are missing from the tool. The choice of the set of spatial concepts was based on pragmatics, namely those for which automatic detection methods were available. The evaluation showed that this is a severe limitation since people often use three dimensional concepts, which are very hard to detect. Other frequently used concepts that the tool could not handle were **connected** and **between**. We are planning to include those in the next version of the spatial annotation tool. Two concepts included in the tool were hardly used by human annotators: **Far** and **Near**. It would be interesting to see whether this is also the case in other domains than art paintings.

The tool detected almost three quarters of the spatial concepts selected by humans. The results for relative spatial relations were slightly better than for absolute positions. This could be due to the fact that the tool assigns one position to each object, while any number of spatial relations can be detected for one pair of objects. This makes it possible to match all statements, even if subjects disagree with each other.

This was just an exploratory study with the aim to see whether this approach could work in principle. We can see the following lines of research as interesting follow-up options. Firstly, one could think of extending the functionality of the image-analysis tool to include a larger set of spatial relations. In the short term, this is likely to be limited to two-dimensional relations. Secondly, we should include ontological reasoning to derive spatial relations from the existing annotations. Such functionality is currently not included. Thirdly, one could consider including facilities for manual segmentation. This could improve the quality for images that are segmented badly by automatic techniques. Ley [5], for example, uses SVG to manually define regions and then annotates each region. Finally, it would be worthwhile to consider whether the content-based segmentation can also be used for other annotation purposes. One can think of other non-spatial properties of which the value can be derived with the help of segmentation. One example would be the color of a particular object. In the VisualSEEk system [11], for example, query by sketch is done based on colors and (relative) spatial locations of regions in an image.

## Acknowledgements

This work is supported by the project “Interactive Disclosure of Multimedia Information and Knowledge” funded by the IOP Programme of the Dutch Ministry

of Economic Affairs. We thank the subjects for investing their precious time in our evaluation study.

## References

1. A.Abella and J.R.Kender. From images to sentences via spatial relations. In *Proc. of the ICCV'99 Workshop on Integration of Image and Speech Understanding*.
2. A.G. Cohn and S.M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamentae Informaticae*, (46):2–32, 2001.
3. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.
4. L. Hollink, A.Th. Schreiber, J. Wielemaker, and B. Wielinga. Semantic annotation of image collections. In *Proc. of the K-CAP 2003 Semannot Workshop*, Florida, USA, October 2003.
5. J. Ley. Raster image description and search in svg. Presented at the third annual conference on Scalable Vector Graphics (SVG Open): <http://www.jibbering.com/svg/talk2004/title.html>, Tokyo, Japan, September 2004.
6. L.Hollink, A.Th.Schreiber, B.Wielinga, and M.Worring. Classification of user image descriptions. *Int. Journal of Human Computer Studies*, November 2004.
7. G.P. Nguyen and M. Worring. Query definition using interactive saliency. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, Berkeley, CA, USA, 2003.
8. I. Niles and A. Pease. Towards a standard upper ontology. In Chris Welty and Barry Smith, editors, *Proc. of FOIS-2001*, Ogunquit, Maine, October 17-19.
9. A. Rashid, B.M. Shariff, and M.J. Egenhofer. Natural-language spatial relations between linear and areal objects: The topology and metric of english-language terms. *Int. Journal of Geographic Information Science*, 12(3):215–246, 1998.
10. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), December 2000.
11. J.R. Smith and S-F. Chang. Visualeek: a fully automated content-based image query system. In *Proceedings of ACM Multimedia*, pages 87–98, Boston, MA, November 1996. ACM Press.
12. L. Talmy. How language structures space. In H. Pick and L. Acredols, editors, *Spatial Orientation: Theory, Research and Application*, New York, 1983. Plenum Press.

# Ontology-enablement of a system for semantic annotation of digital documents

William J BLACK,<sup>1</sup> Simon JOWETT,<sup>1</sup> Thomas MAVROUDAKIS,<sup>2</sup> John McNAUGHT,<sup>1</sup>  
Babis THEODOULIDIS,<sup>1</sup> Argyrios VASILAKOPOULOS,<sup>1</sup> Gian-Piero ZARRI,<sup>3</sup>  
and Kalliopi ZERVANOU,<sup>1</sup>

<sup>1</sup>*Department of Computation, UMIST, PO Box 88, Manchester, UK*

<sup>2</sup>*Hellenic Ministry of National Defence, 151 Messogion Av., 15500 Athens, Greece*

<sup>3</sup>*LaLICC, University of Paris IV/Sorbonne, 96 boulevard Raspail – 75006 Paris, France*

**Abstract.** We describe the recent enhancement of the CAFETIERE formalism (Conceptual Annotation of Facts, Events, Terms, Individual Entities and RELations) with the ability to link natural language words and phrases in textual documents with instances and classes from a language-enabled ontology. The language-enabled ontology is one with an index from one or more natural language expressions to each concept (as in WordNet). In an information extraction application, the index, ontology and instance repository are consulted in place of the usual gazetteer prior to the application of the context-sensitive phrase structure rules of the CAFETIERE formalism. Information from the ontology and its instances is cached so that rules can be constrained by properties of objects and can in turn build representations using those properties. We describe the notational extensions to CAFETIERE and give examples of the extraction of event instances in the analysis of texts relative to a specific application ontology. Relevant background is given on the architecture and common annotation scheme of the Parmenides system (FP5 project), in the context of which this work has been done.

## Introduction

The vision of the Semantic Web implies that digital documents are enhanced with conceptual metadata that can support indexing and inference about the contents of the documents, as argued in [1, 2]. In the Parmenides project (IST project IST-2001-39023), we are also concerned with mining pre-analyzed texts to discover patterns of temporal relations between events[3]. Fully-automatic IR-based approaches to document indexing and search appeal because the alternative is to run up against the knowledge acquisition bottleneck, with its attendant need for expensive intellectual effort.

In Parmenides, we adopt the middle way of using automated analysis at a higher level than pure IR indexing, drawn from the body of Information Extraction techniques[4, 5]. These mechanisms, defined and refined in the MUC conferences,<sup>1</sup> involve intermediate-level natural language analysis techniques to identify the extent and referent class of proper names and other expressions in text, and building on that, extract relational and factoid information, filling slots in templates or predicate-argument structures.

Because of the inherent limits in the accuracy of information extraction, the Parmenides architecture prominently features an annotation editing tool which allows missing and spurious analyses to be corrected, while still benefitting from time savings compared with fully human-edited annotation.

---

<sup>1</sup>See [http://www.itl.nist.gov/iaui/894.02/related\\_projects/muc/proceedings/ie-task.html](http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ie-task.html).

The traditional IE system produces textual output, whereas the *Parmenides* requirement is to have not merely classified text spans, but rather to identify the knowledge-base instances denoted by each extracted phrase, and have the predicates and arguments in template representations identified with ontology classes and instances. In this paper, we show how this can be done both during manual/intellectual annotation and automatic information extraction.

Section 1 introduces the essence of the common annotation scheme which is a DTD defining the format that is used for inter-module communication in all phases.

Section 2 outlines the basic analysis pipeline of the *Parmenides* system and clarifies the role of the basic components, concentrating on the module responsible for looking up items in the knowledge base, and the module responsible for identifying phrases and structures based on a combination of syntactic analysis and the integration of information from different levels of analysis and sources of background knowledge. This discussion includes the role of the “common annotation scheme” as a lingua franca for structural and conceptual annotations. Section 2.2 explains essentials of the Cafetiere formalism which conducts a rule-based analysis to build annotations of spans and to fill templates. Section 3.3 shows how Cafetiere has been extended for ontology linkage to achieve this goal.

## 1 The Common Annotation Scheme

The *Parmenides* Common Annotation Scheme (CAS) is an XML representation which consists of three types of annotations as described in [6]

**Structural Annotations:** These define the structure of the document (head, body and further sections, paragraphs, sentences and tokens). These annotations are in-line annotations i.e. they contain the text spans they label.

**Lexical Annotations:** These identify lexical units of interest (entity instances), such as person’s names, organizations, drug names, time expressions, etc. and are token-reference annotations, i.e. they do not contain textual spans but refer to unique token IDs instead.

**Semantic/Conceptual Annotations:** These are also token-reference annotations referring to specific (already marked up as lexical annotations) entities via co-referential IDs. They mark entities, relationships and events.

## 2 A sketch of the *Parmenides* analysis pipeline

The analysis conducted in *Parmenides* is a pipeline in which each stage of analysis adds to the annotations of its predecessors. This is depicted in Figure 1 where the steps are numbered for convenience. Step 1 involves conversion from external formats to an XML document conformant with the Common Annotation Scheme DTD. Step 2 breaks the text into single word (and equivalent) tokens, and step 3 applies a part of speech tagger [7] to associate the contextually most likely part of speech tag for each token.

Step 4 is a necessary but not sufficient mechanism allowing phrases identified and classified in subsequent stages to be mapped to known classes or instances, i.e. to ground the textual annotations in the ontology. More information on this mechanism follows in Section 2.1.

Step 5 exploits any or all of the prior stages of analysis, together with syntactic rules, to build conceptual annotations representing entities, events and relations. This is discussed further in Section 2.2.

Step 6 allows the user to validate and correct or augment the analyses produced by the automated steps of the system pipeline. This is done using a custom-built annotation editor [8], since such a user may modify annotations but has no right to edit the underlying content.

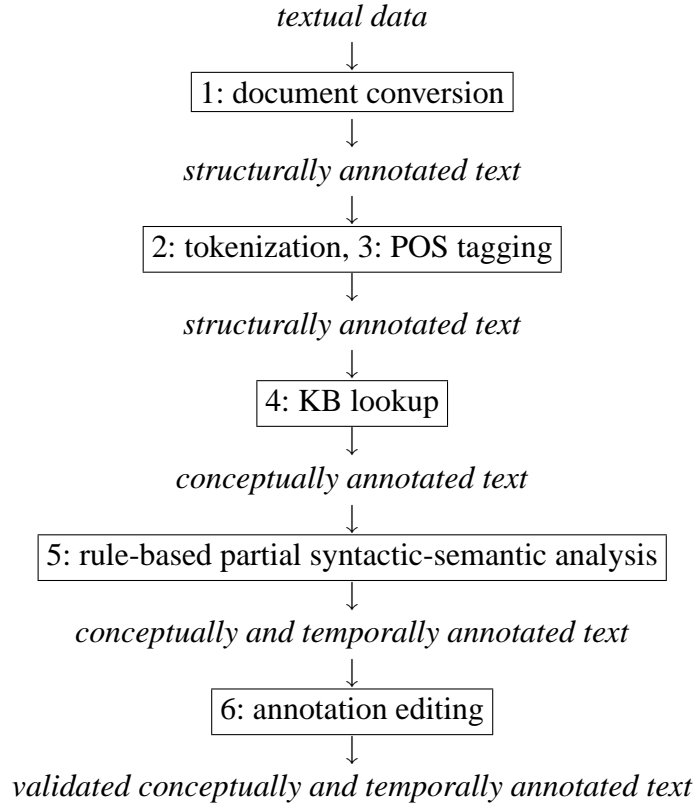


Figure 1: Essential steps in the Parmenides analysis pipeline

### 2.1 Lookup

The lookup module consults an index that maps a word or phrase to a class label or instance identifier. Since the same string (e.g. “*Washington*”) can denote entities of different classes, the lookup annotation is a disjunction of possible *phrase*  $\rightarrow$  *identifier* mappings. Even when singly-valued, the gazetteer entries are not relied on to annotate text spans, but provide additional evidence for the rule-based analysis phase about the concepts represented by text spans.

### 2.2 Rule-based partial syntactic-semantic analysis

The Parmenides temporal text mining architecture uses the CAFETIERE [9] formalism to identify “basic semantic elements” from texts. CAFETIERE stands for “Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations.”

The product of the analysis is a set of conceptual annotations as described in section 1. Unlike a ‘classical’ information extraction (IE) application, the annotations are linked to the classes and instances of an application-oriented ontology.

Since ultimately, the goal is the discovery of trends in the coincidence of event types, the units to be extracted are ultimately occurrences (or *facts*). These occurrences are classified relative to a hierarchy of event classes (the NKRL [10] H-TEMP), which is described further in 3.

In addition to classification, the temporal grounding of the *event* as indicated by verb tense and aspect, and by temporal adverbials are extracted as features of a lexical annotation. The representation of the occurrence needs the arguments (subject, object, etc.) of the verb (or event-denoting noun) to be identified, to complete a template instance, one of the classes of

conceptual annotation supported by the system. The arguments themselves are either named *individual entities* or objects denoted by *terms* in the domain under analysis (identified via the ontology of entities – the H-CLASS in NKRL).

Some basic semantic elements identified in rule-based analysis are instances of concepts already in the domain ontology, although others are discovered during analysis.

All events whose instances are to be annotated must be in the ontology, but for other elements, the class can be determined heuristically from contextual clues.

Not all proper names need to be known to the system prior to analysis, because following the state of the MUC art, it is possible to classify names accurately from their textual occurrence and context. Similarly, not all unnamed entities need to be known beforehand. Common noun phrases can be analysed syntactically, or alternatively, annotations can be confined to those for which statistical evidence suggests domain termhood.

Rule-based analysis is used in creating all lexical annotations above the token level, and all conceptual annotations. Items found in the ontology lookup phase must be confirmed by rules, which may specify contextual constraints that will disambiguate when the same string can name or describe different objects.

The rule-based analysis formalism is essentially similar to that reported in [11], but enhanced to give various extensions to its expressive power, and now based on a compiled FST implementation.

Rules have the form  $A \Rightarrow B \setminus C / D$ ;  $A$  describes the text span if the rule succeeds, and  $C$  represents a sequence of one or more constituent phrases. The rule, being *context-sensitive*, requires elements  $B$  and  $D$  to be found to the left and right of  $C$  in order to label the constituents  $C$  as the phrase  $A$ .

Phrases and their constituents are described by a set of attribute-value pairs enclosed in square brackets; both negation and disjunction of values are supported; attributes range over orthographic, morpho-syntactic and semantic/conceptual properties; attributes are used as in HPSG-like linguistic formalisms both to constrain and to construct representations by means of feature unification (through Prolog-like named variables); there is a mechanism to identify longer-distance relationships such as anaphoric co-reference. Examples of rules are (1) and (2).

- (1) `[syn=NP, sem=ORG, sector=EDU, loc=_LOC] =>`  
`\ [token="University"],`  
`[token="of"],`  
`[sem=LOC, token=_LOC] / ;`
- (2) `[syn=NNP, sem=PERSON] =>`  
`[sem=title]{1,2}`  
`\ [orth=capitalized],`  
`[orth=upperinitial]?,`  
`[orth=capitalized] / ;`

The annotation being constructed is described on the first line of rule (1), by the features `syn`, `sem`, `sector` and `loc`. The first three of these features are ascribed in the rule, but the feature `loc` takes its value from the variable `_LOC`, which *shares* with the other instance which is the value of the `token` feature of the last word in the phrase. (Variables are recognizable to the system by having an initial underscore.) The symbols `\` and `/` mark the boundary between the phrase's constituents and its left and right contexts respectively. In (1) there are no contextual constraints, but in (2) the capitalized words with optional middle initial have to be preceded by a title for the phrase to be considered the name of a person.

### 2.3 Semi-automatic metadata annotation

Annotation is semi-automatic, which means that various levels of NLP processing are applied to the text, but because of the inherent limits to accuracy in such analyses, an editor is able to verify or correct the analysis in an annotation editing tool. As stated earlier, all annotations conform to a common annotation scheme defined in XML.

The annotation editing tool [8] is custom-built for the annotation scheme. We do not use a standard XML editor because the user does not change the underlying text, only the annotations on it.

The three levels of annotation fall in a strict order of precedence: Structural annotations must be present before lexical annotations are added, and the latter must be present before corresponding conceptual annotations may be added.

The user can edit any document that has been through at least the first phase of analysis (Stage 1 in Figure 1).

## 3 Ontology exploitation

In the project, four different applications are being developed, each supported by its own ontology developed by domain experts. Such an ontology needs an explicit mapping of words and phrases to concepts in order to be linked to information extraction rules. If all classes specify a multi-valued string property *synonym*, then it is straightforward to expect the ontology editors to add the synonymous natural language strings for an instance, e.g. “New York”, “NY”, “The Big Apple”. We are, however, interested in matching not just the proper names of known individual objects, but also domain terms, such as “phase III clinical trial” which are represented in natural language by indefinite and definite descriptions and not by names. Similarly, with our focus on events, we want to match natural language verbs and nouns with event or occurrence-denoting concepts. At the least, we need an ontology framework that allows natural language synonyms to be defined for concepts as well as instances, and an index to facilitate lookup via the synonym property.

Rule 3 shows a verb group or event-denoting noun being labelled semantically with the instantiation to the variable `_lp` that has been made by the lookup module (expressed by the condition `lookup=_lp`). The rule also passes on instantiations of the variables `_TNS`, `_POL` and `_ASP`, unpacked by a previous rule from the part of speech tag.

```
(3)  # A generic event
      [sem=_lp, oid=_lp, id=_id, type=PEVENT,class=OCCURRENCE, tense=_TNS,
      polarity=_POL, aspect=_ASP, rulid=event_gen1] =>
      \
      [syn=event_noun|event_phrase, lookup=_lp, lookup!=NIL, lookup<=event,
      tense=_TNS, polarity=_POL, aspect=_ASP, id=_id]
      /
      ;
```

When Rule 3 is applied, all occurrences mentioned in phrases syntactically analysed as `event_noun` or `event_phrase`, and which have synonyms defined in the ontology, will be visible in the annotation editor. The most important features illustrated by this rule are the three conditions `lookup=_lp`, `lookup!=NIL`, `lookup<=event`. The first of these has the effect of instantiating the variable `_lp` if the second condition is satisfied, that is, if there is a non-null result for lookup. The expression `lookup<=event` specifies that the lookup property of the phrase has to be the class `event`, or any of its subclasses or subclass instances. This simple extension of the rule language to exploit inheritance replaces many individual rules in the pre-ontology version.



### 3.1 Templates

An occurrence is not simply a text span in the same way that a name can be. The goal of information extraction is to find from a text the slot fillers for a template representation of occurrences of interest. An ontology that represents prototypical events in the same way can assist this process. Given an interest in management change events, a domain expert has defined an appointment as an occurrence with typed slots for the employer, employee and position, in addition to the time of occurrence.

When used manually following rule-based analysis, the Annotation Editor presents a slot representation of the occurrence to the user for completion, retrieving the names and filler types of each slot from the knowledge base. Candidate fillers for each slot, as found either by rules identifying names and other basic expressions, or by previous editing, are presented in drop-down lists. This ensures the integrity of all annotations, with respect to the ontology.

#### 3.1.1 Rule-based slot filling

So far, ontology linkage has not provided the means to fill slots automatically. Modifying rule (3) with the condition `lookup<=person-company-event` in place of `lookup<=event`, and specifying further constituents to be found in its right context, as in (4), allows the employee and role slots to be filled from the objects of the verb phrase or prepositional phrases modifying the event noun.

```
(4)  # Appointment event with person then role as objects
      [syn=VP, sem=_lp, oid=_lp, id=_id, employee=_eeid,
       c_position=_posid, type=PEVENT, class=OCCURRENCE, tense=_TNS,
       polarity=_POL, aspect=_ASP, rulid=event_Appl]    =>
      \
      [sem=event_noun|event_phrase, lookup=_lp, lookup!=NIL,
       lookup<=company-person-event, tense=_TNS, polarity=_POL,
       aspect=_ASP, id=_id]
      /
      [token="of"]?,
      [sem=person, id=_eeid],
      [token="of"|"as"|"to"]?,
      [sem=position, id=_posid]
      ;
```

Similar rules for other event types will find slot fillers automatically, reducing but not eliminating the amount of annotation to be done by hand. However, the constraints on the slot fillers as recorded in the ontology's event templates have to be reproduced when writing each such rule.

The approach is suitable when only a small number of event types are of interest to the application. For application to the broader domain of the Semantic Web, the slot type constraints need to be expressed in the ontology, and not re-expressed in pattern-matching rules.

#### 3.1.2 NKRL: an event-template oriented knowledge representation framework

From the linguistic information extraction point of view, allowing the user complete freedom to name slots is not ideal, so we have considered a more disciplined approach to knowledge base construction and occurrence annotation, that of NKRL (Narrative Knowledge Representation Language) [10]. The most important innovation of NKRL with respect to similar knowledge representation tools (KRL, Conceptual Graphs, etc) consists in the addition of an ontology of events (i.e. a catalogue of standard, formalised representation of characteristic situations and events) to the usual ontology of concepts. Thus, the NKRL tool relies on

two ontologies, a hierarchy of concepts (H\_CLASS) and a hierarchy of events (templates, H\_TEMP).

H\_TEMP templates are NKRL predicative structures representing general classes of events they are the models of the predicative occurrences. The predicative occurrences are the NKRL representation of specific events: they instantiate the templates by replacing the variables with specific concepts from the hierarchy H\_CLASS. In this way, occurrences describe the semantic contents of documents.

A template has a name, a parent template, a natural language description, a predicate, a set of roles (mandatory, forbidden or optional), a set of mandatory modulators, and a set of forbidden modulators. The predicates are: BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, and RECEIVE. The roles are SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL, TOPIC and CONTEXT. The SUBJ role is mandatory for every template. There are two classes of predicate arguments (role fillers): simple and complex. A simple argument can be a concept from H\_CLASS, or a variable restricted to some values in H\_CLASS. A complex argument is built using an AECS operator (ALTERN, ENUM, COORD or SPECIF) and a list of arguments that, again, can be simple or complex and must comply with the "priority rule": ALTERN (ENUM (COORD (SPECIF))). The roles SUBJ, OBJ, SOURCE, BENF may have a location associated with them. As an example, a template from a Greek MOD case study H\_TEMP is shown below. (5) shows the concept and its hierarchical parent, (6) shows the constraints the event concept has on its arguments, (7) shows a text fragment to which this applies and (8) is a set of related filled templates analysing text fragment (7). In (6), 'symbolic\_label' - an element of the "standard" ontology of concepts of NKRL, H\_CLASS - is there to denote that the ("structured") information to be transmitted is formed by a set of predicative occurrences, associated within a second order structure called a "binding occurrence". In (7), 'symbolic\_label' is then instantiated into 'mod.c3', the symbolic name of a specific binding occurrence stating that the content of the message transmitted by the Philippine Army consists of the two simultaneous - COORD(ination) - events represented by 'mod3.c4' and 'mod3.c5'.

(5) Name: Move:StructuredInformation  
 Parent: Move:TransmitInformation  
 Description: 'Transmit an item of Structured Information'

(6) MOVE SUBJ var1:[(var2)]  
 OBJ var3  
 [SOURCE var4:[(var5)]]  
 [BENF var6:[(var7)]]  
 [MODAL var8]  
 [TOPIC var9]  
 [CONTEXT var10]  
 {[ modulators ], ¬abs}  
 var1 = <human\_being\_or\_social\_body>  
 var3 = <symbolic\_label>  
 var4 = <human\_being\_or\_social\_body>  
 var6 = <human\_being\_or\_social\_body>  
 var8 = <artefact\_>| <information\_support>|<service>| <transmission\_medium>  
 var9 <sortal\_concept>  
 var10 = <situation\_>| <symbolic\_label>  
 var2, var5, var7 = <physical\_location>

(7) ZAMBOANGA CITY: A son of a wealthy Filipino businessman was abducted by

armed members of the most violent Muslim rebel group in the southern Philippines, the military said yesterday. Robustiano Hablo, 30, was on his way home with his father when the Abu Sayyaf rebels blocked their way in a village south of Manila on Saturday.

```
(8) mod3.c2) MOVE      SUBJ      PHILIPPINE_ARMY: (ZAMBOANGA_CITY)
                        OBJ        #mod3.c3
                        date-1:    21/11/1999
                        date-2:

Move:StructuredInformation (4.42)

mod3.c3) (COORD      mod3.c4  mod3.c5)

mod3.c4) PRODUCE SUBJ      (SPECIF GROUP_1 armed_): (VILLAGE_1)
                        OBJ      kidnapping_
                        BENF      ROBUSTINIANO_HABLO
                        date-1:    20/11/1999
                        date-2:

Produce:PerformTask/Activity (6.3)

mod3.c5) MOVE      SUBJ      (COORD1 ROBUSTINIANO_HABLO INDIVIDUAL_20): ( )
                        OBJ      (COORD1 ROBUSTINIANO_HABLO INDIVIDUAL_20):(home_)
                        date-1:    20/11/1999
                        date-2:

Move:PersonDisplacement (4.31)
```

NKRL's limited set of role names in place of predicate-specific roles such as employer, employee and position is a positive benefit, from the point of view of making template filling rules sufficiently generic, but the choice to restrict predicates to a narrow set of primitives is not so compelling.

### 3.2 PS-NKRL

An implementation of a simplified variant of NKRL has been made by Wordmap, who provide commercial taxonomy management systems,<sup>2</sup> for use in Parmenides applications. This variant allows for the import of the two NKRL hierarchies, but does not constrain class definitions to observe the restrictions either on predicate names or role names.

PS-NKRL has three aspects. The first is to define a constrained version of NKRL suitable for the needs of the analysis module. The second is to make the PS-NKRL ontologies available through a suitable navigation API and allow the manipulation of these through the WORDMAP Ontology Manager.

### 3.3 Ontology extensions to the CAFETIÈRE rule formalism

With a knowledge base in place of a gazetteer, the lookup stage of analysis can do more than before: As with the gazetteer, it supplies semantic classes (concepts) corresponding to words and phrases. It returns object identifiers and slot values for known instances, including where aliases and abbreviations name the same object. It retrieves the slots to be filled for anonymous instances of a class, including the types of slots of an event.

The rule formalism is extended with additional operators as follows:

---

<sup>2</sup>See <http://www.wordmap.com>

- The comparison operator `<=` which exploits inheritance at lookup-time, as explained above.
- The dot (`.`) operator between slot names, which allows access to the value of a slot of an object which is the filler of a slot in the current constituent.  
For example `capital.population=_pop` would instantiate `_pop` with the appropriate value, say 10000000, if the current constituent is an instance looked up from the string “United Kingdom”, which has as the value of a slot named `country`, an entity for whom the population slot has the value 10000000.

### 3.3.1 *Obtaining event constraints from the ontology*

In (9), we see a general syntactic rule matching a simple subject-verb-object sequence that builds the semantic representation needed without the template-specific slot names that were used in (4).

```
(9) [syn=_syn, sem=_event, subj=_s, eid=_event, obj=_o] =>
    [syn=np, lookup<=_sc, eid=_s]
    \ [syn=_syn, lookup<=event, lookup=_event, subjectclass=_sc,
      objectclass=_oc] /
    [lookup<=_oc, eid=_o] ;
```

This rule will match an appropriate verbal constituent and fill its slots if the looked-up class has the slots `subjectclass` and `objectclass` and their respective values match the lookup values for the preceding and following constituents.

General syntactic rules like this can recognize and fill the slots for a wide range of event types, provided the slot constraints are expressed in these general terms and not by roles particular to the event type. However, such a policy is not suited to the outlook of application owners, who are not linguistically oriented, and who will be unable to map conceptual slots to abstract syntactic roles unaided.

## 4 Conclusions and Future Work

We have described a technical mechanism by which a rule-based information extraction system can be linked to an ontology and instance repository. This is necessary to produce semantic annotation of digital documents with the aid of natural language processing components. The mechanism is also supported by an ontology-enabled annotation editor. The ontology resource is an implementation of NKRL, embedded in an ontology management tool by Wordmap, although we are able to support other knowledge base formalisms, such as Protégé.

Further work is needed on enabling the needs of natural language ontology lookup to co-exist with that of the application owners to name slots as they see fit, and to attain generality of analysis without the writing of excessive domain-specific rules.

## Acknowledgments

The Parmenides project is co-funded by the European Commission (contract No. IST-2001-39023) and the project partners, and by the Swiss Federal Office for Education and Science (BBW/ OFES). Please see <http://www.crim.co.umist.ac.uk/parmenides> for a detailed description of the project. The Parmenides consortium consists of the following partners (with responsible persons): Biovista (GR) Andreas Persidis; Ministry of Defence (GR) Thomas

Mavrouidakis, Spiros Taraviras; Neurosoft (GR) Giorgos Orphanos; Otto-von-Guericke Universität Magdeburg (D) Myra Spiliopoulou; Coordinator: UMIST (UK) Babis Theodoulidis, William Black; Unilever (NL) Hilbert Bruins Slot, Chris van der Touw; University of Geneva (CH) Margaret King; University of Zurich (CH) Fabio Rinaldi; Wordmap (UK) Will Lowe.

## References

- [1] Giam-Piero Zarri. Semantic Web and Knowledge Representation. In A. Min Tjoa and R.R. Wagner, editor, *Database and Expert Systems: Proceedings of 13th International Conference, DEXA'02*, Los Alamitos, CA, 2002. IEEE Computer Society Press.
- [2] Fabio Rinaldi, Kaarel Kaljurand, James Dowdall, and Michael Hess. Breaking the Deadlock. In *Proceedings of the International Conference on Ontologies, Databases and Applications of SEMantics (ODBASE'03)*, Catania, Sicily, Italy, 2003.
- [3] Myra Spiliopoulou, Fabio Rinaldi, William J. Black, Gian Piero Zarri, Roland M. Mueller, Marko Brunzel, Babis Theodoulidis, Giorgos Orphanos, Michael Hess, James Dowdall, John McNaught, Maghi King, Andreas Persidis, and Luc Bernard. Coupling Information Extraction and Data Mining for Ontology Learning in PARMENIDES. In *Proceedings of RIAO 2004*, Avignon, 2004.
- [4] J. Hobbs. The Generic Information Extraction System. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 87–91, Baltimore, Maryland, 1993. Morgan Kaufmann, San Francisco, California.
- [5] D. Appelt and J. Hobbs and J. Bear and D. Israel and M. Kameyama and A. Kehler and D. Martin and K. Myers and M. Tyson. SRI International FASTUS System: MUC-6 Test Results and Analysis. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 237–248, Columbia, Maryland, 1995. Morgan Kaufmann, San Francisco, California.
- [6] Fabio Rinaldi, James Dowdall, Michael Hess, Jeremy Ellman, Gian Piero Zarri, Andreas Persidis, Luc Bernard, and Haralampos Karanikas. Multilayer Annotations in PARMENIDES. In *K-CAP2003 workshop on Knowledge Markup and Semantic Annotation*, page (to appear), Sanibel, Florida, USA, 2003.
- [7] Argyris Vasilakopoulos. Improved Unknown Word Guessing by Decision Tree Induction for POS Tagging with TBL. In S. Clark and M. Osborne, editors, *6th Annual CLUK Research Colloquium*, Edinburgh, 2003.
- [8] Argyris Vasilakopoulos, Michele Bersani, and William J. Black. A Suite of Tools for Marking Up Textual Data for Temporal Text Mining Scenarios. In *LREC 2004*, Lisbon, 2004. *to appear*.
- [9] William J. Black and John McNaught and Argyris Vasilakopoulos and Kalliopi Zervanou and Babis Theodoulidis and Fabio Rinaldi. CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RELations. Technical Report TR-U4.3.1, Department of Computation, UMIST, Manchester, 2003. <http://www.co.umist.ac.uk/~jwb/parmenides/tr-u4.3.1.pdf>.
- [10] G. P. Zarri. NKRL, a knowledge representation tool for encoding the meaning of complex narrative texts. *Natural Language Engineering*, 2/3(3):231–253, 1997.
- [11] W J Black, L Gilardoni, F Rinaldi, and R Dressel. Integrated text categorisation and information extraction using pattern matching and linguistic processing. In *Proceedings of RIAO97*, pages 321–335, Montreal, 1997.
- [12] J.R. Hobbs, M.E. Stickel, D.E. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993.

# Keyword Extraction from the Web for Personal Metadata Annotation

Junichiro Mori<sup>1,3</sup>, Yutaka Matsuo<sup>2</sup>, Mitsuru Ishizuka<sup>1</sup>, and Boi Faltings<sup>3</sup>

<sup>1</sup> University of Tokyo, Japan

jmor,i,ishizuka@miv.t.u-tokyo.ac.jp

<sup>2</sup> National Institute of Advanced Industrial Science and Technology, Japan

y.matsuo@carc.aist.go.jp

<sup>3</sup> École Polytechnique Fédérale de Lausanne, Switzerland

junichiro.mori, boi.faltings@epfl.ch

**Abstract.** With the currently growing interest in the Semantic Web and Social Networking, personal metadata is coming to play an important role in the Web. This paper proposes a novel keyword extraction method to extract personal metadata from the Web. The proposed method is based on co-occurrence information of words. Our method extracts relevant keywords depending on the context of a person. Our experimental results show that extracted keywords are useful for personal metadata creation. We also discuss the annotation of personal metadata and application to the Semantic Web.

## 1 Introduction

The Semantic Web[2] is a new paradigm which brings “structure” to the meaningful content of the Web. With currently growing interest in the Semantic Web and new standards for metadata description such as the Resource Description Framework (RDF)[13], metadata is gradually gaining popularity in the Web.

Another recent trend in Web development is “Social Networking”[7]. Social Networking sites are community sites through which users can maintain an online network of friends or associates for social or business purposes. Numerous Social Networking sites have been launched recently.

As seen in Social Networking, a user itself is gradually coming to play a central role in the Web contents (e.g. In “Weblog”, variety of contents is created by a user). With these recent Web trends, expressing metadata about people and the relations among them is recently gaining interest. In fact, some vocabularies and frameworks for personal metadata description have been developed [5][9][15][16].

Using these vocabularies, a user is gradually creating his or her personal metadata. However, as a major problem of the Semantic Web is the metadata annotation, personal metadata must also overcome the problem and need methods that facilitate and accelerate metadata annotation [8][10]. Although there are some supporting tools to create personal metadata such as Foaf-a-Matic<sup>4</sup>, this tool facilitates only basic descriptions.

---

<sup>4</sup> <http://www.ldodds.com/foaf/foaf-a-matic.html>

Considering personal metadata, we notice that a lot of information is contained in the Web pages. For example, imagine a researcher: that researcher's information can be in an affiliation page, a conference page, an online paper, or even in a Weblog. In fact, we can expect that these pages contain a lot of personal metadata even including information that we would not expect to find. Therein, questions are:

- What kind of personal metadata are in the Web?
- What kind of Web page contains personal metadata?
- How are extracted metadata applied to semantic annotation?

Considering these points, one of our research goals is to extract personal metadata from the Web and apply them to semantic annotation. As a preliminary report to achieve this goal, we propose a novel keyword extraction method to extract personal information from the Web.

The remainder of this paper is organized as follows: section 2 describes the proposed keyword extraction method using an actual example. In section 3, we show the extracted keywords and analyze them. In section 4, we discuss the annotation of personal metadata. Section 5 contains related works. Finally, we address future works and conclude this paper in section 7.

## **2 Keyword Extraction**

### **2.1 Extraction of the Initial Term Set for Keyword**

As an experimental attempt, we extracted the keywords of Program Committee members of SemAnnot 2004 Workshop (There are 28 members including chair persons). First, we need to acquire Web pages that contain information of respective committee members and their mutual relationships. A simple way of acquiring those Web pages is to use a search engine. It is reasonable to use a search engine because it can search many Web pages in less than a few seconds. It also tracks the temporal variance of the Web. In this experiment, we used Google<sup>5</sup>, which currently addresses data from 4 billion Web pages.

We first put each person's full name to a search engine (name is quoted with double quotation such as "Siegfried Handschuh") and retrieve documents related to each person. From the search result, we used the top 10 documents per person as the initial documents that might contain personal keywords.

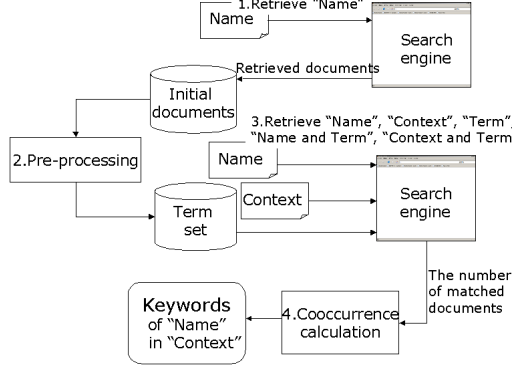
The search result documents include not only html files but also other file types such as .pdf, .doc, .xls, .ppt. In this experiment, we used only html files. Furthermore, we did not use metadata indicators in an html file such as META tags and RDF. In the future, we are planning to use other file types along with html files that already have been attached metadata.

The html files, at to a maximum of 10 files per person, are acquired from the initial documents of each person. They are pre-processed with html-tag deletion and part-of-speech tagging (POS). Then, the term set for keyword extraction is extracted from pre-processed html files using the term extraction tool, Termex [14]. Termex extracts terms

---

<sup>5</sup> <http://www.google.com>

from POS data based on statistical information of conjunctions between parts of speech. Termex<sup>6</sup> can also extract nominal phrases that include more than two nouns such as “Annotation tool”. After the whole procedure of extracting the term set, we extracted about 1000 terms per person on the average. The relevant keyword of each person is chosen from these terms. Figure 1 shows steps of the proposed keyword extraction.



**Fig. 1.** Procedure of keyword extraction

## 2.2 Keyword Extraction Using Co-Occurrence Information

Because the term set includes both relevant and irrelevant terms for personal information, we need to evaluate the relevance of term as a personal keyword. This subsection explains the scoring method that gives relevance as a personal keyword to the term.

**Term relevance based on Co-Occurrence** The simple approach to measure term relevance as a personal keyword is to use co-occurrence. In this paper, we define co-occurrence of two terms as term appearance in the same Web page. If two terms co-occur in many pages, we can say that those two have a strong relation and one term is relevant for another term. This co-occurrence information is acquired by the number of retrieved documents of a search engine result. For example, assume we are to measure the relevance of name  $N$  (e.g. “Siegfried Handschuh”) and term  $w$  (e.g. “Annotation”). Here,  $w$  is the term in the term set  $W$  extracted from the initial documents of the person named “ $N$ ”. We first put a query, “ $N$  and  $w$ ”, to a search engine and obtain the number of retrieved documents that is denoted by  $|N \text{ and } w|$ . We continuously apply a query, “ $N$ ” and “ $w$ ”, and obtain the number of retrieved documents for each,  $|N|$  and  $|w|$ . Then, the relevance between the name  $N$  and the term  $w$ , denoted by  $r(N, w)$ , is

<sup>6</sup> Termex can be used for both Japanese and English POS data



approximated by the following Jaccard coefficient.

$$r(N, w) = \frac{|N \text{ and } w|}{|N| + |w| - |N \text{ and } w|}$$

This Jaccard coefficient captures the degree of co-occurrence of two terms by their mutual degree of overlap.

**Keyword of person** As described in a previous subsection, the term set of a person is extracted from various Web pages. Although the Web pages contain a person’s name in the text, each page may contain personal information in different contexts. For example, imagine that one person, named “Tom”, is both a researcher and a artist, we can expect that his name may appear not only in academic-related pages, but also in other pages related to his art activities. Even among his academic-related pages, there might be different pages depending on his acquaintances, affiliations, and projects. In this way, different Web pages reflect different contexts of a person. Here, we introduce the notion of a context to extract the keyword that captures the context of a person.

To extract the keyword in relation to a certain context, we must estimate the relevance between the term and the context. If we replace the name  $N$  with the context  $C$  in the relevance,  $r(N, w)$ , we can obtain the relevance between context  $C$  and term  $w$ ,  $r(C, w)$ , in the same manner. Then, the relevance of person  $N$  and term  $w$  in the context  $C$ , denoted by  $score(N, C, w)$ , is calculated as the following.

$$score(N, C, w) = \frac{r(N, w)}{MAX(r(N, w))} + \alpha \frac{r(C, w)}{MAX(r(C, w))}$$

$$(\frac{r(N, w)}{MAX(r(N, w))} > threshold)$$

Therein,  $\alpha$  denotes the relevance between the person and the context. For example, we can use  $r(N, C)$  as  $\alpha$ .  $MAX(r(X, Y))$  is the maximum value of the Jaccard coefficient in the term set  $W$ . We define the *threshold* for  $r(N, C)$  to exclude terms that are not relevant for a person, but that have strong relation to the context. *threshold* is decided based on heuristic method. The term  $w$  with the higher  $score(N, C, w)$  is considered to be a more relevant keyword for person  $N$  in context  $C$ .

Regarding the “Tom” example, if we set “Art” as the context, we can get keywords related to his art activities. Alternatively, if we include his research project name as the context, keywords related to his project would be acquired.

**Keywords showing a relation between persons** If we consider the relation between two persons in terms of their contexts, one person can be regarded as a part of the context of another person. Hence, we can apply the previous formula to keyword extraction of the relations among persons as follows:

$$score(N1, N2, W) = \frac{r(N1, w)}{MAX(r(N1, w))} + \beta \frac{r(N2, w)}{MAX(r(N2, w))}$$

$$(\frac{r(N1, w)}{MAX(r(N1, w))}, \frac{r(N2, w)}{MAX(r(N2, w))} > threshold)$$

Therein,  $N1$  and  $N2$  denote each person's names in the relation.  $\beta$  is the parameter of relevance between persons, such as  $r(N1, N2)$ . This formula shows the relevance of person  $N1$ 's term  $w$  in relation to person  $N2$ .

As there are many contexts of a person, the relations among persons also have a variety of contexts. For example, the relation of two persons in the academic field might be coauthors, have the same affiliation, the same project; they may even be friends. The relevance of person  $N1$ 's term  $w$  in relation to person  $N2$  in the context  $C$ ,  $score(N1, N2, C, w)$ , is given as follows:

$$score(N1, N2, C, w) = score(N1, N2, w) + \gamma \frac{r(C, w)}{MAX(r(C, w))}$$

Therein,  $\gamma$  is the parameter of relevance between the persons and the context, such as  $r(N1 \text{ and } N2, C)$ .

### 3 Keyword Analysis for Personal Metadata

#### 3.1 Personal Metadata in Keywords

As an example of extracted keywords, Table. 1 shows higher-ranked extracted keywords of "Siegfried Handschuh". Each column in the table shows higher-ranked keywords based on Term Frequency Inverse Document Frequency (TFIDF), co-occurrence without the context, and co-occurrence with the context, respectively, from the left column.

In TFIDF-based keywords, we can find keywords that are related to the person such as "annotation" and "semantic". Nevertheless, there are many irrelevant words including general words. Because TFIDF is based on the frequency of word appearances in a text, it is difficult for a word to become higher-ranked in terms of relevance with another word. On the other hand, in co-occurrence-based keywords, general words are excluded and relevant words of each person appear in the rank list.

As explained in the previous section, the context can be considered in the keyword extraction. In this experiment, we used "Semantic Web" as the context. With this context, keywords are chosen in relation to one's activity about the Semantic Web. In the column of "Co-Occurrence with the context", we can find that context-related keywords come to appear in the rank list. The order of higher-ranked keywords also changes in relation to the context.

The column at the right side shows a property label for each keyword in "Co-Occurrence with the context". Considering a correspondence to existing personal metadata vocabularies such as FOAF, we have defined six property labels: Name (N), Technical term (T), Event (E), Organization (O), Project (P), URL. In order to analyze what kind of property is included in keywords, we annotated a property label to higher-ranked keywords of each person. Thereby, we acquired 1646 labeled keywords in total (about 60 keywords per person on average).

Table. 2 shows the distribution of property labels. Nearly half of higher-ranked keywords are occupied with names. Notwithstanding, it is noteworthy that other properties such as organizations and projects also appear to a certain degree. In particular, as shown on the right side column, the properties for each person are distributed in a balanced manner. This distribution indicates that if we extract about 60 higher-ranked

**Table 1.** Higher-ranked keywords of “Siegfried Handschuh” using TFIDF and co-occurrence-based method

TFIDF	Co-Occurrence (without the context)	Co-Occurrence (with the context “Semantic Web”)	Property
Semantic	Siegfried Handschuh	Siegfried Handschuh	N
Siegfried Handschuh	Ljiljana Stojanovic	Ljiljana Stojanovic	N
Office	Nenad Stojanovic	Nenad Stojanovic	N
annotation	Marc Ehrig	Steffen Staab	N
Person	Julien Tane	Marc Ehrig	N
Web	Steffen Staab	Julien Tane	N
Karlsruhe	Daniel Oberle	Daniel Oberle	N
Konstanz	Valentin Zacharias	Valentin Zacharias	N
E223	Andreas Hotho	Andreas Hotho	N
CREAM	relational metadata	<b>Semantic Web</b>	T
karlsruhe.de	annotation of web pages	relational metadata	T
message	Knowledge Markup	annotation of web pages	T
Inf.wiss	Large Scale Semantic Web	Knowledge Markup	T
knowledge	automatic CREAtion of Metadata	Large Scale Semantic Web	T
Webmaster	Annotation Workshop	Knowledge Markup Workshop	E
Appointment	Knowledge Markup Workshop	<b>International Semantic Web Conference</b>	E
AIFB	KCAP	KCAP	E
Katarina Stanoevska	AIFB	AIFB	O
Beat Schmid	University of Karlsruhe	University of Karlsruhe	O
Alexander Maedche	OntoAgents	OntoAgents	P

keywords of one person, on average we can obtain about 30 names of his acquaintance, 2 or 3 related organizations, and 1 or 2 projects. These numbers nearly match our research activity and show the possibility of using keywords for personal metadata. In this analysis, we took many keywords together as “technical terms”. If we classify each keyword more precisely, we could discover other personal metadata in keywords.

### 3.2 Personal Metadata in the Web

To further explore the possibility of personal metadata extraction from the Web, we analyzed which Web pages include a higher-ranked keyword. First, we classified all 280 Web pages (10 per person) that were used to extract the initial term set. Thereby, we prepared the 11 categories shown in Table. 3. “Personal page” includes personal Web pages of the affiliation or one’s own domain. “Other page” includes uncategorized pages and non-html pages such as .pdf and .ppt files. “Event page” includes conference, workshop, and meeting pages. ML log is the email exchanged in a mailing list. DBLP<sup>7</sup> is the online bibliography of Computer Science papers. As seen in the table, “Personal page” is the most dominant type of Web page. Because a person’s name was used as a query, it is natural that we obtain a personal page in a search result.

<sup>7</sup> <http://www.informatik.uni-trier.de/~ley/db/>

**Table 3.** Classification of the Web page type

Web Page	Number
Personal page	73 (26.0%)
Other page	42 (15.0%)
Event	32 (11.4%)
ML log	27 (9.6%)
Online paper	26 (9.2%)
DBLP	22 (7.8%)
Organization	17 (6.0%)
Project	16 (5.7%)
Book	11 (3.9%)
Publication list	8 (2.8%)
Weblog	6 (2.1%)
Total	280

**Table 2.** Distribution of properties labeled to higher-ranked keywords

Property	Number	Per person
Name	767 (46.5%)	27.3
Technical term	613 (37.2%)	21.8
Event	105 (6.3%)	3.7
Organization	73 (4.3%)	2.6
Project	48 (2.5%)	1.7
URL	40 (2.4%)	1.4
Total	1646	

**Table 4.** Distribution of each keyword property to each Web page type

Web page	Name	Technical Term	Event	Organization	Project	URL
Personal page	<b>234 (19.3%)</b>	<b>199 (24.0%)</b>	<b>31 (24.0%)</b>	<b>35 (36.8%)</b>	<b>30 (44.1%)</b>	<b>14 (25.9%)</b>
Other page	42 (3.4%)	15 (1.8%)	4 (3.1%)	3 (3.1%)	1 (1.4%)	2 (3.7%)
Event	<b>223 (18.3%)</b>	<b>171 (20.6%)</b>	<b>29 (22.4%)</b>	<b>25 (26.3%)</b>	1 (1.4%)	<b>14 (25.9%)</b>
ML log	165 (13.6%)	122 (14.7%)	11 (8.5%)	<b>16 (16.8%)</b>	<b>8 (11.7%)</b>	<b>8 (14.8%)</b>
Online paper	12 (0.9%)	33 (3.9%)	4 (3.1%)	1 (1.0%)	1 (1.4%)	2 (3.7%)
DBLP	<b>314 (25.9%)</b>	<b>189 (22.8%)</b>	<b>38 (29.4%)</b>	0	<b>11 (16.1%)</b>	0
Organization	66 (5.4%)	45 (5.4%)	4 (3.1%)	8 (8.4%)	5 (7.3%)	9 (16.6%)
Project	46 (3.7%)	13 (1.5%)	0	5 (5.2%)	5 (7.3%)	5 (9.2%)
Book	18 (1.4%)	7 (0.8%)	1 (0.7%)	0	0	0
Publication list	85 (7.0%)	24 (2.8%)	6 (4.6%)	1 (1.0%)	1 (1.4%)	0
Weblog	7 (0.5%)	10 (1.2%)	1 (0.7%)	1 (1.0%)	5 (7.3%)	0
Total	1212	828	129	95	68	54

Table 4 shows which category of Web page a higher-ranked keyword belongs in (a keyword may appear in more than one category). Specifically examining each column, we find which kind of Web page each property is included in. Moving the focus to a row in the table, we can find what kind of property each Web page category includes.

Although name entities can be acquired most from “Personal page”, DBLP is also a good information resource to extract a name entity. DBLP contains coauthor information of a paper. Therefore, the extracted name is related to one’s acquaintance in a research activity. “Event page”, such as conference, workshop, is a information resource of various personal information. However, because the Event page is not specified to a certain person, “Personal page” gives more accurate information about each person.

Overall, “Personal page” is a good information source for personal metadata such as names, organizations, and projects. Event page and DBLP provide metadata that are

related to personal research activities such as coauthors, projects, and events including conferences and workshops.

## 4 Annotation of Personal Metadata

Our keyword extraction method can be applied to semantic annotation in following ways.

- **Annotation for Web page :** As our analysis showed, our personal keyword extraction method offers strong potential for personal metadata extraction from the Web. Extracted personal metadata can be applied to partially annotate the Web pages using metadata description framework such as the RDF[13]. Because metadata are given the relevancy in relation to a person, annotated Web pages can be used in many applications such as Information retrieval and Information integration. For example, using annotated Web pages, the search engine that supports the Semantic Web could answer to following question:
  - Who knows this person?
  - Who is involved in this project?
  - Who knows this research topic well?
  - Which pages include this person's information?
- **Annotation for Personal Metadata File :** Extracted personal metadata is used not only for annotating a Web page, but also for annotating a personal metadata file. As one emerging personal metadata standard, "Friend of a Friend", FOAF[5], defines an RDF vocabulary for expressing metadata about people, the relation among them, and the things they create and do. FOAF provides a way to create machine-readable personal documents on the Web, and to process them easily through merging and aggregating them. Because extracted metadata are easily incorporated in FOAF, we can facilitate the creation of FOAF documents.

This paper presents discussion of the importance of a person's context in keyword extraction. The context often defines the properties. Currently, there is no FOAF vocabulary to define a context. In addition to FOAF, there are many vocabularies and framework for personal metadata such as Topicmaps [9], RDF-vCard [16], Person class of DAML+OIL [15]. However, none of them address the notion of a personal context. One way to introduce a personal context to those metadata frameworks is to prepare schema that corresponds to respective contexts. Regarding the expression of personal metadata, we need further consideration to make the metadata expressive and usable.

## 5 Related works

Aiming at extracting and annotating personal metadata, our method is regarded as one of Information Extraction(IE) methods supporting a semantic annotation. Up to now, many IE methods rely on predefined templates and linguistic rules or machine learning techniques to identify certain entities in text documents[12]. Furthermore, they usually define properties, domains, or ontology beforehand. However, because we try to extract various information from different Web pages, we don't use predefined restrictions in the extraction.

Some previous IE researches have addressed the extraction and annotation of personal metadata. In [1], they propose the method to extract a artist information, such as name and date of birth, from documents and automatically generate his or her biography. They attempt to identify entity relationships, metadata triples (subject-relation-object), using ontology-relation declarations and lexical information. However, Web pages often include free texts and unstructured data. Thereby, capturing entity relationships becomes infeasible because of lacking regular sentences. Rather than focusing on the entity relationship, we find the entity in the Web pages based on the relevance in relation to a person.

In [6], they address the extraction of personal information such as name, project, publication in a specific department using unsupervised information extraction. It learns to automatically annotate domain-specific information from large repositories such as the Web with minimum user intervention. Although they extract various personal metadata, they don't consider the relevance of extracted metadata. Because extracted metadata in our method have the relevance, they can be used as reliable initial seeds for bootstrap learning for automatic annotation in their method.

Although the aim is not extracting personal metadata, in [11], they propose the method to extract a domain terminology from available documents such as the Web pages. This method is similar to our one in terms of that terminology are extracted based on the scoring measure. However, their measure is based not on the co-occurrence but on the frequency. Furthermore, they focus on the domain-specific terms rather than personal metadata and the method is domain dependent. In our method, we can capture the various aspects of personal metadata even from different domain resources using the notion of a context.

## 6 Future works and Conclusion

To apply our keyword extraction methods to personal metadata annotation, we must consider and solve following points in the future.

- **Evaluation of personal metadata :** One problem is that we are not sure that the extracted metadata are true. Although two terms co-occur in many Web pages, they might not have any relation. Therefore, someone should evaluate the propriety of a keyword as actual metadata. One approach to solve this problem would be an interactive annotation system[3]. Reusing and modifying a keyword as a candidate of personal metadata, a user can easily annotate personal metadata.
- **Entity recognition of keywords :** Another critical problem is to decide a certain keyword property. In our experiment, the property label was given manually to each keyword. However, it is not efficient to put a property to numerous extracted keywords. One approach to automatically decide the property of a keyword is to use techniques in the entity recognition research[4].
- **Privacy problem of information extraction from the Web :** A person sometimes does not know that his or her information is extracted from the Web only by name. Therefore, we should take care not to intrude on a user's privacy even in information extracted from the Web. We must clarify the use of the information only for useful services for a user.

The Web holds much personal information that can be used as personal metadata. This paper proposes a novel keyword extraction method to extract personal information from the Web. Our result showed the important possibility of using extracted keywords as personal metadata. Importantly, our method can capture the personal information in different contexts. This allows us to obtain various personal metadata.

Because the Web is such a large information resource, its information runs the gamut from useful to trivial. It presents the limitation that it must be publicly available on the Web. For further improvement of the proposed method, we must analyze “what” information of “who” in the Web, and its reliability.

## References

1. H. Alani et al. Automatic Extraction of Knowledge from Web Documents. *In Workshop of Human Language Technology for the Semantic Web and Web Services, 2nd International Semantic Web Conference*, Sanibel Island, Florida, USA, 2003.
2. T. Berners-Lee, J. Hender, O. Lassila. The Semantic Web. Scientific American, 2001.
3. F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks. User-system cooperation in document annotation based on information extraction. *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*, Springer Verlag, 2002.
4. H. Cunningham et al. GATE:A Framework and Graphical Development Environment for Robust NLP Tools and Application. *In Proceedings of the 40th Anniversary Meeting Assoc. for Computational Linguistics(ACL2002)*, East Stroudsburg, Pa., 2002.
5. Dan Brickley and Libby Miller. FOAF: the 'friend of a friend' vocabulary. <http://xmlns.com/foaf/0.1/>, 2004.
6. A. Dingli, F. Ciravegna, D. Guthrie, Y. Wilks. Mining Web Sites Using Unsupervised Adaptive Information Extraction. *In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, 2003.
7. L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. In *Doing Internet Research*, S. Jones, Ed. Sage, Thousand Oaks, CA, pp. 75–105, 1999.
8. J. Kahan and M. R. Koivunen. Annotea: An open rdf infrastructure for shared web annotations. *In Proceedings of the 10th International WWW Conference*, pp.623–632, 2001.
9. Lars Marius Garshol. Living with topic maps and RDF. <http://www.ontopia.net/topicmaps/materials/tmrdf.html>, 2003.
10. S. Staab, A. Maedche, and S. Handschuh. An Annotation Framework for the Semantic Web. *In Proceedings of 1st International Workshop MultiMedia Annotation*, 2001.
11. P. Velardi, M. Missikoff, R. Basili. Identification of relevant terms to support the construction of Domain Ontologies. *In ACL-EACL Workshop on Human Language Technologies*, Toulouse, France, 2001.
12. R. Yangarber and R. Grishman. Machine Learning of Extraction Patterns from Unannotated Corpora: Position Statement. *Workshop Machine Learning for Information Extraction*, IOS Press, Amsterdam, pp.76–83, 2000.
13. Resource Description Framework(RDF) Schema Specification. In *W3C Recommendation*, 2000.
14. <http://gensen.dl.itc.u-tokyo.ac.jp/win.html>
15. DAML Ontology Library. <http://www.daml.org/ontologies/>
16. Representing vCard Objects in RDF/XML. <http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/>

# Annotation of Heterogeneous Database Content for the Semantic Web

Eero Hyvönen, Mirva Salminen, and Miikka Junnila

University of Helsinki, Department of Computer Science  
Helsinki Institute for Information Technology (HIIT)  
`{firstname.lastname}@cs.helsinki.fi`  
<http://www.cs.helsinki.fi/group/seco/>

**Abstract.** This paper discusses the problem of annotating semantically inter-linked data that is distributed in heterogeneous databases. The proposed solution is a semi-automatic process that enables annotation of database contents with shared ontologies with little adaptation and human intervention. A technical solution to the problem based on semantic web technologies is proposed and its demonstrational implementation is discussed. The process has been applied in creating the content for the semantic portal MUSEUMFINLAND, a deployed Semantic Web application.

## 1 Introduction

A crucial question for the breakthrough of the Semantic Web approach is how easily the needed metadata can be created. Annotating data by hand is laborious, resource-consuming, and usually economically infeasible with larger datasets. Automation of the annotation process is therefore needed. This task is the more severe the more heterogeneous the data is. This paper addresses the problem of annotating heterogeneous and distributed data with a set of shared domain ontologies (within a single application domain). The problem is approached through a real-life case study by describing the annotation process developed for the MUSEUMFINLAND<sup>1</sup> [6, 8, 10] semantic portal. This application publishes cultural collection data from several heterogeneous museum databases in Finland.

We developed the annotation process for MUSEUMFINLAND in order to enable publication of museum collection item data on the Semantic Web. The goal of the annotation process is to transform the heterogeneous local databases into a global, syntactically and semantically interoperable knowledge base in RDF(S) format. The knowledge base is then stored into a common repository. This knowledge base conforms to a set of global domain ontologies, and the services provided by MUSEUMFINLAND to the end-users, i.e., view-based semantic search and browsing [5], are based on it<sup>2</sup>.

The users of the process are museum personnel who want to bring their collections into the Semantic Web. Though the process is originally designed for the use of museums, the same approach can be applied to other heterogeneous database contents that

---

<sup>1</sup> <http://museosuomi.cs.helsinki.fi>

<sup>2</sup> The (meta)data, ontologies and programs used in the process described in this paper are available as open source at <http://www.cs.helsinki.fi/group/seco/museums/dist/>.



need to be annotated with shared domain ontologies. We will discuss the issues that affect the applicability and the workload of the process, and give examples based on the MUSEUMFINLAND case.

The annotation process was designed to meet two requirements: First, new museum collections need to be imported into the MUSEUMFINLAND portal as easily as possible and with as little manual work and technical expertise as possible. Second, the museums should not be forced to change their cataloging conventions for creating collection item descriptions. For example, two museums may use different terms for the same thing. The system should be able to accept the different terms as far as the terms are consistently used and their local meanings — with respect to the global reference ontologies — are provided.

Figure 1 depicts the whole annotation process that consists of three major parts:

- 1. Syntactic Homogenization.** Since the data in museum databases is syntactically heterogeneous, the first step involves reaching syntactic interoperability by representing the database contents in a common syntax. A way of defining the common syntax is to specify an XML schema that all the different content providers can agree on. This task is simplified by the fact that the heterogeneous databases have a homogeneous domain: they contain cultural metadata about artifacts and historical sites, which means that the data items have similar features. For instance, all museum artifacts have features such as object type, material, place of usage, etc. This data can be exported from the different databases into a syntactically uniform XML form [12] (arrow on the left in figure 1).
- 2. Terminology Creation.** To define the meaning of the terms and linguistic patterns used in the XML representation (and in the databases), we need to connect them to the global ontological concepts shared by the portal content providers. The mapping from literal values to concepts is called a *terminology*. In MUSEUMFINLAND, the terminology is created with the help of a tool called Terminator (lower arrow in figure 1).

A problem in terminology creation is that the museums and catalogers use different vocabularies and describe their collection contents in differing manners. From a practical viewpoint, such local variance should be tolerated and should not impose terminological restrictions on other museums. In order to make MUSEUMFINLAND flexible with respect to variance in terminologies used at different museums, the terminology has been separated from the domain ontologies. In our approach, the museums can share globally agreed term definitions but also override them with their own local term definitions without any need to change the shared domain ontologies or global term definitions.
- 3. Annotation Creation.** During the annotation creation process the XML data containing the museum item descriptions is enriched with references to the ontological definitions. This process is based on the terminologies and makes the heterogeneous collection data semantically interoperable with respect to the set of underlying domain ontologies. In MUSEUMFINLAND, a tool called Annomobile has been created to automate the annotation process (arrow on the right in figure 1).

In the following, these three parts of the process are discussed in more detail.

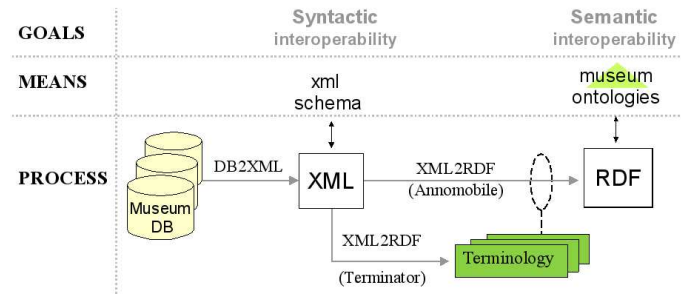


Fig. 1. The content creation process in MUSEUMFINLAND.

## 2 Syntactic Homogenization

The museum databases are both distributed and heterogeneous, i.e. the databases are situated in physically different places, the used database systems are made by different manufacturers, and their logical structure (schemes, tables, fields, etc.) may vary.

The first step of combining domain data from multiple sources is, thus, gaining syntactic interoperability. This task is highly system dependent. For example on the level of structure, combining collection data means that the collection record data fields meaning the same thing but under different labels in different databases, such as "name of object" and "object name", are identified as the same, common labels are given to the fields, and a common way of representing collection data is agreed upon.

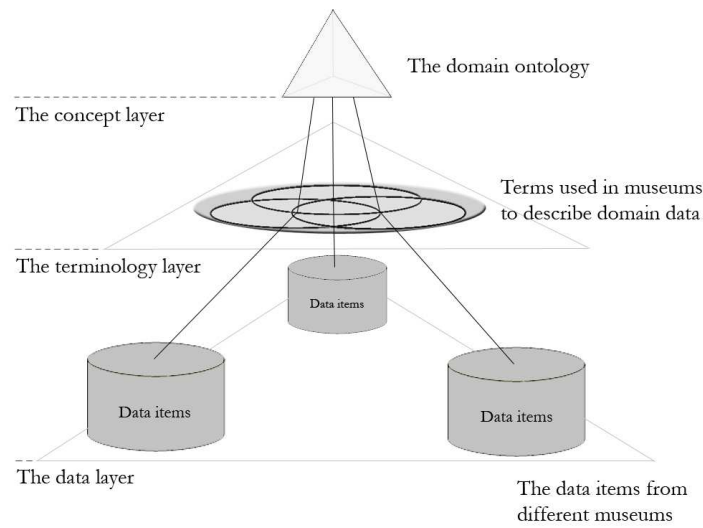
The combining can be done by agreeing on a shared presentation language for collection data. When the museums have agreed on this, the transmission, combination, and WWW publishing of the collections becomes significantly easier. In the MUSEUMFINLAND system, the combination of museum data at the structural level is based on a common XML schema. This schema is used to express the collection data to be published on the WWW. A simplified example of the XML can be found in [7].

The syntactic homogenization into XML makes the other steps of the process system independent, so that these steps don't have to be changed at all when new museums join MUSEUMFINLAND or old museums change their databases.

The transformation procedure from database to XML depends on the database schema and system at hand, and is described more in detail in [12]. For the portal version currently on the web, we created database to XML transformers for three different database systems used in three different museums.

## 3 Terminology Creation

A terminology defines a mapping between terms and concepts. This makes automation of the annotation process possible. Figure 2 illustrates the role of terminology as a mediating layer between the conceptual layer and the data layer. On the top is the concept layer that is described by a set of global domain ontologies. Under that is the terminology layer that contains all the terms used for describing different things that relate



**Fig. 2.** The mapping of data items to the domain ontology through the terminology layer

to the domain. The terminology layer is broader than the concept layer, since concepts can be expressed in various ways. Under the terminology layer is the largest of the layers, the data to be annotated. Terminologies used in different databases intersect on the terminology layer but may have non-overlapping parts as well.

A term on the terminology layer is usually used as a value in several data items at the data layer. It is therefore easier to map data items to concepts by using the terms than by mapping data items directly to concepts. When terms have been excessively annotated, the data itself can be annotated almost automatically.

In MUSEUMFINLAND a terminology is represented by a term ontology, where the notion of the term is defined by the class `Term`. The class `Term` has six properties: `concept`, `singular`, `plural`, `definition`, `usage` and `comment`. They are inherited by the term instances called *term cards*. A term card associates a term as a string with an URI in an ontology represented as the value of the property `concept`. Both `singular` and `plural` forms of the term string are stored explicitly for two reasons. First, this eliminates the need for Finnish morphological analysis that is complex even when making the singular/plural distinction. Second, singular and plural forms are sometimes used with different meaning in Finnish thesauri. For example, the plural term “operas” would typically refer to different compositions and the singular “opera” to the abstract art form. To make the semantic distinction at the term card level, the former term can be represented by a term card with missing singular form and the latter term with missing plural form. Property `definition` is a string representing the definition of the term. Property `usage` is used to indicate obsolete terms in the same way as the `USE` attribute is used in thesauri. Finally, the `comment` property can be filled to

store any other useful information concerning the term, like context information, or the history of the term card.

Two different methods were used in terminology creation:

#### 1. Thesaurus to Taxonomy Transformation

Some 6000 new term instances were created based on the Finnish cultural thesaurus MASA [9] that was converted into a domain ontology (taxonomy). A term card for each thesaurus entry was created and associated with the ontology class corresponding to the entry. For obsolete terms, the associated ontology resource can be found by the USE attribute value. The morphological tool MachineSyntax<sup>3</sup> was used for creating the missing plural or singular forms for the term cards.

#### 2. Term Ontology Population from Databases

New term cards are created automatically for unknown terms that are found in artifact record data. The created term cards are automatically filled with contextual information concerning the meaning of the term. This information helps the human editor to fill the `concept` property. For example, assume that one has an ontology *M* of materials and a related terminology *T*. To enhance the terminology, the material property values of a collection database can be read. If a material term not present in *T* is encountered, a term card with the new term but without a reference to an ontological concept can be created. A human editor can then define the meaning by making the reference to the ontology and also create new entries for own terms if needed.

For efficiency reasons, the new terms are ranked by their frequency of use, so that the human editor can annotate the most used terms, and leave the most infrequent terms unannotated. This way the editor's work amount in relation to the coverage of the term ontology is optimized.

Figure 3 depicts the general term extraction process in MUSEUMFINLAND. The process involves a local process at each museum and a global process at MUSEUMFINLAND. The tool Terminator extracts individual term candidates from the museum collection items presented in XML. The entity of one item is called an *XML card*. A human editor annotates ambiguous terms or terms not known by the system. The result is a set of new term cards. This set is included in the museum's local terminology and terms of global interest can be included in the global terminology of the whole system for other museums to use.

The global terminology consists of terms that are used in all the museums. It reduces the workload of individual museums, since these terms do not need to be included in local terminologies. The global terminology can be extended when needed. On the other hand, the local terminology is important because it makes it possible for individual museums to use and maintain their own terminologies.

The problem of the term creation approach described above is how to deal with free text descriptions. It is not very useful to regard field values that consist of long textual descriptions as single terms. For example "art poster" is a good term, but the term "A time-worn middle sized poster of a painting by Van Gogh" is not. This term probably wouldn't have any duplicates in the rest of the data, and annotation of the data item on

<sup>3</sup> [http://www.conexor.fi/m\\_syntax.html](http://www.conexor.fi/m_syntax.html)

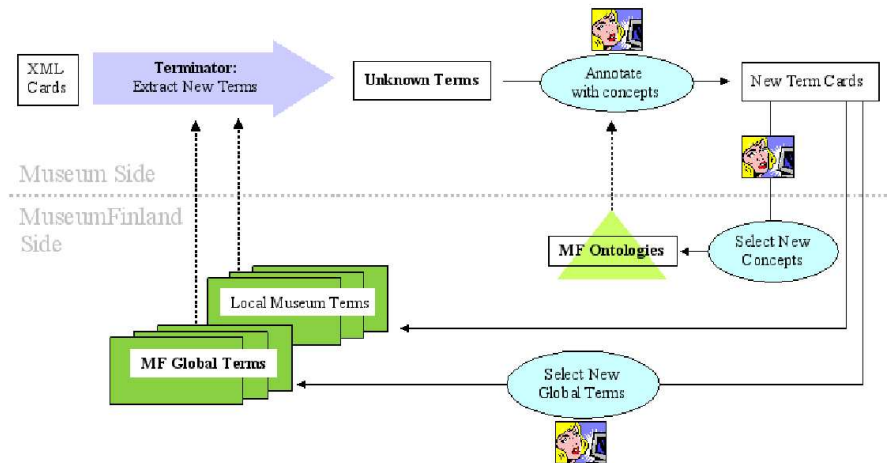


Fig. 3. Creating new term cards in MUSEUMFINLAND.

the data layer (cf. figure 2) instead of annotating it as a term would be as simple and more natural.

Sometimes the data in the databases is erroneous. For example spelling errors were common. In these cases a term card can be created for an erroneous term that has been excessively used, so that the semantic enrichment makes the right ontological links, even though the database data is not corrected.

## 4 Annotation Creation

The last step in the content creation process is the semi-automatic annotation, which makes the data semantically interoperable. This can be done when the database contents have been transformed into coherent XML form, and the terminology mappings have been created.

In this paper, semantic interoperability means that the terms used in describing the data have to be interpreted semantically in a mutually consistent way. This is done by linking literal data values on the XML level, called *features*, to the ontological concepts on the RDF level. In practice, the string-valued features that are expressed in the shared XML syntax are transformed into the Uniform Resource Identifiers (URI) of the corresponding classes and individuals in the ontologies.

The features of the data items fall in two categories: *literal features* and *ontological features*. Literal features are to be represented only as literal values on the RDF level. They are, for example, used in the user interface. Ontological features are values that need to be linked not only to literal values but also to ontological concepts (URI).

The XML to RDF transformation can be done by algorithm 1. Each ontological feature is associated with a separate domain ontology by the property-domain mapping. For example, the material values of artifacts are found from a domain ontology of

Let  $X$  be a set of XML cards with literal features  $L$  and ontological features  $P$ , having values  $V$  (terms) ;  
 Let  $O$  be a set of ontologies ;  
 Let *Property-domain mapping*  $d : P \rightarrow O$  map each ontological property to a domain ontology ;  
 Let *Terminology mapping*  $t : V, O \rightarrow S$  map the XML card feature values  $V$  of the ontological property  $P$  to the classes and individuals  $S$  in  $O$  ;  
**Result:** A set  $R$  of RDF triples.

```

 $R := \emptyset$ ;
foreach XML card  $x \in X$  do
  Create an RDF card instance  $i$ ;
  foreach feature  $f \in P \cup L$  having value  $v$  do
     $R := \{ \langle i, f\text{-literal}, v \rangle \} \cup R$ ;
    if  $f \in P$  then
       $R := \{ \langle i, f, s \rangle \} \cup R$ , where  $s = t(v, o)$  is a collection of resources
      in the underlying domain ontology  $o = d(f)$  so that  $s$  is found through
      terminology mapping;
    end
  end
end

```

**Algorithm 1:** Creating ontological annotations.

materials, place of usage feature values are found from a location ontology, and so on. This mapping can be used for disambiguating homonymous terms referring to resources in different ontologies. The algorithm creates for each XML card feature  $f$ , represented as an XML element, a corresponding RDF triple with a corresponding predicate name  $f\text{-literal}$  and a literal object value. For ontological features, an additional triple is created whose predicate name is the name of the feature and the object value consists of URIs to the possible resources that the literal feature value may refer to according to the terminology  $t$ .

Algorithm 1 is the basis of the semi-automatic annotation creation tool Annomobile (cf. figure 1) in MUSEUMFINLAND. Annomobile gets XML cards as input and produces the corresponding annotations in RDF format as output. The annotations follow an annotation schema that is expressed by an RDF Schema.

We have chosen fifteen different fields from the museum collection data records to be shown in the portal to the end-user. Nine of these features are ontological and hence linked to domain ontologies during the annotation process. The nine ontological features and their ranges, i.e. the seven domain ontologies to which the features are linked to, are presented in table 1. The ontologies (ranges) define the domains on which the term disambiguation is based on. The ontological features and domain ontologies are described in some more detail in [8].

When mapping ontological feature values to URIs in domain ontologies, two problem situations may occur:

Ontological feature	Ontology/Range	Ontological feature	Ontology/Range
<b>Object type</b>	Artifacts	<b>Material</b>	Materials
<b>Creator</b>	Actors	<b>Location of creation</b>	Locations
<b>Time of creation</b>	Times	<b>User</b>	Actors
<b>Location of usage</b>	Locations	<b>Situation of usage</b>	Situations
<b>Collection</b>	Collections		

**Table 1.** The nine ontological features of collection items and seven ontologies used in MUSEUMFINLAND.

**Unknown values.** The feature value may be unknown, i.e. there are no applicable term card candidates in the terminology. The solution to this is to map the feature value either to a more general term, e.g. to the root of the domain, or to an instance that represents all unknown cases. For example, if one knows that an artifact is created in some house in the city of Helsinki, but the address is unknown, one can create an instance called “unknown house” which is part of Helsinki and annotate the item with this instance.

**Homonyms.** The problem of homonymous terms occurs only when there are homonyms within the content of one domain ontology. The simple solution employed in our work is to fill the RDF card with all potential choices, inform the human editor of the problem, and ask him to remove the false interpretations on the RDF card manually. Our first experiments seem to indicate, that at least in Finnish not much manual work is needed, since homonymy typically occurs between terms referring to different domain ontologies. However, the problem still remains in some cases and is likely to be more severe in languages like English having more homonymy.

Table 2 shows some statistical results were obtained from the annotation process of building MUSEUMFINLAND. The content material came from three heterogeneous collection databases in three different museums. The number of collection items in the material totaled 6046, and every item had nine fields on average that needed to be linked to ontological concepts through the annotation process. All these nine fields could contain multiple literal values, all of which should be linked to different ontological concepts. For example, the place of usage field could contain several location names.

The table indicates that homonyms do not occur too often in the data. It can be seen also that in most cases the homonyms belong to different domains. Hence, the simple disambiguation scheme based on feature value domains worked well in practice and not much human editing was needed after using Annomobile.

	Museum 1	Museum 2	Museum 3
<b>Total of annotated museum items</b>	1354	1682	3010
<b>Items with homonyms (total)</b>	567	388	448
<b>Items with homonyms disambiguated</b>	424	332	334
<b>Items with homonyms not disambiguated</b>	143	56	114

**Table 2.** Results from annotating data with Annomobile in MUSEUMFINLAND.

## 5 Discussion

### 5.1 Lessons Learned

A general problem encountered in the content work was that the original museum collection data in the databases was not systematically annotated. Various conventions are in use in different museum systems and museums. Automatic annotation was relatively easy when descriptions in the database tables are done in a consistent manner using thesauri and without inflecting words. However, the descriptions in many cases were given in more or less free text. For example, use of free text was common in the data fields describing the techniques by which the artifacts were created. Furthermore, individual catalogers have used different terms and notations in cataloging. To handle these cases, the free text was tokenized into words or phrases which were then interpreted as keywords. This approach works, if term cards with ontological links are created from these keywords, and was adopted to both Terminator and Annomobile. The drawback here is, that if the vocabulary used in the free text is large, also the number of new term cards will be high and the manual workload in their annotation will be considerable. The vocabulary used in the MUSEUMFINLAND case, however, mostly conforms to the entries in the Finnish cultural thesaurus MASA, and this approach seems to be feasible. The homonymy problem is most severe in general free text description fields, since they are most prone to consist of conceptually general data where disambiguation cannot be based on the ontology to which the text field is related. Nonetheless, the Terminator and Annomobile tools proved out to be decent programs, annotating the data well enough for the purposes of the project.

### 5.2 Related Work

Lots of research has been done in annotating web pages or documents using manual or semi-automatic techniques and natural language processing. CREAM and Ont-O-Mat [1] and the SHOE Knowledge Annotator [3] are examples of such work.

Stojanovic et al. [14] present an approach that resembles ours in trying to create a mapping between a database and an ontology, but they haven't tackled the questions of integrating many databases or using global and local terminology to make the mapping inside a domain. Also [2] addresses the problems of mapping databases to ontologies, but their way of doing the mapping is very different from ours; in deep annotation the data is kept in the database, and the data is dynamically fetched from the database. Also, in our process we annotate the data through terminology, while deep annotation uses the database structure.

Also others have used the distinction of different layers of domain data and knowledge (figure 2). In [13] the concepts-terms-data model has been used to define different elements used for creating an ontology out of a thesaurus.

The idea of annotating cultural contents in terms of multiple ontologies has been explored also, e.g. in [4]. Other ontology-related approaches used for indexing cultural content include Iconclass<sup>4</sup>[15] and the Art and Architecture Thesaurus<sup>5</sup> [11].

---

<sup>4</sup> <http://www.iconclass.nl>

<sup>5</sup> [http://www.getty.edu/research/conducting\\_research/vocabularies/aat/](http://www.getty.edu/research/conducting_research/vocabularies/aat/)



As far as we know, our annotation process is the first one to provide semantic enrichment through terminological interoperability among several content providers, and to the semantic extent described in this paper. The output of the process, i.e. the annotated museum collection items, have been published in a semantic web portal MUSEUM-FINLAND for all Internet users to enjoy.

## References

1. S. Handschuh, S. Staab, and F. Ciravegna. S-cream - semi-automatic creation of metadata. In *Proceedings of EKAW 2002, LNCS*, pages 358–372, 2002.
2. S. Handschuh, S. Staab, and R. Volz. On deep annotation. In *Proceedings of International World Wide Web Conference*, pages 431–438, 2003.
3. J. Hefflin, J. Hendler, and S. Luke. Shoe: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1999.
4. L. Hollink, A. Th. Schreiber, J. Wielemaker, and B.J. Wielinga. Semantic annotation of image collections. In *Proceedings KCAP'03, Florida*, October, 2003.
5. E. Hyvönen, S. Saarela, and K. Viljanen. Application of ontology-based techniques to view-based semantic search and browsing. In *Proceedings of the First European Semantic Web Symposium, May 10-12, Heraklion, Greece*, pages 92–106. Springer-Verlag, Berlin, 2004.
6. E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen. Finnish Museums on the Semantic Web. User's perspective on MuseumFinland. In *Selected Papers from an International Conference Museums and the Web 2004 (MW2004), Arlington, Virginia, USA*.
7. E. Hyvönen, M. Junnila, S. Kettula, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen. Publishing collections in the Finnish Museums on the Semantic Web portal – first results. In *Proceedings of the XML Finland 2003 conference. Kuopio, Finland*.
8. E. Hyvönen, M. Salminen, S. Kettula, and M. Junnila. A content creation process for the Semantic Web, 2004. Proceeding of OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments, May 29, Lisbon, Portugal.
9. R. L. Leskinen, editor. *Museoalan asiasanasto*. Museovirasto, Helsinki, Finland, 1997.
10. E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. OntoViews — a tool for creating semantic web portals. In *Proceedings of the Third International Semantic Web Conference, Nov 7-11, Hiroshima, Japan*. Springer-Verlag, Berlin, 2004.
11. T. Peterson. Introduction to the Art and Architecture thesaurus, 1994. <http://shiva.pub.getty.edu>.
12. V. Raatikka and E. Hyvönen. Ontology-based semantic metadata validation. Number 2002-03 in HIIT Publications, pages 28–40. Helsinki Institute for Information Technology (HIIT), Helsinki, Finland, 2002. <http://www.hiit.fi/publications/>.
13. D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: the agrovoc example. *Journal of Digital Information*, (4), 2004.
14. L. Stojanovic, N. Stojanovic, and R. Volz. Migrating data-intensive web sites into the semantic web. In *Proceedings of the ACM Symposium on Applied Computing SAC-02, Madrid, 2002*, pages 1100–1107, 2002.
15. J. van den Berg. Subject retrieval in pictorial information systems. In *Proceedings of the 18th international congress of historical sciences, Montreal, Canada*, pages 21–29, 1995.

# Automated OWL Annotation Assisted by a Large Knowledge Base

Michael Witbrock, Kathy Panton, Stephen L. Reed, Dave Schneider, Bjørn Aldag,  
Mike Reimers and Stefano Bertolo

{witbrock, panton, sreed, daves, aldag, mreimers, bertolo}@cyc.com

**Abstract.** Widespread adoption of the semantic web depends critically on lowering the “barriers to entry” facing document producers. We describe a system that applies automatic partial parsing of web pages into the representations of the large ResearchCyc ontology, combines this with convenient mixed initiative knowledge capture, and produces an OWL annotated document as output. Semantic web publishers can then use this document as a starting point for more elaborate, manual annotation.

## Introduction

The rapid adoption of the World Wide Web, in its initial form, was driven in part by the ease with which content could be produced; although specialized tools and techniques quickly evolved, web pages could be produced, reasonably conveniently, by anyone with a text editor and an hour to read a description of the available HTML tags. Semantic markup in languages like OWL has the potential to vastly increase the utility of web content, but describing the logical content of a document is far from straightforward, even without the requirement that that description be done in an XML-based markup language.

In addition to the simple tools and syntax required for HTML authoring, the ready availability of example pages with mark-up produced by others further flattened the already shallow learning curve for Web authoring. Providing such examples for the semantic web would have similar utility but is not as obviously straightforward. While the syntax of OWL is consistent, the conceptual tag set to be used is highly dependent on the domain of the document, and, even within a domain, is set only by convention. Rather than require prospective authors to identify the appropriate vocabulary, complex XML syntax, and relevant set of example documents before semantic annotation can begin, it seems worthwhile to provide a tool that, while imperfect, can make an initial, automatic pass at annotating a document. From that rough annotation, it should be more straightforward for human content providers to incrementally improve the representation of page content as they increase their understanding of relatively narrow components of the relevant ontology and OWL syntax.

In this paper, a system, based on Cyc, is described that can automatically produce initial OWL annotations of arbitrary text documents. This is done in the vocabulary of the OpenCyc scaffolding ontology, which is freely available<sup>1</sup> and freely usable. The annotation process takes advantage of existing Cyc system components for automated text analysis and guided knowledge entry, as well as newly-created components for interactive disambiguation using natural language and reduction of internal CycL representations to the OWL languages. Interactive components of the process are optional, and annotation can proceed wholly automatically.

## Document Analysis

The Cyc OWL annotation system operates in two phases. First, the page is read and as much of the content as possible is represented in the CycL language. Second, the OWL export component of Cyc, developed as part of the DARPA DAML project, is used to generate the appropriate annotation file.

---

<sup>1</sup> <http://www.cyc.com/2004/06/04/cyc>

**BBC NEWS** WORLD EDITOR

Last Updated: Sunday, 29 February, 2000

E-mail this to a friend

## Spain police 'foil Eta bo

**Two suspected members of Basque separatist group Eta have been arrested as they headed to Madrid in a truck laden with explosives.**

Spanish police said they were arrested early on Sunday about 140km outside the Spanish capital, with 500kg of explosives hidden in the vehicle.

Government officials believe the men were planning an attack in the lead-up to Spain's general election.

Eta has killed more than 800 people in its campaign since the late 1960s.

Earlier this month the group said it was extending its campaign against Spanish tourist targets from the summer season to year round attacks.

**“ More than 500 kgs of explosives ... was a cargo that would have caused an explosion with very serious consequences ”**

Interior Minister Angel Acebes

The BBC's Katya Adler, in Madrid, says Spain's anti-terrorist

```

<AttackOnObject rdf:ID="AttackOnObject0413">
  <rdfs:label xml:lang="en">attack on object 0413</rdfs:label>
  <guid>96b8ee54-13e8-41d9-9b21-e518bbe00e6e</guid>
  <in-UnderspecifiedContainer rdf:resource="#LeadUp415" />
</AttackOnObject>
<Individual rdf:ID="LeadUp415">
  <rdfs:label xml:lang="en">lead up 415</rdfs:label>
  <guid>d013f98c-13e8-41d9-8277-e9bc8abd0e93</guid>
  <to-UnderspecifiedLocation rdf:resource="#Election0407" />
</Individual>
<Election rdf:ID="Election0407">
  <rdfs:label xml:lang="en">election 0407</rdfs:label>
  <guid>d691721c-13e8-41d9-9a6b-cefe0a553dfe</guid>
</Election>
<MakingAPlan rdf:ID="MakingAPlan0397">
  <rdfs:label xml:lang="en">making A plan 0397</rdfs:label>
  <guid>41dd4d62-13e8-41d9-804e-96b90890aa3e</guid>
  <performedBy rdf:resource="#AdultMaleHuman0411" />
</MakingAPlan>
<AdultMaleHuman rdf:ID="AdultMaleHuman0411">
  <rdfs:label xml:lang="en">adult male human 0411</rdfs:label>
  <guid>66cc1a4a-13e8-41d9-9d18-820d1b1d46bb</guid>
</AdultMaleHuman>
<Schedule rdf:ID="Plan1">
  <rdfs:label xml:lang="en">plan 1</rdfs:label>
  <guid>e1722afa-13e8-41d9-9057-94ac2bca1e8c</guid>
  <scheduledEvents rdf:resource="#Event1" />
</Schedule>
<Event rdf:ID="Event1">
  <rdfs:label xml:lang="en">event 1</rdfs:label>
  <guid>13fef4c2-13e8-41d9-9848-ce8a1032ef0d</guid>
</Event>
</rdf:RDF>

```

**Figure 1: The Cyc Document Annotator assists organizations and individuals interested in adapting their document production processes to the Semantic Web. By providing an approximate OWL annotation of an existing document, the system simplifies the initial learning curve, allowing editing to improve the annotation, to replace the complex task of manually annotating a document from scratch. Interoperability is supported by annotation using the more than 60,000 freely usable terms in the OpenCyc scaffolding ontology.**

The OWL export component of the system is described in more detail later, but the core of the annotation system depends on Cyc's imperfect but growing ability to interpret free text into a detailed logical representation in CycL. This is provided by combined application of Cyc's natural language processing subsystem, disambiguation dialogue, and the Factivore, a highly usable knowledge-driven knowledge acquisition interface.

## Parsing into the CycL Logical Language

CycL is a fully higher order and modal knowledge representation formalism<sup>2</sup>, which makes it suitable for representing a wide range of natural language constructions. Cyc also allows the partition of knowledge into separate ‘microtheories’ arranged in a subsumption hierarchy which enables the consistent management of contradictory information and the representation of context (e.g. statement of background assumptions). The strategy followed by our annotation systems is to parse input documents, rendering as much as currently possible into a CycL representation, to provide users with the opportunity, but not the necessity, to interactively disambiguate and elaborate the CycL representation, and then to project the resulting assertions onto the subset of representations allowed by the OWL language, yielding an XML annotation file.

## Extracting the Text Content of target web pages

We use two packages from the Apache Project (CyberNeko,<sup>3</sup> and Xerces<sup>4</sup>) to convert an HTML document into a Document Object Model (DOM) as a Java Object. The application traverses the DOM tree, extracting the web page title, meta-description, and text leaf nodes. This will provide us with the ability, in future versions of the annotator, to tailor its focus onto salient content and cause it to ignore distractions (e.g. sidebars, menu items, advertisements, navigation links, and so forth often found with news articles). This will be a substantial improvement over simple web page text extractors, which apply the simple algorithm of stripping out HTML tags, thereby omitting most cues to salience and noise.

## Chunking Input into Sentences, Phrases and Words

The second stage of the parsing pipeline populates a “TextDocument” object with sentences, phrases and words obtained from the web page’s DOM. Currently, we use the LINGUA sentence splitting module<sup>5</sup> to extract whole sentences from text strings, and the remaining text fragments are then organized as phrases and words. All our web page annotation experiments to date have been conducted on English language documents, but, since the character set used for parsing is UTF-8, it should in principle be straightforward to apply this step of processing to other languages. Full processing of other languages will depend on extending the Cyc Lexicon beyond its rudimentary coverage outside English, and extending the segmentation and syntactic parsing infrastructure to handle a wider range of syntactic phenomena.

## Natural Language Knowledge and English Parsing

Natural language processing in Cyc is supported by the Cyc Lexicon, an increasingly comprehensive collection of syntactic and semantic knowledge about English, and a framework in which knowledge about other languages can be embedded. The table below gives some indication of the current coverage.

	Noun	Verb	Adjective
CycL terms representing Lexemes	15450	4454	4716
Denotations	14442	1838	1640
Semantic Translation Patterns	464	3178	1787

CycL terms representing lexemes include `Burger-TheWord` and `Of-TheWord`, representing the English words “burger” and “of”, respectively; denotations connect word senses to KB concepts. For example,

```
(denotation Burger-TheWord CountNoun 0 HamburgerSandwich)
```

means that “burger”, when used in its first CountNoun sense, refers to a hamburger sandwich;

---

<sup>2</sup>The Cyc inference engine however currently only supports the first order fragment and some of the second order and modal extensions.

<sup>3</sup> <http://www.apache.org/~andyc/neko/doc/html/>

<sup>4</sup> <http://xml.apache.org/xerces-j/>

<sup>5</sup> <http://people.brandeis.edu/~matthewg/cpan-lingua.html>

```
(verbSemTrans Venerate-TheWord 0 TransitiveNPCompFrame
  (feelsTowardsObject :SUBJECT :OBJECT Reverence highAmountOf)),
```

means that the word “venerate”, when used as the verb in a transitive verb frame taking an NP complement, should be understood in the Cyc logical language, CycL, as meaning that the agent denoted by the subject of the sentence feels a high degree of reverence towards the thing denoted by the object of the sentence.

Similarly,

```
(nounSemTrans Bride-TheWord 0 GenitiveFrame
  (and
    (isa :NOUN FemaleHuman)
    (isa ?W WeddingEvent-Entire)
    (eventHonors ?W :NOUN)
    (eventHonors ?W :POSSESSOR)))
```

tells Cyc that, for example, “Frankenstein’s Bride” or “the bride of Frankenstein” should be interpreted as meaning that the bride is a female person, and that some wedding happened that honored both the bride and Frankenstein.

The third stage of the document annotation pipeline iterates over the sentences and phrases in the TextDocument object. Phrases are treated as whole sentences on the first pass. Each sentence is parsed by Cyc’s natural language parsing system, resulting in a list of CycL logical sentences. If the list is empty, then Cyc could not determine a semantic interpretation that covered the entire sentence, and if more than one CycL sentence is returned, then Cyc found one or more ambiguous concepts in the input natural language sentence. Typical performance for a parsing run on a news article is:

Total number of phrase parses attempted	210
Number of phrases for which a CycL translation was found	79
Average time to translate	5 seconds

On the second pass over the TextDocument object, Cyc’s word denotation parser processes the uninterpreted sentences, returning Cyc terms for lexically mapped words and phrases.

### Parsing into Semantic Representations

Although a great deal of progress has been made over the past decade in the development of efficient syntactic parsers for natural languages, semantic parsers, which attempt to reach a detailed understanding of the NL input, have been less well studied and less successful. This may be due in part to the lack of a suitable target representation, for which the existence of PropBank<sup>6</sup> [Gildea and Palmer 2002], and, more recently, the availability of OpenCyc and ResearchCyc<sup>7</sup> may offer some relief. The lack may also be due to the difficulty of the process, since unlike syntactic parsing, semantic interpretation depends critically on solutions to difficult linguistic problems, including anaphor resolution, disambiguation, interpretation of metaphors, preposition interpretation, and quantification. It is therefore worth spending a little time to explain the progress we have made during our research and how we have deployed it within this application.

Suppose one is faced with a sentence like “Bill Clinton bought a house in New York”. The first step in interpretation is to perform a syntactic parse targeting the TreeBank tag set. For this prototype we made use of the parser developed by Eugene Charniak at Brown University [Charniak 2000]<sup>8</sup>. This parser yields:

```
[S [NP [NNP "Bill"] [NNP "Clinton"]]
  [VP [VBD "bought"]
    [NP [NP [DT "a"] [NN "house"]]
      [PP [IN "in"]
        [NP [NNP "New"] [NNP "York"]]]]]]
```

<sup>6</sup> <http://www.cis.upenn.edu/~ace/>

<sup>7</sup> Open Cyc is a completely unrestricted subset of the Cyc KB and inference system, and includes a scaffolding taxonomy of approximately 60,000 terms that ensure interoperability with other Cyc KB versions. Research Cyc includes all of OpenCyc together with a large number of assertions and rules concerning the scaffolding terms; this high utility version of Cyc is currently in beta and will be available under a research purposes license.

<sup>8</sup> The system, however, is not dependent on the use of this parser; in a current research project our team is collaborating with Stanford University in an effort to achieve semantic parses of English and Chinese using the Stanford Parser (Klein and Manning 2003). We are also exploring the use of the CMU Link parser [Sleator and Temperley 1993].

From this parse, the system identifies the main verb, “bought” in this case, and finds its denotation in the KB (#\$Buying) and the appropriate semantic translation pattern (SemTrans):

```
(and (isa :ACTION Buying)
      (buyer :ACTION :SUBJECT)
      (objectPaidFor :ACTION :OBJECT))
```

This is used, in turn to understand the argument structure of the syntactic parse. The syntactic subject,

```
[NP [NNP "Bill"] [NNP "Clinton"]],
```

and the syntactic object,

```
[NP [NP [DT "a"] [NN "house"]]
     [PP [IN "in"]
          [NP [NNP "New"] [NNP "York"]]]]
```

are isolated for the purposes of completing the retrieved SemTrans, and interpreted using the Cycorp-developed recursive noun phrase parser, for the base NPs (“Bill Clinton”, “house”, “New York”<sup>9</sup> in this case<sup>10</sup>), combined and compositional parsing of modifiers (“in New York”, in this case), producing the CycL interpretations #\$BillClinton and

```
(and
  (isa ?HOUSE House-Modern)
  (in-Underspecified ?HOUSE NewYork-State)).
```

Substituting these into the SemTrans, and replacing the remaining role key ‘:ACTION’ with an existentially qualified variable, yields the final CycL interpretation:

```
(thereExists ?ACTION
  (thereExists ?HOUSE
    (and (isa ?ACTION Buying)
          (buyer ?ACTION BillClinton)
          (objectPaidFor ?ACTION ?HOUSE)
          (isa ?HOUSE House-Modern)
          (in-Underspecified ?HOUSE NewYork-State))))
```

The rendering of the prepositional phrase as “in-Underspecified” represents a residual ambiguity which future versions of the system will attempt to resolve using background knowledge and discourse context<sup>11</sup>. The current system typically produces translations that render much of the sense of input sentences, but that omit some of the information they contain.

### User Interaction in Annotating Partially Translated Documents.

To help ameliorate some of the imperfections in the semantic translation process, the system provides the opportunity, but not the necessity, for users to interact with the current interpretation of a document, resolving ambiguities and adding additional information. Analyzed documents can be displayed in an interface that maintains correspondences between the text of the original document and the current logical interpretation. Fully interpreted terms in the document are highlighted in green; clicking on them takes the user to an appropriate “Factive” knowledge acquisition form, allowing rapid knowledge entry in natural language. While some of the most commonly used forms have had their representation in the KB hand-crafted by knowledge engineers, the vast majority of forms are produced automatically by the system, using background knowledge and inductive inference over known cases. In experiments performed in the course

<sup>9</sup> Another possible interpretation is New York City. For this example, we assume a user has correctly disambiguated.

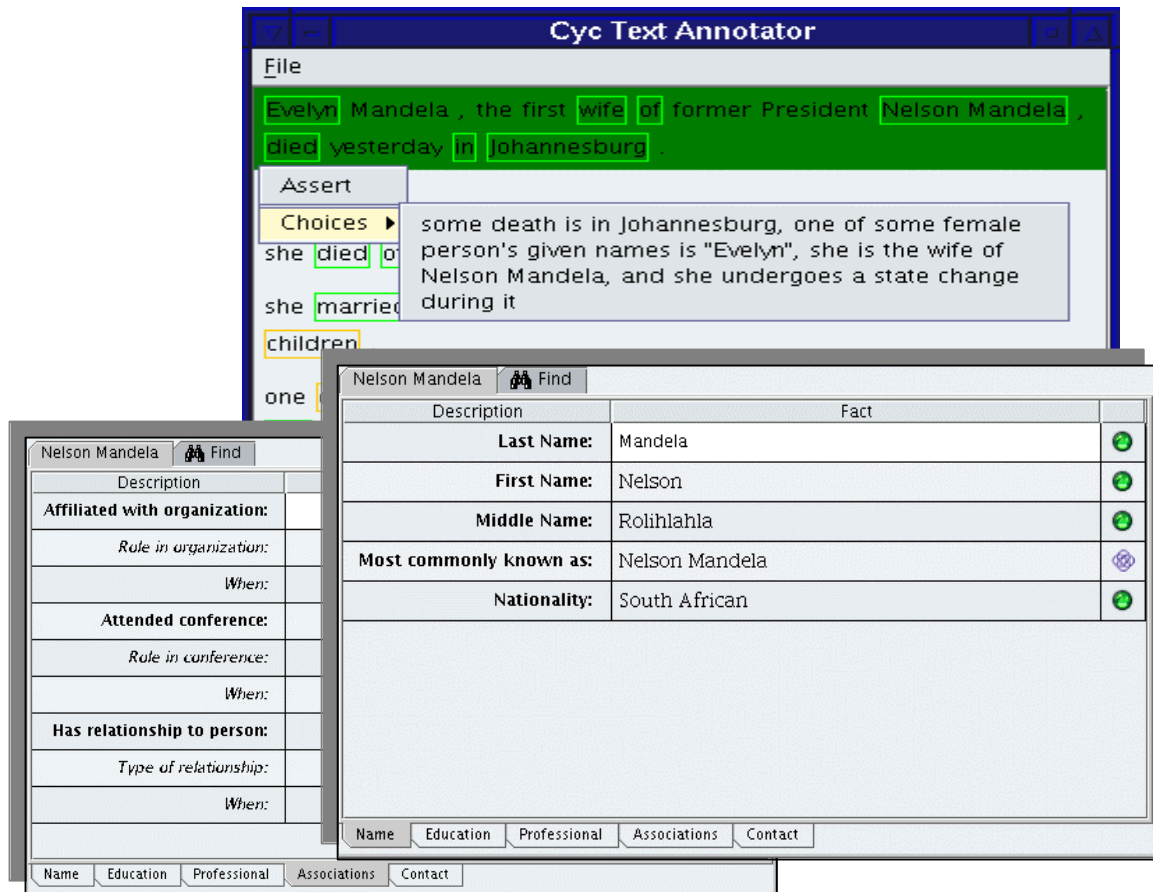
<sup>10</sup> In addition to being able to map single and multi-word tokens into CycL terms – e.g. “Bill Clinton” to #\$BillClinton – the NP parser can interpret a wide variety of compound NPs, e.g. “Bronze age farmers” are farmers that were active during the Bronze age and “black leather jackets” are jackets made of leather and black in color.

<sup>11</sup> To the predicate #\$ObjectFoundInLocation, in this case.



of entering knowledge about terrorists and their activities, lightly trained domain experts have achieved knowledge entry rates exceeding thirty facts<sup>12</sup> per hour using this interactive interface.

The other interactions available to users are selection from amongst interpretation alternatives (via menus rendered by the Natural Language Generation system) for terms highlighted in orange, and obtaining a complete English paraphrase of the current logical interpretation of a sentence, before it is asserted.



**Fig. 2:** After the system has analyzed a document, it can be made available to the user for further annotation. Terms recognized within sentences are marked in green, if fully interpreted, and orange, if ambiguous to the system. Users can chose to resolve ambiguities in pull down menus, forcing reinterpretation of the affected sentence, or can leave the ambiguity intact. The current interpretation can be disclosed to the user by automatically paraphrasing it back into English, as shown in the pop up. More information can be provided about terms in the document, at the users whim, by accessing “Factivore” knowledge entry forms, which provide a rapid, NL mechanism for assertion into the knowledge base.

### Asserting CycL Sentences into a Unique Cyc Microtheory

The fourth stage of the parsing pipeline asserts the CycL sentences and Cyc terms into a unique Cyc microtheory (context) within the knowledge base. The microtheory represents the propositional content of the target web page, and it is placed within the Cyc microtheory inheritance lattice so that commonsense assumptions about the target web page document are made explicit within Cyc. For example, a current

<sup>12</sup> A fact is a single assertion made into the Cyc KB. Facts can express simple concepts (such as “George W. Bush is a person”) or more complicated concepts (such as “something is consumed during every eating event”).

news article microtheory inherits rules and facts from Cyc's CurrentWorldDataCollectorMt. Existential variables are replaced by concrete terms during the CycL sentence assertion. Below is an assertion as parsed from the text "Bill Clinton bought a house in New York":

```
(thereExists ?ACTION
  (thereExists ?HOUSE
    (and (isa ?ACTION Buying)
      (buyer ?ACTION BillClinton)
      (objectPaidFor ?ACTION ?HOUSE)
      (isa ?HOUSE House-Modern)
      (in-Underspecified ?HOUSE NewYork-State))))
```

Replacing the existentially quantified variables with their skolem equivalents in the formula yields:

```
(and (isa Buying21 Buying)
  (buyer Buying21 BillClinton)
  (objectPaidFor Buying21 House-Modern22)
  (isa House-Modern22 House-Modern)
  (in-Underspecified House-Modern22 NewYork-State))))
```

**"Government officials believe the men were planning an attack in the lead-up to Spain 's general election."**

**PATH:** HTML[ 2 ] / BODY[ 1 ] / TABLE[ 3 ] / TR[ 1 ] / TD[ 3 ] / TABLE[ 2 ] / TR[ 2 ] / TD[ 1 ] / FONT[ 1 ] / P[ 2 ] /

```
(thereExists :INF-COMP, ?PLANNING0397, ?MEN0411, ?ATTACK0413,
  ?LEADUP0415, ?ELECTION0407, ?SPAIN0416,
  ?GOVERNMENT-OFFICIALS040
```

```
(and
  (isa ?GOVERNMENT-OFFICIALS0409 PublicOfficial)
  (beliefs ?GOVERNMENT-OFFICIALS0409
    (and
      (and
        (equals ?SPAIN0416 Spain)
        (isa ?ELECTION0407 Election)
        (to-UnderspecifiedLocation ?LEADUP0415 ?ELECTION0407)
        (in-UnderspecifiedContainer ?ATTACK0413 ?LEADUP0415)
        (isa ?ATTACK0413 AttackOnObject)
        (isa ?MEN0411 AdultMaleHuman)
        (and
          (isa ?PLANNING MakingAPlan)
          (performedBy ?PLANNING0397 ?MEN0411)
          (isa ?PLAN PlanSpecificationMicrotheory)
          (scheduledEvents ?PLAN :INF-COMP)
```

**Paraphrase:** there is some :INF-COMP such that some public official believes some other individual ?ELECTION3835 is an election, some purposeful composite physical and mental activity is an attack, someone ?MEN3839 is a man, Spain has ?ELECTION3835, in some sense, ?ELECTION3835 is the location of some other individual ?LEADUP3843, that purposeful composite physical and mental activity is in ?LEADUP3843, and some other action ?PLANNING3825 is a planning, some plan is a plan, ?MEN3839 deliberately performs ?PLANNING3825, that plan for :INF-COMP, and the plan is the result of ?PLANNING3825

**Figure 3:** The result of translating one sentence of a document into CycL. These translations are often quite complex, and, as in this case, imperfect, but provide a good basis for editing the OWL representation into an accurate reflection of document semantics. The paraphrase is the result of automatic conversion of the CycL translation back into English, and is given as an aid to reading. Paraphrase into English is not present in the Cyc Annotator output.



## Exporting CycL into OWL

The fifth and final stage of the web page annotation pipeline exports the document microtheory contents into an OWL XML document. All the built-in OWL Classes and properties have CycL equivalents. Here are sample rules for exporting some CycL predicates that happen to have built-in OWL definitions:

```
#$disjointWith --> owl:disjointWith
#$equals --> owl:sameAs
#$genlPreds --> rdfs:subPropertyOf
#$genls --> rdfs:subClassOf
#$isa --> rdf:type
#$TransitiveBinaryPredicate --> owl:TransitiveProperty
```

The sample CycL formula results in the following OWL RDF triples, with boldface to indicate the transformation of CycL predicates that are defined in Cyc's OWL ontology:

```
<Buying rdf:ID="Buying21">
  <buyer rdf:resource="#BillClinton">
    <objectPaidFor rdf:resource="#House-Modern22">
</Buying21>
<House-Modern>
  <in-Underspecified rdf:resource="#NewYork-State">
</House-Modern>
```

A portion of the OWL output for a particular news story is included in Figure 1, above. The primary difficulty in the OWL export process was the expressiveness limitation of OWL with respect to CycL. We overcame this by ensuring that the CycL assertions were ground atomic formulae, without functional terms and using only binary predicates. For cases such as rules, where the representation is not amenable to OWL export, we omit them from the OWL markup.

## Conclusions and Future Work

The Cyc OWL annotator seeks to lower the barriers to the acceptance and growth of the semantic web by using the Cyc system to produce fully automatic, partial OWL markup for unrestricted text documents. This is done by applying lexical information and background knowledge from the Cyc knowledge base, subsystems for text analysis, optional interactive knowledge acquisition and disambiguation, isolation of incomplete knowledge within a microtheory structure, and down-projection of CycL logical representations into OWL.

One of the central thrusts of our research is improving the process of translation from unrestricted natural language text into full logical representations; over the next year we expect substantial improvements in the quality of English interpretation, and initial results for Chinese interpretation; these improvements should directly improve the resulting OWL annotations.

An independent research direction involves adding the ability for the system to optionally produce OWL extended with RuleML and other proposed extensions to the language of the semantic web, improving the quality of the output produced by down-projection from CycL. These extensions should be straightforward to produce once the relevant standards are adopted.

This work was supported by DARPA's DAML program, and used additional technology supported by ARDA's AQUAINT program and a Phase I SBIR grant.

## References

- Burns, Kathy J. and Anthony R. Davis. 1999. "Building and Maintaining a Semantically Adequate Lexicon Using CYC" in Evelyn Viegas, *Breadth and Depth of Semantic Lexicons*. Kluwer: Dordrecht.
- Charniak, Eugene. 2000. "A Maximum-Entropy-Inspired Parser". *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics* (NAACL'2000), Seattle, Washington.
- Gildea, Daniel and Martha Palmer. 2002. "The Necessity of Parsing for Predicate Argument Recognition" In *Proceedings of ACL 2002*, Philadelphia, PA.
- Klein, Dan and Christopher D Manning. 2003. "A\* Parsing: Fast Exact Viterbi Parse Selection." HLT-NAACL 2003, Edmonton, Canada.
- Sleator, Daniel and Davy Temperley. 1993. "Parsing English with a Link Grammar". Third International Workshop on Parsing Technologies, Tilburg, The Netherlands and Durbuy, Belgium.





# Multimedia Distributed Knowledge Management in MIAKT

David Dupplaw, Srinandan Dasmahapatra, Bo Hu, Paul Lewis, Nigel Shadbolt

IAM Group, University of Southampton, Southampton, SO17 1BJ, UK  
[dpd|sd|bh|phl|nrs]@ecs.soton.ac.uk,  
WWW home page: <http://www.aktors.org/miakt>

**Abstract.** Digital media facilitates tight integration of multi-modal information and networking allows this richly textured knowledge to be shared. We present the system we have developed in the MIAKT (Medical Imaging with Advanced Knowledge Technologies) project that provides knowledge management, and facilities for semantic annotations on mammographic images in the context of clinical and histopathological information.

This paper also describes the novel generic architecture we have built on semantic web technologies to facilitate the annotation of images with ontological concepts, and storage thereof, in any domain. Functionality of a specific domain application is provided through web-resources, which are called through a task invocation system which abstracts the actual service implementation from the client application implementation.

## 1 Introduction

The drive towards semantic web [4] technologies has provided a research area that brings together semantic annotation and image feature extraction. Automating semantic annotation of images is a difficult process in most domains, the annotation requiring some level of intervention from users. In the medical domain, images are rarely clearly defined, and often regions of interest are difficult to spot by a trained expert. By combining a number of different technologies, including the semantic web technologies, into a generic system, we can begin to provide some support for both the manual and automatic annotation of these images, as well as providing a means for retrieval and reuse of the data.

Breast cancer screening is now mandatory for women over the age of 50. This process consists of the capturing of an x-ray mammogram and a radiologist examining it for any areas considered abnormal. They are then assessed, if necessary, by means of pathology tests (biopsies) by a histopathologist. Data from the radiologist, the histopathologist, and the clinician (who has knowledge of the history of the patient) are brought together to make a consultative appraisal of each particular case in a Multi-Disciplinary Meeting (MDM). This process is known as the Triple Assessment Procedure and the work presented here, as part of the MIAKT (Medical Imaging with Advanced Knowledge Technologies) project, is intended to support this collaborative meeting and manage the knowledge that goes with it, using the Semantic Web technologies.

To achieve this we have developed a novel architecture for delivery of applications to users based on ontological application descriptions. The application's data sources and computation sources are distributed which provides access to the application from any available application server via a roving client application. An important and convincing argument for the use of such a distributed framework is that all parties involved in an application's data or functionality pool retain control of their respective property whilst still being able to access the relevant parts from remote application clients. It eases both the integration issues as well as the intellectual and ethical issues for institutions to retain rights to their data, property or system, and provide a service to which interested parties can connect.

Image annotation is conducted locally on a user's machine to ensure adequate user feedback. However, the images are retrieved from remote servers, image features are generated using remote analysis services, and the results of the annotations are stored in the ontological database that contains the patient record of the patient concerned.

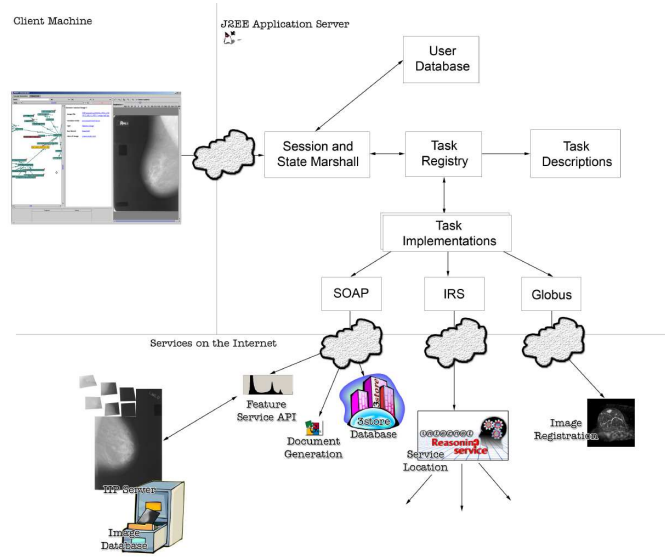
In the following section we briefly review related work. Section 2 presents an overview of the generic knowledge management framework used to provide the middleware to the multimedia and knowledge management process, and in sections 3 and 4 we describe how we use this in the MIAKT application. In the MIAKT scenario we describe the image distribution system and the image annotation tools. Section 5 gives conclusions and a brief mention of future work.

## 2 The Generic Framework

The novel, distributed architecture, that the MIAKT application is built upon, uses web-based services to provide discrete and disparate functionality to a generic application base shown in figure 1. The architecture is deliberately abstracted from any particular application domain (and its description) providing a generic structure for rapidly prototyping new knowledge-based applications that require media annotation in new domains. Abstracting the architecture from the application domain provides a considerable challenge in the designing of an API that ensures components are still interoperable in disparate domains.

A user transparently interacts with the architecture through an application client that is also built around a generic architecture that can be rapidly implemented into a specific application by mediation through an 'application ontology'. This application ontology is distinct from the domain ontology and provides application settings for a specified domain such as which media viewers are used to display and annotate images.

The core of the framework is based on the invocation of web-services through a task invocation sub-system that provides configurable functionality for the target application. The services available to a specific application are described in that application's ontology. The methods made available by these services are automatically discovered through description mining, or by server interrogation depending on the implementation of the target service and the nature of the 'han-



**Fig. 1.** The MIAKT framework

dlers' or description provided. The methods are associated through a mapped repository with task names which are called from a client unconcerned with the task's implementation. Currently the architecture supports both SOAP [11] (web-services) and the Internet Reasoning Service (IRS) [8] task implementations and they are imported into a task registry using WSDL [10] mining for the SOAP tasks, and server interrogation for the IRS. The architecture makes it simple to add new service providers, and it is possible that invocations of services on Globus servers will be supported in the future.

It is important over such a communal service architecture to have interoperable function calls, and all data given to, and returned by, services are to be of primitive types : strings, integers, etc. Complex data is marshalled to and from XML by domain-specific handlers.

To store the domain data in instantiated ontologies, our database service is based on an RDF-triple database called 3Store [3]. This database is accessed over a SOAP webservice, and to maximise interoperability, results are returned as XML and parsed in the client to extract pertinent information or display results.

The on-demand delivery of the application description to the generic client provides a means to customise the application to a given domain, while the distributed nature of the framework provides potentially unlimited interoperability, giving access to any web-service based from any application domain. In the next section we describe how this generic base is put to use for medical image and knowledge management in the project in which it was developed.

### 3 The MIAKT Application Architecture

For the medical knowledge management domain, a number of data sources which are used to provide the underlying data for this domain are required. For the support of the multi-disciplinary meeting we require at least the following:

- Patient records including information about the patient, what examinations they have undergone, and what results were concluded from those examinations.
- Multimedia data such as X-Rays and MRI mammograms that are taken during examinations and are required for marking up suspicious areas and then relating those to the patient’s medical data, including biopsies.

Using the relevant media viewers and analysis services, the client application can automatically provide a method for associating annotations made on the multimedia data to semantic concepts in the domain data, which is the patient information in this application. The following sections describe these data sources and their usage.

#### 3.1 Patient Records

The Breast Cancer Imaging Ontology (BCIO), developed in the MIAKT project, is designed in a modular manner representing different levels of resolution of the application domain. Highly abstract terms, such as “Medical Image” or “Image Descriptor”, are on one end of the descriptive grain-size while concrete descriptors, like “Spiculated Margin” describing the shape of a region of interest, are on the opposite end. Between them are several levels of interim concepts constructing a referencing bridge. Such an approach makes it possible to replace a particular part of the ontology to adapt to minor, or fundamental, changes of the application domain.

The BCIO ontology is based on a standardised lexicon called BI-RADS (Breast Imaging Reporting and Data System) developed by the American College of Radiographers (ACR) [7]. We have utilised recommended guidelines by the ACR and the National Health Service in the UK to extend this lexicon and develop the ontologies. The ontologies are compliant[2] with the Web Ontology Language (OWL)[1] standard.

We currently source our data from anonymised, legacy data [12], but in practise this would be entered by radiographers as new patients arrive. The patient records would, in practise, be stored on 3store databases located at the institution controlling the data, and accessed by webservices with appropriate safeguards to ensure privacy and compliance with ethical procedures.

#### 3.2 Images and Multimedia

The way in which images and other multimedia data are integrated into the framework can have an important effect on the flexibility of the image annotation

systems, and the system as a whole. Therefore, the framework does not stipulate any particular conceptual position in the architecture for storage, or analysis of multimedia data. It is possible to have the data stored separately from both the application client, where the user is viewing it, and the analysis algorithms which are calculating and storing feature vectors for features in the data. Indeed, multimedia data, like the various modalities of images that are produced in the medical domain, can be stored on institutionalised servers which are able to deliver the data to the client on demand by image servers. This provides the potential to integrate with current hospital image repositories such as PACS (Picture Archiving and Communications System).

Details of the multimedia data are entered as instance information along with the patient record, thereby linking the remote data sources with the patient examination record which facilitates immediate access to the relevant multimedia data at the client.

The digital x-ray mammogram images, fundamental to this domain, have very large dimensions (on average about 2500x4000 pixels, or about 4Mb when compressed). The images are transferred using the Internet Imaging Protocol (IIP) over a standard servlet interface, which gives the ability to view and manipulate them over relatively low bandwidth connections despite this size. The IIP servlets deliver image tiles, on-demand, from various precalculated resolutions of the image, to the client's IIP image viewer. MRI images are delivered slice-by-slice from an MRI image server to the client's MRI image viewer.

The framework is not confined to using any particular protocol for serving images, and indeed it would be undesirable to limit flexibility in this way. Access to image servers is initiated from only those processes that understand the respective image modality, such as an image viewer in the application client or a feature vector generator on a feature service. This relieves the application server from the transfer of any potentially large images, because image transfer is conducted directly between the image client and the image server.

```
<source-information>
  <source type="IIP Image">
    <image-server>http://imgsrvr/cgi-bin/iipsrc.cgi</image-server>
    <image-filename>/images/case0042/LEFT_ML0.LJPEG.tif</image-filename>
  </source>
  <source-region type="PointSet2D">
    <boundary>(100:100),(200:200),(200:100)</boundary>
  </source-region>
</source-information>
```

**Fig. 2.** Complex object marshaling using primitive strings containing XML, showing the marshaled result for a simplified example of a region of interest on an image.



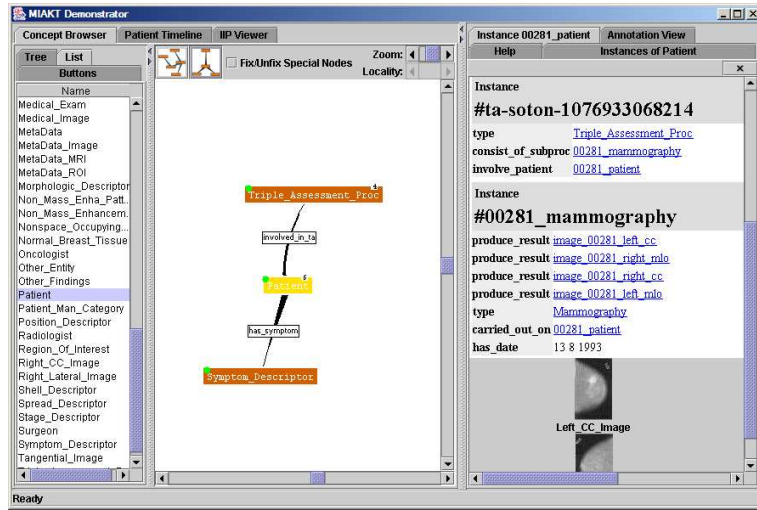
To allow the comparison and classification of images based on their content, image descriptors are generated. Currently some generic shape and colour descriptors are integrated into the system, and our collaborators at Oxford are developing X-ray-specific feature modules that deliver descriptors more relevant to the domain, providing a better means for classification. We have developed an API that provides general functionality for generating and comparing feature vectors on images and publishing these analysis algorithms as a webservice. It provides a defined interface for client processes to interact with any feature modules that are offered as a service on the server. Feature vectors are created from annotations made in the relevant viewers. When an annotation is created the feature module is automatically called to create the relevant vectors. When calling the feature service, the inputs and outputs, such as the definition of a source image or image region, are marshaled into an XML object such as the simplified example shown in Figure 2. The flexibility of this feature service architecture means it would be a simple integration process to replace the default relational database with specialised feature-based indexing databases.

Using these frameworks, the MIAKT application supports specific medical image analysis algorithms, one of which is the registration (alignment) of images [6] from different time frames, or even different modalities such as histopathology slides or MRI slices, which provides a good method for abnormality detection in MRI images (using subtraction of registered images). These types of registration process are highly computationally intensive and have been implemented using Grid technologies, which are currently accessed using a standard web-service invocation mechanism. In the MIAKT application, there are also services for the generation of descriptors for masses in both MRI images and X-ray images. The descriptors these services calculate are based on the domain-ontology. In the ontology the shape of a mass can be described as ‘irregular’, ‘round’, etc., and data including this information is generated from an X-ray image analysis service. Subsequently, classification services using Bayesian networks, attempt to classify an abnormality as malignant or benign based on its descriptors. A similar service is available for MRI images that returns a final finding ‘malignant’ or ‘benign’ based on a set of low-level, image-content features taken from an annotation roughly delineated by a user.

## 4 The Client Application

A generic application client provides the knowledge management user interface, including the image annotation tools. Images are stored as instances in the domain ontology, and the application client gives the user access to these instances via an interactively navigable ontology visualisation tool, as described below. The domain ontology is retrieved from a location on the network stipulated by an application ontology instance.

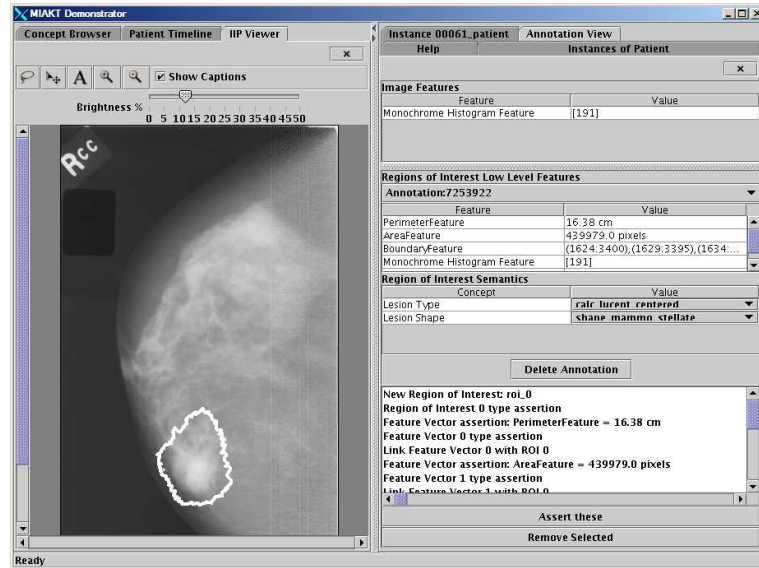
We provide two major ways to navigate the ontology. Firstly there is the hierarchical view, which shows the concepts in the ontology based on the typical subsumption relationship allowing quick navigation to concepts. A Touch-



**Fig. 3.** The generic client application has hierarchical, list and TouchGraph [9] views of the ontology. Here a patient instance is being viewed using an application dependant handler for the mammography and the generic handler for other instances.

Graph [9] view of the ontology allows the full network structure to be viewed and manipulated in real-time (Figure 3). Instances of particular concepts can also be retrieved from this view and information about a specific instance can be recalled by clicking on the relevant instance identifier. Instances are displayed as a list of slots and values. To allow customisation of the client application, the instance handlers that display these lists can be extended for particular concept types, thereby allowing the client application to provide context-based, and domain-based instance visualisation. Media viewers are then dynamically loaded into the application client and instantiated based on the instance type, media type, and/or the delivery mechanism of the media.

The media viewers are implemented using a defined interface which allows them to be invoked to produce annotations of regions of interest in a given medium. For example, the IIP image viewer allows users (in the case of MIAKT, radiologists) to draw around regions of interest in the image, as shown in Figure 4, and these form the basis of annotations. An annotation observer process receives annotations from the media viewers and automatically invokes feature modules, both local and remote, that are able to take the given region of interest as input and produce feature vectors. Domain-dependent feature analysis modules may output concepts relevant to the domain-ontology, thereby allowing direct (but manually verified) insertion into the instances of concepts from the domain. For example, the margin of a mass may be classified using shape features (irregular, round, etc.). Non-domain feature vectors may be inserted into



**Fig. 4.** The IIP image viewer allows image annotations to be generated using drawing tools. When feature vectors are available, they are displayed alongside the image.

the domain instances where appropriate, or under a generic ‘Image Descriptor’ banner.

The architecture provides automatic activation of modules that perform media processing using a regimented API between the application and the observers that receive annotations from viewers. The framework’s indifference to local and remote activation of media modules facilitates sites with large computational power, or storage capacity, to be used to generate descriptive vectors from media which is remote to both the feature module and the client. For example, the MIAKT project uses image analysis modules running remotely in our partner sites.

Once features have been generated by feature modules, they are automatically mapped to domain concepts to be associated with the ontology as instances in the database. This is currently achieved by feature and domain-specific classification code, although we are investigating using generic classification techniques for this step, that classify feature vectors into a set of controlled classes specified by the ontology. This classification provides default values for the semantic descriptions of the relevant annotation, which the user can validate. The insertion of instances into the database is done manually by the user, thereby allowing the user to disregard features which are giving no added information, or incorrect information. To aid the speed of this process, and to allow the user to make alterations to the instances, assertions into the database are pooled prior to a batch assertion. We are investigating ways to make this insertion an easier process; us-

ing form-based input is a well-understood method of knowledge storage for the medics we have contacted, but providing a method for the generation of general forms provides some challenge. It is possible that domain-based context-based form generators would be necessary, but at the expense of generic flexibility.

#### 4.1 Other Services

To enrich the value of the MIAKT application, other services have been included which are available through the server-side task invocation sub-system.

Consultation during a multi-disciplinary meeting, on the best course of action for the patient, relies on the outcomes of the examinations that the various medical staff performed on the patient. These outcomes are based mainly on the doctor's experience, but also rely on their full attention and concentration. It is possible for human errors to be made. For this reason we have developed naive Bayes and MLP-based classification algorithms that, based on patient records for previous patient cases, attempt to classify the type of lesion from its ontological description. In the near future, we hope to extend this classification to image-content-based features. On our current data sets they are giving correct results around 75% of the time, although currently, this accuracy does seem to be limited by our datasets.

Using technologies such as GATE [5], collaborators at Sheffield have developed a technology which allows natural language documents to be generated based on the ontological instance data. By applying this technology to a patient's case notes, the effort of writing up routine patient reports is reduced for the busy medical staff who currently have to do this by hand.

The UMLS (Unified Medical Language System) is a repository of thousands of medical terms along with their description, mediated through a meta-thesaurus. We have made this service available through a web-service to the client application to allow descriptions to be sought for relevant medical terms.

## 5 Conclusions and Future Work

This paper has described how the web-based application architecture that has been developed in the MIAKT project can be used to provide knowledge management for applications where semantic image annotation is necessary, and in particular, how it has been used to provide multi-media knowledge management in the medical domain. As well as image annotation, the system provides for multi-platform service invocation based on the instances of an application ontology, which we believe is a generic and flexible protocol for multi-application deployment.

In future developments the application ontology will be formalised into an abstract process model based description of the application which will provide a mechanism for generating unique application clients that are suited to users of different applications. Also the MIAKT application will be built-upon to provide greater support to the application of the multi-disciplinary meetings with

scheduling of hospital resources, further image analysis modules and classification services.

Our generic architecture lends itself to many different domains and we are looking forward to using the system to prototype different applications in different domains to prove its genericity.

## 6 Acknowledgements

The authors are grateful to their collaborators on the project - Kalina Bontcheva, Michael Brady, Liliana Cabral, Fabio Ciravegna, John Domingue, David Hawkes, Hugh Lewis, Enrico Motta, Maud Poissonier, Christine Tanner, Yorick Wilks and Yalin Zheng - for many valuable discussions. This project (MIAKT) is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number GR/R85150/01. MIAKT is part of the e-Science initiative and is a collaboration between two IRCs supported by the EPSRC . AKT (GR/N15764/01) and MIAS (GR/N14248/01).

## References

1. McGuinness, D., van Harmelen, F.: *Ontology Web Language (OWL) Overview*. Available at <http://www.w3.org/TR/owl-features/> (August 2003).
2. McGuinness, D., Hendler, J., Stein, L.A.: *DAML+OIL: An Ontology Language for the Semantic Web*. *IEEE Intelligent Systems*, (2002) 72–80.
3. Harris, S., Gibbins, N.: *3store: Efficient Bulk RDF Storage*. In *Proceedings 1st International Workshop on Practical and Scalable Semantic Web Systems*, Sanibel Island, Florida, USA, (2003) 1–15.
4. Berners-Lee, T., Hendler, J., Lassila, O.: *The Semantic Web*. *Scientific American*, May 2001.
5. Bontcheva, K.: *Reuse and Problems in the Evaluation of NLG systems*. In *Proceedings of EACL.03 Workshop on Evaluation Initiatives*, April 2003.
6. Zheng, Y., Tanner, C., Hill, D.L.G., Hawkes, D., White, M., Khazen, M., Leach, M.: *Alignment of Dynamic Contrast Enhanced MR Volumes of the Breast for a Multicenter Trial: An Exemplar GRID Application*. In *Medical Imaging 2004*, San Diego, CA, USA. SPIE, February 2004 (to appear).
7. American College of Radiology. *Breast Imaging Reporting and Data System: BI-RADS*. Available at [http://www.acr.org/departments/stand\\_accred/birads/](http://www.acr.org/departments/stand_accred/birads/).
8. Motta, E., Domingue, J., Cabral, L., Gaspari, M.: *IRS-II: A Framework and Infrastructure for Semantic Web Services*. *2nd International Semantic Web Conference (ISWC2003)*, Florida, USA, (20th–23rd October 2003) 306–318.
9. TouchGraph LLC. *TouchGraph*. Available at <http://www.touchgraph.com/>.
10. *The WSDL Specification*, WWW Consortium, March 2001. Available at <http://www.w3.org/TR/wsdl>.
11. *The SOAP Specifications*, WWW Consortium. Available at <http://www.w3.org/TR/soap/>.
12. Heath, M., Bowyer, K.W, Kopans, D. et al.: *Current Status of the Digital Database for Screening Mammography*, in *Digital Mammography*, Kluwer Academic Publishers (1998), 457–460.

# Low Cost Mark-Up for Lightweight Semantics

Simon Harper and Sean Bechhofer  
sharper, seanb [ @cs.man.ac.uk ]

Information Management Group, Dept of Computer Science,  
University of Manchester, Manchester, UK.  
<http://augmented.man.ac.uk>

**Abstract.** Visually impaired users are hindered in their efforts to access the largest repository of electronic information in the world, namely the World Wide Web (Web). A visually impaired user's information and presentation requirements are different from a sighted user in that they are highly egocentric and non-visual. These requirements can become problems in that the web is visually-centric with regard to presentation and information order / layout, this can (and does) hinder users who need presentation-agnostic access to information. Our objective is to address these problems by creating usable appropriately 'displayed' web pages for use by all users who wish to understand the meaning as opposed to the presentation and order of the information. We assert that the only way to accomplish this is to encode the pages semantic information directly into the page. And the only way this will occur in the real world is if authors have no 'semantic overhead' when creating these pages. In this paper we describe preliminary work towards a system to enable just this kind of semantic encoding so that, in effect, authors get low cost semantics.

## 1 Introduction

We assert that the most preferential way to enhance visually impaired peoples access to information on web-pages is to encode the meaning of that information into the specific web-page it refers to. However, there are problems. Empirical evidence suggests that authors and designers will not separately create semantic mark up to sit with standard XHTML<sup>1</sup> because they see it as an unnecessary overhead.

Recently, we have seen a movement towards a separation of presentation, metadata (XHTML), and information. However, this has not been enough to support the unfettered access of visually impaired users. Consider, the excellent 'CSSZenGarden' (see Fig. 1). The site is a model of the state-of-the-art: the application of standards, separation of presentation and content, and visually stunning too. But, it is still reasonably inaccessible to visually impaired people. Inspect the site without an applied stylesheet (see Fig. 2). Visually impaired users interact with these systems in a 'serial' (audio) manner as opposed to a 'parallel' (visual) manner. Content is read from top left to bottom right, there is no scanning and progress through information is slow. Given this interaction paradigm we can see that visually impaired users are still at a disadvantage because they have no idea items are menus, what the page layout is, what the extent is. In effect, the implicit meaning contained in the visual presentation (see Fig. 1). is lost and any possibility of enhanced meaning is also not available as only authoring concepts (like footnote, heading, leftcolumn) are listed (see Fig. 2).

While authors and content creation engines still create non-standard CSS<sup>2</sup>-XHTML

<sup>1</sup> Extensible Hypertext Markup Language

<sup>2</sup> Cascading Style Sheet



**Fig. 1.** Zen Garden with CSS 83

identifiers, they also often compound the problem by using linear paper based (book) metaphors such as: footer, header, bold, big, etc. This information can in fact be inferred from the coded style and presentation information contained within the CSS. This means the combination of identifier and presentation information together often represent a tautology.

Even when authoring concepts do look as though they have a meaning with regard to the information they are often mixed with un-descriptive qualifiers; and the problem is again compounded by the lack of an ontology in the event of there actually being some useful information to reason over. Therefore, the question which we faced and which this paper is dedicated to answering was this:

How can semantic information be built into general purpose web-pages such that the information is as accessible to visually impaired users as it is to sighted users, without compromising the page's design vision?

We based our question on a set of beliefs thus:

1. Visually impaired surfers need access to the meaning of information to assist in their cognition, perception, movement around that information, and to assist in the formulation of their world-view [4, 9]. This is the same for sighted users however pages are normally created with sighted users in mind.
2. Based on empirical and anecdotal evidence, authors and designers will not suffer a 'Semantic Overhead' when building pages.
3. A web page should be thought of an application, comprising functional elements and presentation / information elements, within an application (the browser).
4. Information should not need to be recreated (i.e. exist as XHTML for humans and RDF<sup>3</sup> for agents) when the intended audience is human. The meaning should be seamless and be part of the data.

<sup>3</sup> Resource Description Framework (Schema)

5. If we don't need to create explicit resources (RDF feeds etc) why should we?
6. Authoring concepts used as presentation identifiers are redundant when used with CSS as their presentational meaning is implicit in their technical definition.

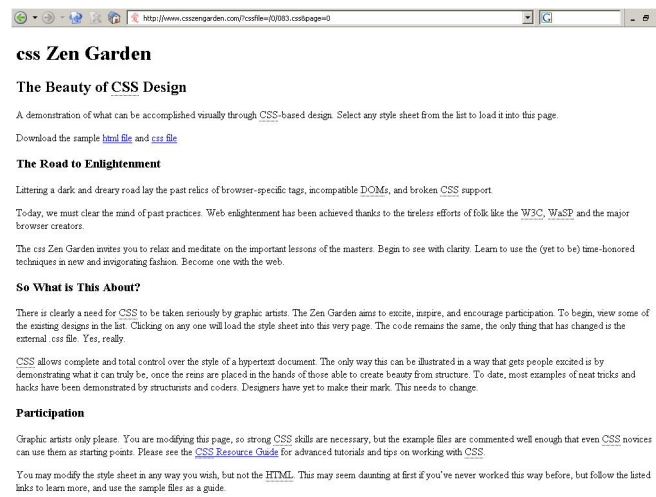


Fig. 2. Zen Garden without a CSS

This goal and set of beliefs led us to a simple, lightweight, and powerful solution. Create a grammar to represent the meaning of data within XHTML meta tags and encoded it into the data by leveraging the 'class' and 'id' attributes common to most XHTML elements. CSS presentation will be unaffected but semantics will be an implicit part of the data as opposed to an explicit duplicate representation (in say RDF(s) or N<sup>3</sup> Notation). To achieve this we combine both XHTML elements that have meaning or that can be used to accurately infer meaning; and a bespoke grammar developed to enhance the limited XHTML syntax.

The focus of our system is to represent instances as information enclosed within meta elements along with concept and property identifiers as part of XHTML meta elements themselves. These elements can then be related to OWL Lite [8] ontologies defined in the normal way.

## 1.1 Synopsis

One of the goals of the Semantic Web vision is to make knowledge accessible to agents but with a strong human input and benefit. In this framework, our goal is to make the role of the objects, that support visual accessibility through presentation, explicitly interpretable by humans (via web browsers) rather than just being visually interpretable. Therefore, it is necessary to associate metadata and semantics with XHTML objects (machine-readable vs machine-understandable). The rest of the paper can be summarised as follows:



**Background** We give an overview of how visually impaired people currently interact with web pages. We describe the problems associated with these methods and give an overview of current access paradigms and authoring concepts.

**Related Work** We present a small section on related work to place our contribution in context.

**Low Cost Semantics** We describe the concepts, rationale, and techniques behind our system focusing on the XHTML `abbr` / `acronym` elements and the `class` attribute. We show how these are referenced on XHTML pages and how our lightweight system can contribute to the accessibility of information via lightweight semantics.

**Example** As a preliminary case study we consider the simple ontology taken from ‘A Semantic Web Primer’ in an attempt to show how an ontology is represented using our methodology.

**Why Does This Approach Aid Visually Impaired Users?** We have identified the problem and suggested a solution but why do we think this is a useful solution?

**Conclusion** Finally, we focus on our conclusions from the work undertaken and look at future work including system evaluations.

## 2 Background

Access to, and movement around, complex hypermedia environments, of which the web is the most obvious example, has long been considered an important and major issue in the Web design and usability field [5, 10]. The commonly used slang phrase ‘surfing the web’ implies rapid and free access, pointing to its importance among designers and users alike. It has also been long established [4, 6] that this potentially complex and difficult access is further complicated, and becomes neither rapid or free, if the user is visually impaired<sup>4</sup>.

### 2.1 Current Access Paradigms

Visually impaired people usually access Web pages either by using screen readers or specialist browsers. If the Web pages are properly designed and laid out in a linear fashion, these assistive technologies can work satisfactorily. Some screen readers access the HTML / XHTML source code rather than solely reading the *screen*, which enables them to provide better support. However, not many pages are properly designed; the focus is usually on the visual presentation which makes audio interaction almost impossible. Furthermore, chunking the page into several parts and presenting it in a nonlinear fashion is becoming popular which makes the provided functionalities of these assistive technologies insufficient. There are guidelines to aid the designers in creating accessible pages [1], unfortunately few designers follow these guidelines and therefore Web accessibility is still a problem.

Further problems also exist when trying to gain an overview of the page. Some screen readers, for instance Jaws [11], provide overview information when the user first accesses a page. This information often includes, for example, the number of headings in the page based on the “heading” tags in the source code. However, if the page is not appropriately designed, such information could be misleading.

<sup>4</sup> Here used as a general term encompassing the WHO definition of both profoundly blind and partially sighted individuals [13].

## 2.2 The Problem with Authoring Concepts

Even when XHTML meta elements are used correctly and pages are created to standards and specifications, poor accessibility still persists. We believe this is because there are common misconceptions about what information is actually required by users. In our opinion this continued inaccessibility stems from the incorrect use of authoring concepts within the web-page.

Authoring concepts often hold information about the layout vocabularies used in transcoding and content management systems; but from a visual perspective. In this case, they do not consider the meaning of the objects in the page framework but are more interested in how the objects are presented in the Web landscape. The Web landscape is defined as the combination of the page and the agent (e.g, browser and assistive technologies such as screen readers). These concepts are more to do with the specific structures that can be used to define the overall layout of a page including for example, sections, summaries, abstracts, footers, etc. These constructs are usually implicit in the visual presentation of the page, and so many authors and transcoding systems seek to explicitly encode them in the underlying source code (e.g., HTML). However, this kind of terminology is less useful and therefore inaccessible in any other form of interaction (e.g., audio interaction through screen readers). Transcoders aim to define a vocabulary that is already widely used between the designers but not formally explained and defined, that is to say they try to make the domain knowledge explicit. However, they use the wrong paradigm, that of the linear and visual layout as opposed to the really useful information – the meaning of the actual instance of data itself.

Authors and systems need to move away from this paradigm of providing what they **THINK** users need and focus on what the creator actually **MEANS**. In this way visually impaired users can decide for themselves what is useful, and what is not.

## 3 Related Work

Adding semantics to an XHTML document is not a new concept. It has been thought about since the late 1990's however concrete solutions were proposed as early as 2002. Tim Berners-Lee proposed embedding XML RDF in HTML documents as part of the tag project [3], however these documents would not validate as XHTML and so did not find favour among the community [12]. A version was created that did validate by the inclusion of a small DTD using XHTML Modularisation. However, this was not deemed a good solution as unique extensions have to be created on a whim. In fact the work concluded that the RDF specification specifies how to understand the semantics (in terms of RDF triples) in an RDF document that contains only RDF, but does not explain how and when one can extract semantics from documents in other namespaces which contain embedded RDF. It goes on to say that the XHTML specification explains how to process XHTML namespace content, but gives no indication about how to process embedded RDF information [3]. Other methods have been proposed in which the object or script elements are used, however, the code becomes unreadable and therefore less workable although the RDF can be linked to in an external file [14]. The use of the XHTML `link` element has also been proposed, however the main problem with this method is that the RDF is not actually then embedded in the HTML source but in a separate file [14]. This file is then at the mercy of changes and synchronisation issues with the original and the amount of work needed to create the resource is the same as creating two separate and disjoint files – time and effort are not saved. Dan Connolly proposed a system called HyperRDF in which HTML is used as the conduit

to use XSLT to transform information into RDF. However, HyperRDF cannot be validated since the head element does not allow an ID attribute [7]. Augmented Metadata for XHTML is an implementation that allows Dublin Core metadata to be incorporated in Web pages in a way that is compatible with today's Web browsers. The basic premise is that one can take the profile attribute to be a global namespace prefix for all of the `rel / meta` and `name` attributes throughout the document. This approach is mainly for those authors that want to use a simple mechanism for producing RDF from their XHTML. It is ineffective from the point of view of anyone that wants to randomly extract RDF from XHTML, since one cannot tell whether the author wanted the assertions to be converted into the triples produced by the algorithm or not [2]. Finally, the most recent thinking on the subject comes in the form of GRDDL (Gleaning Resource Descriptions from Dialects of Languages). This work is being undertaken by the W3C Web Co-ordination Group and is a mechanism for encoding RDF statements in XHTML and XML. GRDDL shares some commonalities with HyperRDF and works on the principle that the HTML specification provides a mechanism for authors to use particular metadata vocabularies and thereby indicate the author's intent to use those terms in accordance with the conventions of the community that originated the terms. Authors may wish to define additional link types not described in this specification. If they do so, they should use a `profile` to cite the conventions used to define the link types. GRDDL is one of these profiles which uses XSLT to transform a page to an RDF description.

### 3.1 Why GRDDL Doesn't Work For Us

Our research centres around both the designer and the user. We wish to support the designer because in doing this we make sure our target user group are supported by the designers' creation. In our conversations with designers the resounding message we receive is

“If there is any kind of overhead above the normal concept creation then we are less likely to implement it. If our design is compromised in any way we will not implement. We create beautiful and effective sites, we're not information architects.”

Many web designers move from print media to web design and this pre-gained experience in creating static designed artifacts forces them to see design as fixed and immovable once created. A designer creates and controls the development of what is in effect a piece of art and therefore once created should not be changed or violated. It can be difficult to convey that users often require web pages to adapt to their needs, and the fact that this sometimes goes beyond art.

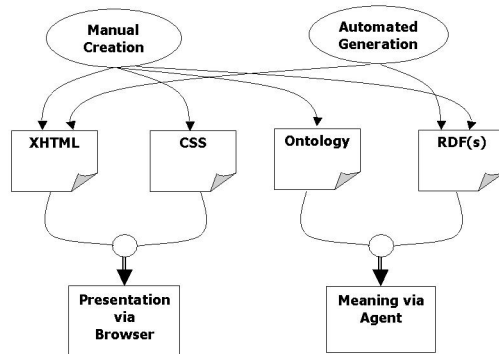
We suggest that designers need a lightweight no-frills approach to include semantic information within XHTML documents; in effect the presence of the semantic information should be seamless indivisible and have a low cost design overhead.

## 4 Low Cost Semantics

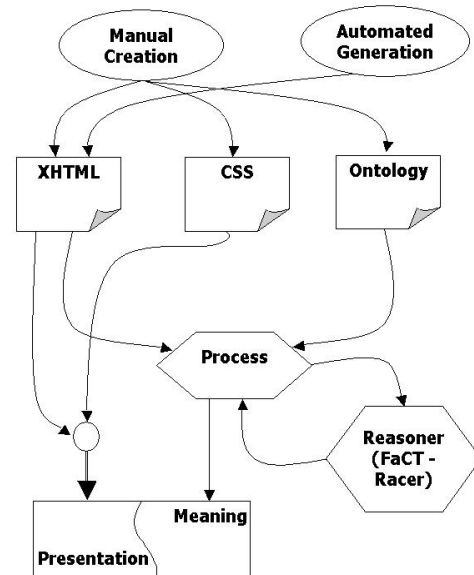
Our system is in reality a process for associating ontology concepts with instances encoded within XHTML pages. Currently, presentation and meaning are separated as we can see in figure 3. The CSS and ontologies are mostly manual created while the instances, the XHTML, and the semantics associated with instances are created either

manually or are automatically generated. The CSS and XHTML are assembled on the client and joined by the browser functionality while the RDF and ontology are used by either automated agents or RDF ‘feed’ readers (for use by humans). We suggest that this type of separation is both unhelpful, damaging, and counter to the Semantic Web vision. With Tim Berners-Lee’s desire to describe resources (many on the web as standard XHTML documents) more fully the division between the web and the semantic web will increasingly become a hindrance. Although users can currently interact with web resources, and agents are starting to interact with semantic resources, surely progress should be made towards a joining of the two. We believe there should be just one web where semantics, presentation, and information are conjoined giving a holistic world-view.

Our system is a first step towards this. We suggest that meaning should be encoded within the elements of the XHTML and CSS along with ontologies which can be created as normal. Ontological concepts and properties are encoded into both the elements and attributes of the XHTML document and are used as identifiers within the CSS which link presentation to XHTML elements. Our system revolves around a software process (see Fig. 4) which converts an RDF–XHTML document into a series of instances and ontological descriptors for supply to the reasoner. Users view the document in a web browser as normal, however, browsers that are ‘semantic-aware’ can use the ontological information to provide more intelligent access to the instances of information than before. Currently, no browsers are ‘semantic-aware’ of our system except those with a system plug-in. However, all is not lost as RDF(s) can be generated by our process and inserted into the document such that RDF(s) aware browsers can take advantage of our system (as a ‘Kludge’).



**Fig. 3.** As Things Currently Stand



**Fig. 4.** Our Preliminary System

#### 4.1 Encoding Ontologies in XHTML

Because we are suggesting a lightweight system our paradigm for encoding OWL Lite ontologies is simple, flexible, and without a semantic overhead. We use a trinity of techniques to encode semantics directly into a page:

**Class and ID Attributes** XHTML `class` or `id` attributes are used to encode a piece of semantic information in the form of a concept-class or property into a defined piece of XHTML delimited by the closing element identifier. This is normally achieved by using the `div` and `span` elements to conjoin both the presentation style (CSS) and the semantic meaning (ontology) to the user (see Fig. 6).

**Non Presentational XHTML Attributes** We can leverage the implicit information contained in the names of XHTML elements if we have a corresponding ontology. Elements that are non-presentational (like `<address>`) can be used to encapsulate meaning within the page (see Fig. 6).

**Individuals** Unique individuals are defined by use of the anchor element where the `href` attribute is used to point to the URI or MAILTO of the unique item. If `http` / `mailto` are used then the link will be click-able. If `uri` is used then the link is not click-able (see Fig. 6).

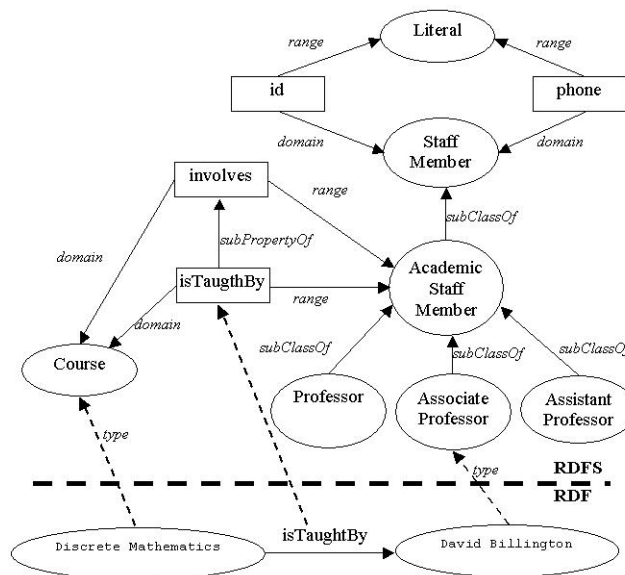


Fig. 5. RDF and RDFS Layers Taken from ‘A Semantic Web Primer’ Pg 84

We include namespaces in XHTML documents so that multiple ontologies can be used to describe one document. To implement this we use the `link` element of the XHTML header section.

```

<link rel="ontology" type="NAMESPACE" href="LOCATION"/>
<link rel="ontology" type="xmlns:owl"
href="http://www.w3c.org/2002/07/owl#" />

```

The first line represents the format the second an example. The `rel` attribute is always ‘ontology’ as this differentiates it from stylesheets and the like. Elements can be related to a namespace by using either the namespace identifier in the class attribute of the enclosing `div` element or by joining the namespace to the attribute name using an underline (.). The suggested approach provides a mechanism for encoding “lightweight” information. Of course this approach has its limitations – we can capture simple instantiation of atomic classes along with property assertions, but not richer assertions such as instantiation of arbitrary class expressions. We stress that this is not intended as a replacement for other representations but is a complementary mechanism. For example, we can still expect the class and property definitions in the ontology to be encoded using existing approaches such as RDF/XML.

Designers often want to adjust the visual design of a web-page without altering the actual meaning. We support the idea that this ad-hoc visualization can be handled by specialising ontological concepts with visual extensions if required.

## 5 Example

As a preliminary case study let us consider the simple ontology taken from ‘A Semantic Web Primer’ (in press) page 84 Figure 3.6 and recreated here for convenience as Fig. 5. Let us now see how information culled from ‘David Billington’s’ Web page can be annotated (see Fig. 6) such that the semantics of the instance are available for inference following the ontology in Fig. 6. We can see that this information is just a general description of the course information. However, by adding a `div` element we enclose the information such that the enclosure implicitly relates any enclosed sub-elements. Secondly, we see that a `span` and anchor element are introduced to denote `Course` and `IsTaughtBy`. We can via the ontology now infer the conceptual range (using ABox reasoning via ‘Racer’) that discrete mathematics is taught by the associate professor David Billington, and what is more, so can assistive agents. This seems to represent what we want to say from a reasoning approach and when presented it is displayed correctly and with no additional overhead for the designer.

```

<div class="leftcolumn">
  <span class="course">Discrete Mathematics</span>, taught by
  <a class="IsTaughtBy" href="mailto:dbillington@uni.edu">
    David Billington</a>, is a second year course designed for Computer Science
    students who need a more formal mathematics training.
</div>
<div class="aboutnote">
  <a class="associate professor" href="mailto:dbillington@uni.edu">
    David Billington</a> is Associate Professor of Information Systems.
</div>

```

**Fig. 6.** XHTML Code

## 6 Why Does This Approach Aid Visually Impaired Users?

By knowing the meaning of the information that is being encountered visually impaired users can perform their own triage on that information. As we have previously mentioned, web pages are read from top left to bottom right. If there is a lot of information on the page then the user can get lost, disoriented, or at least frustrated with their progress through this information. By presenting the meaning of the information using standard transcoding methods, users can choose which information is important to them, not the visual designer.

## 7 Conclusions

Our system suggests a method of encoding lightweight mark-up into webpages to incur a low cost semantic benefit. With the meat of the information design being abstracted from the graphic / web designer the system has given a taster of how semantics can be represented within web-pages. Additionally, we also show how this can be achieved without incurring a significant overhead with regard to marking-up that semantic information and have it validate to XHTML 1.0 strict.. We propose that the inclusion of semantic information directly into the XHTML is the only way to assist visually impaired users access web pages while not increasing or compromising the creation activity of authors and designers. Indeed we show the first stage in a more elaborate system to enable semantic information to be freely accessible by all users.

## References

1. Web content accessibility guidelines 1.0, 1999. <http://www.w3.org/TR/1999/WAI-WEBCONTENT/>.
2. M. Altheim and S. B. Palmer. Augmented Metadata in XHTML, 2002. <http://infomesh.net/2002/augmeta/> - valid 2004.
3. T. Berners-Lee. RDF in HTML, 2002. <http://www.w3.org/2002/04/htmlrdf> - valid 2004.
4. M. Brambling. Mobility and orientation processes of the blind. In D. H. Warren and E. R. Strelow, editors, *Electronic Spatial Sensing for the Blind*, pages 493–508, USA, 1984. Dordrecht, Lancaster, Nijhoff.
5. C. Chen. Structuring and visualising the www by generalised similarity analysis. In *Proceedings of the 8th ACM Conference on Hypertext and Hypermedia*, New York, USA, 1997. ACM Press.
6. A. Chieko and C. Lewis. Home page reader: IBM's talking web browser. In *Closing the Gap Conference Proceedings*, 1998.
7. D. Connolly. HyperRDF: Using XHTML Authoring Tools with XSLT to produce RDF Schemas, 2000. <http://www.w3.org/2000/07/hs78/> - valid 2004.
8. M. Dean and G. Schreiber. OWL Web Ontology Language Reference. W3C Recommendation, World Wide Web Consortium, 2004. <http://www.w3.org/TR/owl-ref/>.
9. A. G. Dodds. The mental maps of the blind. 76:5–12, January 1982.
10. R. Furuta. Hypertext paths and the www: Experiences with walden's paths. In *Proceedings of the 8th ACM Conference on Hypertext and Hypermedia*, New York, USA, 1997. ACM Press.
11. Henter-Joyce, Inc. *Jaws*. <http://www.hj.com>.
12. N. Kew. Why Validate?, 2002. <http://lists.w3.org/Archives/Public/www-validator/2001Sep/0126.html> - valid 2004.
13. V. RNIB. A short guide to blindness. Booklet, Feb 1996. <http://www.rnib.org.uk>.
14. Sean B. Palmer. RDF in HTML: Approaches, 2002. <http://infomesh.net/2002/rdfinhtml/> - valid 2004.

---

# Short Papers

---





# Managing the semantics of coreference relations with Open Ontology Forge

Ai Kawazoe Asanobu Kitamoto Nigel Collier

National Institute of Informatics  
2-1-2 Hitotsubashi Chiyoda-ku  
Tokyo 101-8430 JAPAN  
{zoeai, kitamoto, collier}@nii.ac.jp

**Abstract.** In this paper, we will discuss managing the semantics of “coreference relations” in the framework of the Semantic Web. In the Semantic Web context coreference is important in integrating many kinds of information sources (of various linguistic forms, images, etc.) and helping users to share such information. In this paper we propose a knowledge model for describing the semantics of co-referential identity relations on *annotations*, and introduce the Open Ontology Forge (OOF) software to support users to manually annotate coreference in texts and between images and texts.

## 1 Introduction

In this paper, we will discuss how to manage the semantics of coreference by human experts. In the domain of natural language processing, coreference has been a key problem for computers to understand the meaning of natural language texts through anaphoric expressions that require disambiguation, and accurate identification of coreference makes it possible for computers to maximize the amount of useful information. In the Semantic Web context, “coreference” play a role not only in augmenting information, but also in integrating information sources which refer to the same class instance and helping users to share the information. So far, the Semantic Web initiative [1] has enabled RDF [5] and OWL [7] to become a common meta-data standard for sharing knowledge on the World Wide Web, and this allows for the explicit description of concepts, properties and relations in an *ontology*. Instances of concepts in an ontology appear in documents in many different forms (ex. various linguistic forms, images, etc.) and in order to integrate information it is necessary to manage the coreference relations between such surface forms.

In this paper we outline a knowledge model for describing the semantics of coreference on *annotations* implemented within Open Ontology Forge (OOF), software to support users for annotating coreference relations by hand. The coreference cases which we aim to deal with in this work are taken from the domain of molecular biology and include 1) coreference among linguistic expressions including anaphoric relations (ex. Interleukin-2...it) and term variations (ex. Interleukin-2 vs. IL-2), 2)

multimodal coreference among texts and images (ex. a biomedical image of a cell and description of the cell in a figure legend), and 3) cross-document coreference.

## 2 Representing Coreference

In the OOF knowledge model, we represent coreference relations as illustrated in Fig.1. In this model, co-referring expressions are related to what we call a “coreference pool” which is linked to an ontological class.

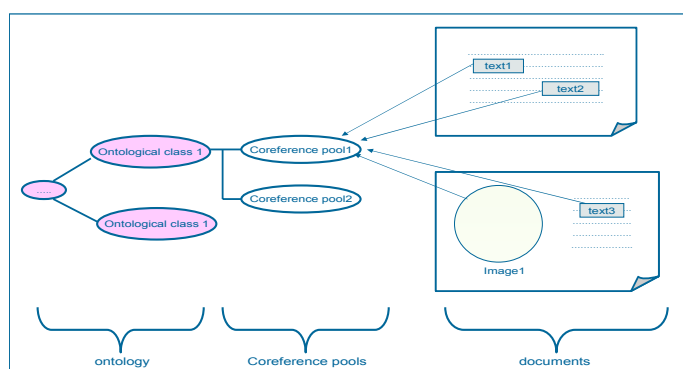


Fig. 1. Overview of the coreference annotation model.

The important feature of the model here is that the co-referring expressions in a coreference pool have the same status (unlike “coreference sets” in Conceptual Graph) and there is no hierarchical relation among them. Each of the expressions, regardless of their type, are independently related to a coreference pool. From a practical point of view, we can say that this style of annotation is one of the easiest ways to represent cross-modal and cross-document coreference. Also in intra-text coreference annotation, it will reduce a lot of user efforts and make no assumption of linguistic training, especially in that it does not require specifying antecedents for pronouns or canonical forms for names, unlike other schemes such as MUC [4].

## 3 OOF annotation support tool

We have developed the OOF software to support users in annotating coreference relations by hand. The main features of the software are 1) an integrated function for ontology creation and text/image annotation, and 2) a user interface which realizes an easy way of annotation for cross-modal items. OOF has a full Web-browser view of a html document, along with a window showing the ontology and coreference pools. Users can create the class hierarchy by expanding the root class and defining new class names. The software allows for two modes of instance capture: the named entity annotation mode and the coreference annotation mode. In both, users can make text annotation by dragging and dropping the selected part of text (Fig.2) to the class

in the taxonomy. For image annotation, OOF has an SVG editor window for selecting a part of image and editing properties as in Fig. 3. The selected part of the image can be linked to an ontological class or a coreference pool in the same manner as texts: a simple drag-and-drop fashion.

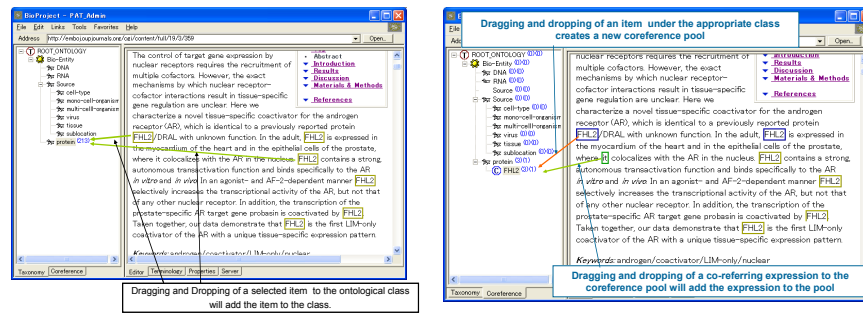


Fig. 2. (left) Named entity annotation and (right) coreference annotation with OOF.

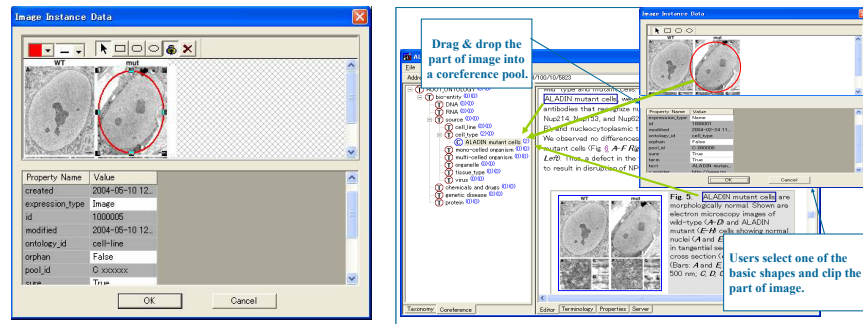


Fig. 3. (left) A window for image editing. (right) Annotation of text-image coreference

The OOF knowledge model has several similarities to other ontology editors such as Protégé-2000 [6] and OntoEdit [8]. What distinguishes OOF is its focus on providing support for content annotation. An annotation in OOF is regarded as an *instance* with a linkage to the document and tracking information about the annotator, recorded by the pre-defined properties including (1-3) below, whose values are automatically assigned by the software. Annotations are grouped within ‘coreference pools’ via (4), and (5) and (6) to record characteristics of instances.

1. *XPointer* takes an XPointer value [2] and relates an annotation to the resource
2. *author* is a name of the author of the annotation
3. *ontology\_id* relates an annotation to an ontological class
4. *pool\_id* takes an ID of the *coreference pool* to which the item belong
5. *expression\_type* specifies the subtype of the instance (*name*, *pronoun*, *image*, etc.)
6. *svg* records a description of the annotated part of image in SVG [3].

## 4 Discussion

We have conducted two kinds of annotation case studies using OOF: 1) annotation of biomedical articles (text only), and 2) annotation of documents which include images of Buddhist statues in Dunhuang. We had two biologists and some experts in the cultural heritage domain as annotators, who are not experts in NLP. In the former experiment, the annotators seemed to have a good understanding of the notion of coreference pool, and they made coreference annotations in the way we have intended. However, since the current version of OOF does not have semi-automatic annotation function for coreference, some coreference relations were left unannotated. The latter experiment have revealed some limitations of the current OOF knowledge model. For example, when annotating an image of a Buddhist statue, users sometimes want the same image to be both under a class of styles (ex. Ghandara\_style) and under a class of motifs (ex. Bodhisattva), but OOF does not allow such annotation. Further, relations such as a part-of relation seem necessary, where we are describing the relationship between a Buddhist statue and its halo. We should reconsider how to manage these situations with OOF.

## 5 Concluding remarks

Open Ontology Forge was released in February 2004 and available freely for download from <http://research.nii.ac.jp/~collier/resources/OOF/index.htm>. We plan to release a new version of OOF in December 2004 with several functions including multi-document annotation.

## References

1. Berners-Lee, T., Fischetti, M., and Dertouzos, M. 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. Harper, San Francisco, September.
2. DeRose, S., Maler, E., and Daniel, R. eds. 2001. XML Pointer Language (XPointer) Version 1.0. W3C candidate recommendation, 11th September
3. Ferraiolo, J., Fujisawa, J., Jackson, D. 2003. Scalable Vector Graphics (SVG) 1.1 Specification, *W3C Recommendation* 14 January 2003. (<http://www.w3.org/TR/SVG11/>)
4. Hirschman, L., and Chinchor, N. 1997. MUC-7 Coreference Task Definition, Version 3.0. In *Proceedings of the Seventh Message Understanding Conference*.
5. Lassila, O., and Swick, R. eds. 1999. Resource Description Framework (RDF) Model and Syntax Specification. Recommendation, W3C, February.
6. Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., and Musen, M. A. 2001. Creating semantic web contents with Protégé-2000. In *IEEE Intelligent Systems*, 16(2):60–71.
7. Smith, M. K., Chris Welty, C., McGuinness, D. L. (eds.) OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004. <http://www.w3.org/TR/owl-guide/>
8. Sure, Y., M. Erdmann, J. Angele, S. Staab, S. Studer, and D. Wenke. 2002. OntoEdit: Collaborative ontology engineering for the semantic web. In I Horrocks and J. Hendler, editors, *The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy*, Lecture notes in Computer Science. Springer-Verlag,

# MetaDesk: A Semantic Web Desktop Manager

Robert MacGregor, Sameer Maggon, Baoshi Yan

Information Sciences Institute  
4676 Admiralty Way, Marina del Rey, CA 90292, U.S.A.  
{macgregor, maggon, baoshi}@isi.edu

## Abstract

MetaDesk is an RDF authoring tool that facilitates entry of facts, rather than construction of ontologies. MetaDesk places no restrictions on vocabulary—users can invent terms on-the-fly, which the system converts into underlying RDF structures. Knowledge entry focuses on the creation of semantic structures that form scaffolding both for retrieving and interpreting facts. The most common hierarchic relationships turn out to be partonomies (whole/part structures) and set membership (as opposed to the traditional is-a hierarchies and class memberships). MetaDesk is also a semantic desktop that includes references to folders and documents within its knowledge base. We have found that the same semantic structures are appropriate for organizing desktop information

## Introduction

A year ago we experimented with a tool for attaching RDF metadata to Web pages that used Protégé [Eriksson 1999] as the data entry (authoring) component. The tool required that a class be selected for instantiation as a prerequisite to knowledge entry. Our experiment was a failure, for two reasons. We found that the ontology-driven paradigm resulted in creation of artificial classes (often suffixed with the term “Annotation”) that drew an artificial boundary between the objects being annotated and the metadata descriptions. Worse, it was just annoying—the effort to select a class before typing in an annotation discouraged use of the tool.<sup>1</sup>

In response, we invented a new tool, MetaDesk that makes RDF data authoring as quick and painless as possible. We use MetaDesk to record the kinds of metadata we generate during everyday tasks. We quickly discovered that the kinds of knowledge structures users (the authors, in this case) produced with the tool differ from the structures found in typical RDF databases. Currently, we are using MetaDesk as a personal information manager to keep track of projects, proposals, to-do lists, slides, etc., and as a launching pad for quickly bringing up specific folders and documents (like Windows shortcuts, only better organized and optionally possessing metadata annotations). Our intention is to add one or more additional knowledge sharing capabilities to MetaDesk, and then release it as a generic tool for managing

information and for collaborating with other MetaDesk users.

**Example:** MetaDesk provides two metaphors for entering information—users can create “nodes” (represented internally as RDF resources) that are arranged in a hierarchy, and they can attach attribute-value pairs to nodes.

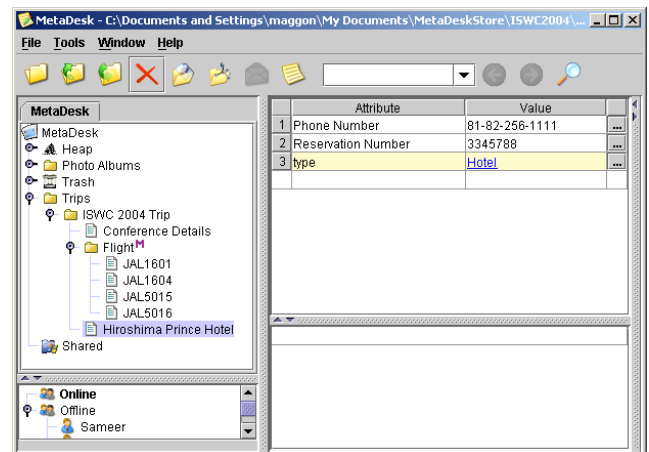


Figure 1: Recording Trip information in MetaDesk

Suppose you are planning a trip to the forthcoming ISWC conference and you need to record information about the trip in an organized fashion. Details could include flight carrier, confirmation number, hotel preferences, prices etc. In addition, you would like the information to be represented in such a way that restructuring of the data is feasible. Storing such information in the current RDF authoring tools is a tedious process. As opposed to directly writing the information in the tool, you first have to create a myriad of classes and properties like Trip Class, Flight Class, and Hotel Class etc. Also, the domain and range constraints of the properties have to be specified. Further more, the ontological information is not very obvious in particular cases. For example, it is difficult to name the relationship between Trip class and Flight class and between Trip class and Hotel class. As a result, a naive user, or one in a hurry, would prefer to create such information in a text format than recording it in an ontology-driven RDF authoring tool. Our tool excels in simplicity, providing an efficient data entry paradigm.

<sup>1</sup> These artificial classes were created to provide domains for “annotation properties”.

Recording the information in this example is easy and fast with MetaDesk. One can simply create a Trip node and add some child nodes to it. The child nodes could be a Flight node, a Hotel node and a Conference node. One can attach other information to individual nodes; for example, add a confirmation number to the Flight node. The resultant hierarchy is shown in Figure 1.

MetaDesk is all RDF-based--although users enter the data rather quickly without knowing anything about RDF, the created data is converted to RDF triples. Below we list the underlying RDF triples (in N3 format for readability) for the information shown in Figure 1. The "parentChild" links specify that under the "ISWC\_2004\_Trip" node are three nodes: "Flight" node, "Hiroshima Prince Hotel" node and "Places\_to\_Visit" node. Under "Flight" node are four other nodes representing individual connecting flights: "JAL1604", "JAL5016", "JAL5015", and "JAL1601". There are also RDF triples defining the reservation number and phone number for the hotel, etc.

```
myNS:Trips
  rdfs:label "Trips" ;
  sew:parentChild myNS:ISWC_2004_Trip .

myNS:ISWC_2004_Trip
  rdf:type myNS:Trip ;
  rdfs:label "ISWC 2004 Trip" ;
  sew:parentChild myNS:Hiroshima_Prince_Hotel
    ,myNS:Places_to_Visit
    ,myNS:Flight .

myNS:Hiroshima_Prince_Hotel
  rdf:type myNS:Hotel ;
  rdfs:label "Hiroshima Prince Hotel" ;
  myNS:Phone_Number "81-82-256-1111" ;
  myNS:Reservation_Number "3345788" .

myNS:Places_to_Visit
  rdf:type sew:Desktop_Folder ;
  rdfs:label "Places to Visit" ;
  fileNS:fullpath "C:\\Documents and
Settings\\maggon\\My Documents\\Places to Visit".

myNS:Phone_Number rdfs:label "Phone Number".
myNS:Flight
  rdfs:label "Flight" ;
  sew:parentChild myNS:JAL5016 , myNS:JAL1604 ,
myNS:JAL5015 , myNS:JAL1601 .
```

## Mapping MetaDesk to RDF

MetaDesk is represented as “triples all the way down”—every link in MetaDesk maps to a triple. A new node is created by highlighting an existing node, and explicitly typing the name of a child node, or by dragging something (a Web page, PDF file, Word Document, etc. or another node) onto the highlighted node. MetaDesk

consciously imitates the gestures, look and feel used to construct hierarchies using Windows Explorer.

If ‘P’ is a node, and ‘C’ is one of its children, the link between them is represented by a triple of the form <P, R, C> where ‘R’ is either ‘parentChild’ or one of its subproperties. The ‘parentChild’ relationship is roughly definable as the most-general, directed structural relationship. As such it subsumes more specific relations such as whole/part, class/subclass, set/set member, or folder/subfolder. We originally assumed that it should also subsume the class/instance property (the inverse of ‘rdf:type’), but when viewing children of a class, we found that we wanted to see only its subclasses, not mixed in with its instances. A node can have multiple parents (it occupies the object position of multiple ‘parentChild’ triples). A special node called ‘Heap’ exists as a catch-all—an RDF resource that does not have a parent node is considered to be “on the heap”. This is handy for operations such as tabbed search that assumes that each node it displays is located somewhere in the hierarchy.

Each node N has zero or more attributes, represented by triples of the form <N, R, V> where ‘R’ is not a subproperty of ‘parentChild’ (or of its inverse). There are no restrictions on what attributes can be attached to a node (i.e., violations of domain constraints may be flagged, but are not forbidden). Users are encouraged, but not required, to fill in the attributed named “type”, which denotes the property ‘rdf:type’. A future version of MetaDesk will semi-automate the filling-in of type attributes.

RDF structures in their raw format are not readable, so we want to hide all details of RDF from users, including URIs and namespaces. Hence, all non-literal names that a user sees in MetaDesk (names attached to nodes in the hierarchy, attributes, and in attribute value position) correspond to RDF ‘labels’. Underneath, each label ‘N’ maps to a URI ‘U’, and MetaDesk asserts the triple <U, rdfs:label, N>. Some labels have semantics built in, e.g., “type” maps to ‘rdf:type’ and “parent class” maps to ‘rdfs:subClassOf’. By default, a label “xxx” that does not match an existing label is mapped to the URI ‘myns#xxx’, where “myns” is the URI for a user’s personal namespace.

An attribute value ‘V’ is stored as a literal (a string) if the relevant range information references a literal class (a subclass of ‘rdfs:Literal’), or as a resource if the range indicates a non-literal. If there is no range information, then the system first looks for a label matching ‘V’, creating a matching resource if there is. Otherwise, ‘V’ defaults to a string, but the user can convert a literal value it into a new resource (by gestures provided by MetaDesk) any time. Values representing brand-new resources are considered a part of the “heap”.

## Importing Data

Arbitrary RDF files can be dropped into a MetaDesk hierarchy, but MetaDesk will not know which new resources to treat as nodes within the hierarchy. Instead, all of these nodes are assigned to the “heap”. An exception is Class and Property resources. These are entered under the Ontology node, below either ‘owl:Thing’ or ‘sew:Attribute’(‘sew’ is the nickname for the namespace that is internally used by MetaDesk).

Arbitrary XML files can also be dropped into a MetaDesk hierarchy. These are automatically converted into RDF, with the top-most tag forming the root resource. The ‘parentChild’ Property is used to represent the relationship between tags and subtags (except when the subtag represents a literal). For example, for the following XML

```
<trip>
  <hotel confirmation="39880A78B">
  <flight fltnum="884"
    confirmation="S38BN04">
    <carrier>America West</carrier>
  </flight>
</trip>
```

Our translator would create resources of RDF type ‘myns:Trip’, ‘myns:Hotel’, and ‘myns:Flight’, with ‘parentChild’ links from the Trip resource to the Hotel and Flight resources. Each of the three attributes is converted into the obvious RDF triple. The Flight resource is linked via a triple to the string “America West” via a property named ‘myns:carrier’.

## Interaction with Windows Applications

The primary means provided currently for interacting with desktop objects are (i) drag and drop actions to/from the desktop and (ii) launching applications by double-clicking on nodes denoting them that reside in the hierarchy. Windows folders are a special case—when a Windows folder is dropped into the hierarchy, the corresponding MetaDesk node can materialize additional child nodes (on demand) corresponding to the contents of the folder when the node is “opened”. Annotations attached to folders are persistent, but the ‘parentChild’ links that relate folders and subfolders are not stored persistently (to save space). Move and copy operations on folder nodes cause corresponding changes in the underlying Windows desktop hierarchy.

A complete semantic desktop should demonstrate similar levels of integration for other applications such as e-mail. Ideally, one or several commercial e-mailers could be integrated with MetaDesk. Alternately, one could mimic Haystack [Quan 2003] and implement an entire e-mail application (as a plug-in) within MetaDesk.

## Plug-ins

MetaDesk architecture can be extended by using plug-ins to create alternate displays for the top and bottom panes to the right of the hierarchy pane. Plug-ins are associated with particular data types – when a node is highlighted, the default display plus all relevant plug-ins that correspond to the type of that node are presented as options. MetaDesk also enables users to select a *default* plug-in for the data type; this way MetaDesk remembers the user’s choice for the next time. We have developed a photo viewer plug-in (Figure 2) that enables users to view the thumbnails of the images organized in MetaDesk. Whenever the user clicks on the Album Node (a node with the rdf:type – Photo Album) in the hierarchy, the photographs are shown in the bottom pane. User can view as well as annotate the pictures thus embracing an interactive session.

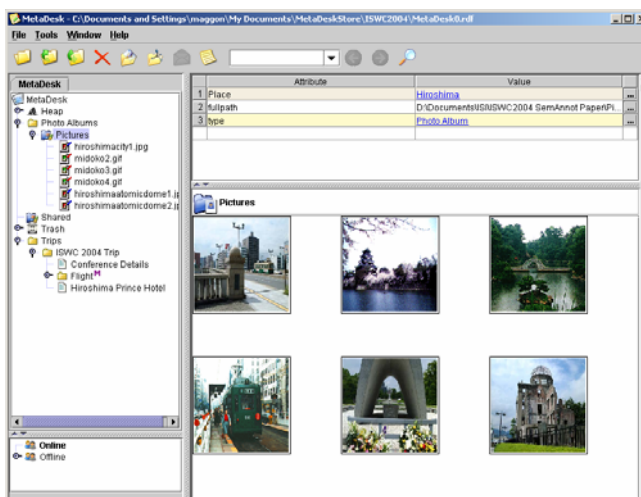


Figure 2: Photo-Plugin for displaying graphics resources

MetaDesk allows a user to choose the plug-in for any data-type (or class). For example, a user might want to associate the photo viewer plug-in with the nodes that have the type Photo Graphs instead of Photo Album. This leverages the ease of customizing MetaDesk according to personal preference. In addition to developing plug-ins for specific data types, one might consider writing a plug-in that enforces type restrictions on its input, or one that displays Protege-like templates in place of the free-form attribute editor that comes standard in MetaDesk. Such plug-ins would enable MetaDesk to mimic more traditional Semantic Web RDF editors. Thus, MetaDesk uses these plug-in points to keep track of the user's working behavior and provide self-personalization.

## Ongoing Work

**Search:** Currently, MetaDesk supports keyword search. When searching for a match to the keyword “xxx”, a triple <S, P, V> matches if one of S, P, or V has a label



containing “xxx” as a substring, or if V is a literal value that contains “xxx”. Results may be in the form of a tabbed search, wherein each hit of the ‘tab’ key opens the hierarchy to the location of the next matching node, or the results may be placed under a newly-created search node which can further be annotated.

**Ontology Alignment:** Philosophically, MetaDesk runs completely against the grain by promoting “ontological promiscuity” and advocating bottom-up development of ontologies. “Promiscuity” refers to MetaDesk’s encouraging users to make up their own vocabulary. In our scheme, we first let a thousand flowers bloom, and then specify semantic mappings (alignments) that say how one user’s terminology relates to another’s. We call this “grassroots alignment”, since it empowers ordinary users to build terminologies, instead of requiring ontology experts. The current MetaDesk is missing two things: (i) “carrots” that encourage MetaDesk users to align their terminology with terms used by others and to fill in the type attribute on each node, and (ii) alignment tools that make aligning terms very simple. One example of such a carrot is a search facility that exploits alignments to increase the recall of its matches. Another is a report generator that produces denser, better organized reports when alignments are taken into account. ISI’s WebScripter[Yan 2003] report generator incorporated both a carrot and an alignment capability into a single tool. Determining whether quality ontologies can be achieved bottom-up via a sufficiently mature set of carrots and alignment tools is at this point an open question—one that we believe deserves to be tested.

### Future Directions

At present, we have hypothesized that end-user alignment can compensate for the ontological promiscuity engendered by multiple MetaDesk users, enabling a community of MetaDesk users to profitably share information. This hypothesis needs to be tested. Our near term goal is to add sharing capability, and then to distribute MetaDesk to a community of users. Our supply of “carrots”—tools that encourage end-users to align with each others’ vocabulary—is still sparse. We will find out whether we are close to having a viable sharing infrastructure, or if more incentives are needed.

MetaDesk will eventually support multiple search regimens—more sophisticated ones will trade precision for user convenience (more typing yields more precision).

### Conclusion

We have introduced MetaDesk, an original RDF authoring tool. MetaDesk’s approach to RDF authoring is extreme: users immediately create metadata without

defining ontology first. Instead, it is our belief that ontologies can be created later in a bottom-up fashion, as the by-product of creating and using data, rather than a straightjacket that inhibits the evolution of domain vocabularies. Compared with other ontology-driven RDF authoring tools (SHOE Annotator [Heflin 1999] OntoMat [Handschuh 2002] SMORE [Kalyanpur 2003] Melita [Ciravegna 2002]), MetaDesk is more ordinary-user friendly, more flexible in metadata creation, and provides immediate rewards to users’ effort.

MetaDesk’s metadata authoring paradigm allows quick data entry and organization. As a result, MetaDesk is already viable as a personal information manager. MetaDesk has been extended as a usable semantic desktop application. It is integrated with an actual user desktop, allowing direct annotations on file systems and direct launching of applications from within it. MetaDesk’s simplicity in metadata creation as well as usefulness as a semantic desktop makes it a rewarding semantic web application.

### References

- F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks. *Timely and Non-Intrusive Active Document Annotation via Adaptive Information Extraction*. Semantic Authoring, Annotation and Knowledge Markup, ECAI Workshop, July 2002.
- H. Eriksson, R. W. Fergerson, Y. Shahar, and M. A. Musen. *Automatic Generation of Ontology Editors*. 12th Banff Knowledge Acquisition Workshop, 1999.
- S. Handschuh and S. Staab. *Authoring and Annotation of Web Pages in CREAM*. WWW, May 2002.
- Heflin, J., Hendler, J., and Luke, S. SHOE: *A Knowledge Representation Language for Internet Applications*. Technical Report CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland at College Park. 1999.
- A. Kalyanpur, B. Parsia, J. Hendler, and J. Golbeck. *SMORE – Semantic Markup, Ontology, and RDF Editor*.
- D. Quan, D. Huynh, and D. R. Karger. *Haystack: A Platform for Authoring End User Semantic Web Applications*. International Semantic Web Conference, Oct 2003.
- B. Yan, M. Frank, Pedro A. Szekely, R. Neches, J. Lopez: *WebScripter: Grass-roots Ontology Alignment via End-User Report Creation*. International Semantic Web Conference, Oct 2003.

# **MPEG7ADB: Automatic RDF annotation of audio files from low level low level MPEG-7 metadata**

Giovanni Tummarello, Christian Morbidoni, Francesco Piazza, Paolo Puliti

DEIT - Università Politecnica delle Marche, Ancona (ITALY)

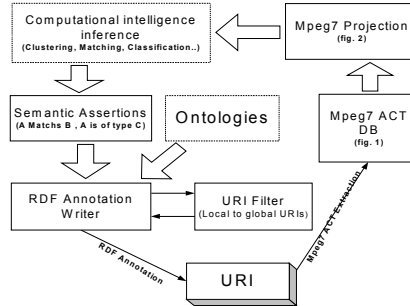
**Abstract.** MPEG-7, a ISO standard since 2001, has been created recognizing the need for standardization within multimedia metadata. While efforts have been made to link the higher level semantic content to the languages of the semantic web, a big semantic gap remains between the machine extractable metadata (Low Level Descriptors) and meaningful, concise RDF annotations. In this paper we address this problem and present MPEG7ADB, a computational intelligence/signal processing based toolkit that can be used to quickly create components capable of producing automatic RDF annotations from MPEG-7 metadata coming from heterogeneous sources.

## **1 Introduction**

While MPEG-7 and the tools of the Semantic Web (Notably RDF/S) were developed concurrently, the two efforts have been largely independent resulting in several integration challenges. At data model level, MPEG-7 is directly based on XML+Schema while the tools of Semantic Web use these just as an optional syntax format while conceptually relying on graph structures. At the semantic description level, it is thanks to a later effort [8][24] that RDF/DAML+OIL mappings have been made to allow interoperability. While such mappings are possible, their scope (semantic scene description) is currently beyond anything that can be machine automated. Previous works have also shown [4] that pure XML tools are very ineffective for handling MPEG-7 data. Although the syntax is well specified by the standard, generalized MPEG-7 usability is not simple. In fact, while it is relatively easy to create syntactically compliant MPEG-7 annotations, the freedom in terms of structures and parameters is such that generally, understanding MPEG-7 produced by others is difficult or worse. For the same reason, computational intelligence techniques, which are bound to play a key role in the applications envisioned for the standard, are not easy to apply directly. As MPEG-7 descriptions of identical objects could in fact be very different from each other when coming from different sources. Recognizing the intrinsic difficulty of full interoperability, work is currently under way [3] to standardize subsets of the base features as “profiles” for specific applications, generally trading off generality and expressivity in favor of the ease and lightness of the implementation. Necessarily, this also means to give up on interesting scenarios. In this paper we address the hard problem of “semantic mismatch”, that is, techniques to “distill” concise RDF annotations from raw, low level, MPEG-7 metadata. These techniques are implemented in a set of

tools (MPEG7ADB) by which it is possible to simply build powerful RDF audio automatic annotation components feeding on MPEG-7 low level descriptors (LLDs).

## 2 The MPEG7ADB



**Figure 1.** The overall structure of the proposed architecture.

The simplified representation of the proposed architecture (as currently implemented by the MPEG7ADB project [7]) is depicted in Figure 1. URIs are both used as references to the audio files and become the subjects of the annotations produced in standard RDF/OWL format.

When the database component is given the URI of a new audio clip to index, it will first try to locate an appropriate MPEG-7 resource describing it. At this logical point it is possible to envision several alternative models of metadata research including calls to Web Services, queries on distributed P2P systems or lookup in a local storage or cache. If this preliminary search fails to locate the MPEG-7 file, a similar mechanism will attempt to fetch the actual audio file if the URI turns out to be a resolvable URL and process it with the included, co-developed MPEG7ENC library[6].

Once a schema valid MPEG-7 has been retrieved, the basic raw sequences of data belonging to Low Level Descriptors are mapped into flat, array structures. These will not only serve as a convenient and compact container, but also provide abstraction from some of the basic free parameters allowed by MPEG-7. As an example, the MPEG7 ACT type provides the basic time interpolation/integration capabilities to handle the cases when LLDs have different sampling periods and different grouping operators applied.

To exploit the benefits of computational intelligence (e.g. neural networks) and perform clustering, matching, comparisons and classifications, each MPEG-7 resource will have to be projected to a single, fixed dimension vector in a consistent and mathematically justified way. The projection blocks performs this task, best understood as driven by a “feature space request”. A “feature space” deemed suitable for the desired computational intelligence task will be composed of pairs, one per dimension, of feature names and functions capable of projecting a series of scalars or vectors into a single scalar value. Among these, the framework provides a full set of classical statistical operators (mean, variance, higher data moments, median, percentiles etc..) that can be

cascaded with other “pre processing” such as, i.e. a time domain filter. Since MPEG-7 coming from different sources and processes could have different low level features available and not necessarily those that we have selected as the application “feature space”, the projection block will attempt to recursively predicting the missing features by means of those available (cross prediction). It is also interesting to notice that when a direct adaptation algorithm is not available, cross prediction based on neural networks proves to be, for a selected number of features, a viable alternative. For a more detailed tractation see .

Once a set of uniform projections have been obtained for descriptions within the database, classical computational intelligence methods, such as those provided in the framework and used in the example application (section 9), can be applied to fulfill the desired annotation task. Once higher level results have been inferred (e.g. piece with URI “file://c:/MyLegalMusic/foo.mp3” belongs to the genre “punk ballade”) they can be saved into “semantic containers” which will, hiding all the complexity, provide RDF annotations using terms given in an appropriate ontology pre-specified in OWL notation. Finally, prior to outputting the annotation stream, the system will make sure that local URIs (e.g. “file://foo.mp3” ) are converted into globally meaningful formats like binary hash based URIs (e.g. hash “urn:md5: “ , “ed2k:// “ , etc.).

## 6 Producing annotations for the Semantic Web

Once obtained the mathematically homogeneous projection vectors representing the MPEG-7 files in the db, these can easily processed using a variety of well known techniques. While MPEG7ADB provides internal tools such as neural networks classifiers and clustering, many more can be interfaced at this point.

Among the tools provided by MPEG7ADB are those allowing the production of RDF annotations. Annotations produced by the MPEG7ADB will be of “rdf quality” that is, much more terse and qualitatively different than the original LLD metadata. Finally it is important to stress the importance of explicit context stating when delivering computational intelligence derived results on the Semantic Web. Virtually all the computational intelligence results are in fact subjects to change or revision according to the local state of the entity providing the annotation (e.g. the extraction settings). As new knowledge or settings could make previously obtained results invalid, this sort of inference is by nature nonmonotonic. Although the RDF framework is monotonic, it is known that results coming from nonmonotonic processes can be still mapped as long as context information are provided .

## 8 Implementation and conclusions

In this paper we discussed some of the challenges associated with making use of MPEG-7 low level audio descriptors to provide RDF annotations. Furthermore, we introduce MPEG7ADB, a library by which it is possible to create automatic RDF annotation components feeding not on actual (e.g. PCM or MP3) audio sources but on low

level MPEG-7 metadata descriptions. Sophisticated adaptation capabilities are provided to compensate for the many free parameters of the MPEG-7 standard itself. With these capabilities, “profile less” use can be made which fits the picture of the Semantic Web as also made of heterogeneous devices

MPEG7ADB has been implemented in Java (see [5] on why this is also computationally acceptable) and is available [7] for public use, review, suggestions and collaborative enhancement in the free software/open source model. Among the examples provide in the MPEG7ADB is a *Voice recording quality annotation component* . This purely demonstrative example, shows how a full RDF/MPEG-7/Neural Network audio annotation component can be built in approximately 40 lines of source code using MPEG7ADB. For lack of space the source code or an accurate description cannot be given directly here but is available at [7] and . Being, to the best of our knowledge, currently the only available tool with these capabilities, MPEG7ADB is hard to compare it directly but we believe it to be a good starting point for both implementation and research into audio MPEG-7 / Semantic Web annotation components.

## References

- [1] ISO/IEC JTC1/SC29/WG11 N4031. MPEG-7 (2001)
- [2] ISO/IEC JTC1/SC29/WG11 N5527, MPEG-7 Profiles under Consideration, March 2003, Pattaya, Thailand.
- [3] Utz Westermann, Wolfgang Klas. “An analysis of XML database Solutions for the management of MPEG-7 media descriptions”ACM Computing Surveys (CSUR) Dec. 2003.
- [4] Ronald F. Boisvert, Jose Moreira, Michael Philippsen, and Roldan Pozo. “Java and Numerical Computing” IEEE Computing in Science and Engineering March/April 2001
- [5] Holger Crysandt, Giovanni Tummarello, MPEG7AUDIOENC – <http://sf.net/projects/mpeg7audioenc>
- [6] G.Tummarello, C.Morbidoni, F.Piazza – <http://sf.net/projects/MPEG7ADB>
- [7] Jane Hunter, “Enhancing the semantic interoperability through a core ontology”,. IEEE Transactions on circuits and systems for video technologies, special issue. Feb 2003.
- [8] Ralf Klamma, Marc Spaniol, Matthias Jarke “Digital Media Knowledge Management with MPEG-7”. WWW2003, Budapest.
- [9] G. Tummarello, C. Morbidoni, P. Puliti, A. F. Dragoni, F. Piazza “From Multimedia to the Semantic Web using MPEG-7 and Computational Intelligence”, Proceedings of Wedelmusic 2004, IEEE press, Barcellona
- [10] J. Lukasiak, D. Stirling, M.A. Jackson, N. Harders. “An Examination of practical information manipulation using the MPEG-7 low level Audio Descriptors” 1st Workshop on the Internet, Telecommunications and Signal Processing
- [11] Classification Schemes used in ISO/IEC 15938-4:Audio, ISO/IEC JTC 1/SC 29/WG 11N5727, Trondheim, Norway/Jul 2003
- [12] J. Hunter, "An RDF Schema/DAML+OIL Representation of MPEG-7 Semantics", MPEG Document: ISO/IEC JTC1/SC29/WG11 W7807, December 2001, Pattaya
- [13] H. Crysandt, G. Tummarello, F. Piazza “An MPEG7 Library for Music”, 3<sup>rd</sup> MUSICNETWORK Open Workshop. Munich, 13-14 March 2004.
- [14] J. Hunter, C. Lagoze, "Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles", WWW10, HongKong, May 2001.
- [15] J. van Ossenbruggen, F. Nack and L. Hardman.” That Obscure Object of Desire: Multimedia Metadata on the Web (Part I and II)”, IEEE Multimedia, to be published in 2004,

# Action: A Framework for Semantic Annotation of Events in Video

Melanie Feinberg and Ryan Shaw

School of Information Management and Systems, U.C. Berkeley  
feinberg@alumni.sims.berkeley.edu; ryanshaw@sims.berkeley.edu

**Abstract.** We propose a model for semantic annotation of *events*, such as weddings or birthday parties, as depicted in video. Our framework consists of an event taxonomy, implemented as a faceted classification, and an event partonomy, implemented using the ABC ontology proposed by Lagoze and Hunter [1]. Our approach enables the annotation of a low-level physical action depicted in video, such as a kiss, to be linked to its higher-level event context (such as the kiss that signifies the conclusion of a Western wedding ceremony).

## 1. Introduction

This paper describes an attempt to develop a semantically rich model for annotating events in video. Taking our cue from cognitive psychology research on event perception, we use a combination of taxonomy and partonomy for our event annotation model. We also take advantage of the faceted classification structure from information science to enable robust querying and differentiation of similar events without specifying all event possibilities in advance. Our original taxonomy enables discrimination of events on seven key levels (facets). The facet structure both facilitates fine-grained distinctions between events and enables recognition of broad commonalities. Finally, we use a multi-layered partonomy, familiar from artificial intelligence, that uses the existing ABC ontology [1] for expression as RDF. Our partonomic structure relies on principles from cognitive psychology research to segment events into logical, recognizable parts.

## 2 Related Work

We cite work in multiple disciplines, which reflects our synthetic approach to this project. By assimilating principles from cognitive psychology, information science, and artificial intelligence, we can create a cohesive model for event annotation.

Our approach is grounded in the work of the cognitive psychologists Jeff Zacks and Barbara Tversky [2], who assert that people perceive events similarly to the way that they perceive objects. Zacks and Tversky assert that, like objects, events are perceived according to two sorts of hierarchical structures. Events are structured taxonomically (that is, with superordinate, basic, and subordinate categories, as initially described by Eleanor Rosch [3]) and partonomically (divided into salient parts, as described by Tversky and Barbara Hemenway [4]). Zacks, Tversky, and Iyer [5] conducted experiments to show that test subjects viewing videotaped events segmented the events in predictable, regular ways.

In our framework, the taxonomic part of the annotation clarifies an event in relation to other types of events (for example, weddings and birthday parties are both celebrations, while basketball is a sport). In implementing our event taxonomy, we used a *faceted* structure, a form that comes from bibliographic classification [6]. The ability to create new terms through combination is a particular advantage of faceted classification. All concepts do not need to be predefined, as new concepts can be

created by combining terms from different facets. In addition, the ontology itself can be simpler and less redundant.

While faceted classifications have not yet been commonly used to describe events, the AI community has used partonomies to do so. Marvin Minsky's frames [7], Schank and Abelson's scenes and scripts [8], and Ortony and Rumelhart's event schemata [9] are examples of events being described in terms of their typical parts.

### 3. Ontology Structure

In this section, we describe our taxonomy and our partonomy.

#### 3.1 Taxonomic Structure

We designed our taxonomy to include the following facets. Each facet identifies a separate set of descriptors, organized in a hierarchy from general to more specific. In classifying an event, descriptors can be chosen from some or all of the facets.

- Time (with sub-facets Boundaries, Ordering, Recurrence, and Duration). The Time facet includes descriptors to specify temporal aspects of an event, such as whether the event has strict beginning and ending points, whether event segments can be reordered, whether the event is part of a series, and variability in the event's total extent.
- Physical Effect (with sub-facets Product and State Change). This facet describes changes in the environment as a result of the event, whether that change involves the creation of a new product (such as baking cookies) or changes to an existing object (such as repairing a clock).
- Focus. This facet differentiates between events with identifiable focal points and those without. A focal point describes an element that, if not viewed, would compromise the sense of having seen the event. For example, video of a birthday party without showing the candles being blown out would seem incomplete.
- Organization. This facet describes the differences between events that have imposed structure and those that are more improvisational. For example, this facet seeks to describe the difference between a professional basketball game and a pickup basketball game on a public neighborhood court.
- Style. This facet indicates manner. For example, a birthday celebration in the United States is structured differently from one in Mexico.
- Activity. This is the basic descriptor. Expressing the activity generically allows for subtleties to be conveyed using the Purpose facet. For example, for the activity of playing music, context could further define the event as a performance, practice, audition, and so on. These latter distinctions, which might apply to many activities, are moved into the Purpose facet, reducing redundancy in the taxonomy.
- Purpose. This facet adds a more complex semantic layer onto the generic description enabled by the Activity facet. The Purpose facet differentiates playing the piano (with no additional purpose) from a piano competition or piano concert, for example.

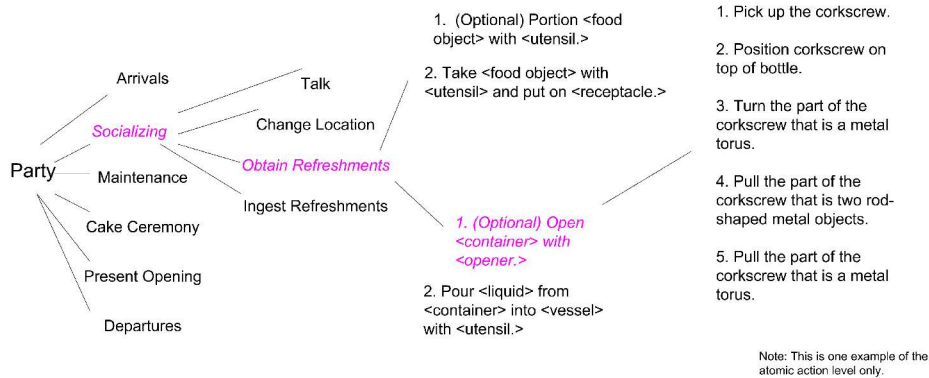
The faceted structure enables us to differentiate between events that are similar in some ways but different in others, without explicitly specifying each possible variation. The faceted structure also enables us to describe events that might be unique or are impossible to anticipate, such as a birdcage-making contest. Such an activity might occur only once in the world, but we can specify this improbable event easily by combining descriptors from different facets. We might use an Activity facet descriptor to represent carpentry, a Physical Effect descriptor to indicate the product of a birdcage, and the Purpose facet to clarify a competition.

The use of facets makes searching for related footage more robust, as the search relates to concepts, and not keywords. For the birdcage example, one could search for video of carpentry, the Activity facet descriptor, without searching on additional facets, and obtain footage of any object being created through carpentry, not just birdcages. Similarly, one could search for competition, the Purpose facet descriptor, and obtain results of math competitions, swimming races, and eating contests in addition to the birdcage-making competition. And of course, one could use all three facets for specific results.

### 3.2 Partonomic Structure

The partonomic aspect of our framework describes the structure of individual events from the taxonomy, including sub-events, actions, agents, and objects, and how they relate to one another. In creating the partonomy we used strategies for event segmentation hypothesized by Zacks, Tversky, and Iyer [5].

The primary activity level represents the basic modules of the event. The generic action level represents the basic actions within each activity. At the specific action level, we indicate the different actions required for classes of variables that are involved in implementing a generic action. For example, obtaining refreshments, a generic action, differs if the guest is obtaining a beverage or a solid food. At the atomic action level, we specify the physical actions necessary to complete a generic action for the instantiation of a specific variable.



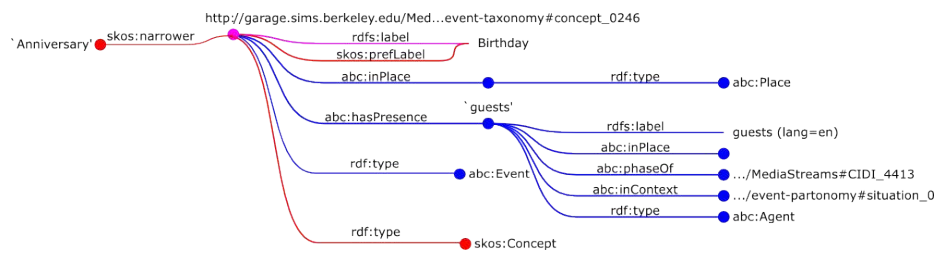
**Fig. 1.** From left to right: primary activities, generic actions, specific actions, atomic actions.

Explicitly linking actions from different levels of description potentially allows for greater recall when searching for annotated video content. For example, a query for “birthday party” could be expanded to include sub-events such as “gift-opening.” Likewise, queries made at a more specific level of description can be expanded to return footage that has been annotated at a broader level.

## 4. Implementation

After conceptualizing our taxonomy and partonomy in proof-of-concept form, we formally expressed each of them as RDF graphs [10,11] and linked them together, as shown in Figure 2. For our taxonomy we took advantage of the Simple Knowledge Organization System (SKOS) Core, an RDF vocabulary developed for thesauri [12], while for the partonomy we utilized the aforementioned ABC Ontology [1]. The results can be browsed interactively at [13].





**Fig. 2.** An excerpt of the graph showing how the taxonomy (red) links to the partonomy (blue).

## 5. Conclusions

Video content is difficult to search. Video annotation can help by identifying and contextualizing video content at a level relevant to users' experience. Video of events is particularly in need of contextualized annotation, because the physical actions depicted in a particular video segment may reappear in many different contexts. To enable robust search and retrieval of video events, we need a multi-layered annotation framework that combines the low-level actions that facilitate maximum reuse with the higher levels that people are more likely to identify. To accomplish this goal, we have combined an event taxonomy, which classifies events in relation to similar events, with an event partonomy, in which events are successively segmented into smaller and smaller parts. In the future, we hope to use this conceptual model as the basis for a Semantic Web application that enables collaborative annotation of events depicted in web video.

## References

1. Lagoze, C., Hunter, J. (2001) "The ABC Ontology and Model." *Journal of Digital Information* 2(2).
2. Zacks, J. M., Tversky, B. (2001) "Event Structure in Perception and Conception." *Psychological Bulletin*, 127, 3-21.
3. Rosch, E. Principles of Categorization. In E. Rosch & B. Lloyd (Eds.), *Cognition and Categorization* (pp. 27-48). (Hillsdale, NJ: Lawrence Erlbaum Associates, 1978).
4. Tversky, B., Hemenway, K. (1984) Objects, Parts, and Categories. *Journal of Experimental Psychology: General*, 113, 169-193.
5. Zacks, J. M., Tversky, B., Iyer, G. (2001) "Perceiving, Remembering, and Communicating Structure in Events." *Journal of Experimental Psychology: General*, 130, 29-58.
6. Broughton, V., "Faceted Classification as a Basis for Knowledge Organization in a Digital Environment: The Bliss Bibliographic Classification as a Model for Vocabulary Management and the Creation of Multidimensional Knowledge Structures," *The New Review of Hypermedia and Multimedia* 7, no. 1 (2000): 67-102.
7. Minsky, M. "A Framework for Representing Knowledge." MIT-AI Laboratory Memo 306, June, 1974.
8. Schank, R., Abelson, R. *Scripts, Plans, Goals, and Understanding. An Inquiry into Human Knowledge Structures*. (Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1977.)
9. Rumelhart, D. E., Ortony, A. (1977) "The Representation of Knowledge in Memory." In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.), *Schooling and the Acquisition of Knowledge* (pp. 97-135). Hillsdale, New Jersey: Lawrence Erlbaum Associates.
10. Action Taxonomy, <http://www.sims.berkeley.edu/~ryanshaw/action/taxonomy.rdf>.
11. Action Partonomy, <http://www.sims.berkeley.edu/~ryanshaw/action/partonomy.rdf>.
12. Alistair J. Miles, Nikki Rogers, and Dave Beckett, "SKOS-Core 1.0 Guide," SWAD-Europe, <http://www.w3.org/2001/sw/Europe/reports/thes/1.0/guide/>.
13. Action RDF Visualization, <http://dream.sims.berkeley.edu:8080/ryanshaw/visualize>.

# Integrating Event Frame Annotation into the Open Ontology Forge Annotation Tool

Tuangthong Wattarujeeekrit and Nigel Collier

National Institute of Informatics, National Center of Sciences,  
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan  
[tuangthong@grad.nii.ac.jp](mailto:tuangthong@grad.nii.ac.jp), [collier@nii.ac.jp](mailto:collier@nii.ac.jp)

**Abstract.** In this paper, we propose a scheme for event frame annotation integrated into the Open Ontology Forge (OOF) annotation tool. This is a key requirement for realization of knowledge description on the Semantic Web. Semantic information contained in each event frame is a set of relationships between a predicate and its arguments. As our aim is to keep OOF flexible for various types of annotation projects, the scheme proposed in this paper is designed based on the specialization three popular schemes: MUC-7's template relation, PropBank's predicate-argument structure and FrameNet's semantic frame.

## 1 Introduction

This paper provides the scheme for the annotation of event frames which define relationship information between objects or entities and their predicates indicating the event. This scheme is being integrated into Open Ontology Forge (OOF)<sup>1</sup>, a free annotation tool created in the PIA project [4].

As the Web of information readable by machines is the central concept of the Semantic Web [2], Web pages require annotation to make instances of objects and events explicit and to show the linkage to the context in which they occur. Thus, the development of annotation tools becomes a focus of the research community (e.g. GATE [6], MnM [9], OntoMat [10]). Like other semantic annotation tools, OOF tries to reduce the effort required to create semantic annotated texts and it focuses mainly on content annotation for Information Extraction (IE) as such we consider issues of large-scale knowledge mark-up, inter-annotator agreement, ease of use by non-linguistics, etc. One of the significant characteristics of OOF is that it not only supports annotation but also provides for the creation of ontology and the linkage between each instance and its occurrence in the text. To provide an environment that integrates annotated texts with ontology promotes knowledge sharing.

The basic aim within the PIA project is to create an automatic information extraction system by applying machine learning to annotated corpora [3]. At present OOF can be used to construct annotated named entities (NEs) and coreference relations [7]. It still lacks though the scheme to support the higher level IE task such

---

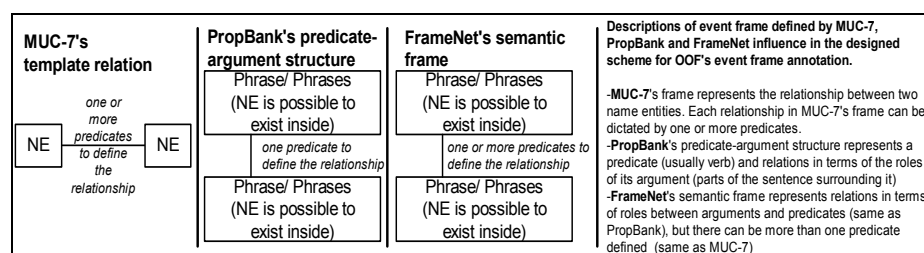
<sup>1</sup> <http://research.nii.ac.jp/~collier/resources/OOF/index.htm>

as event extraction task which provides facts in terms of relationships between entities obtained from NE and Coreference task. Therefore, the event frame annotation scheme needs to be integrated into OOF. With respect to our interest in the application of IE to special domains such as molecular biology event extraction, we currently plan to annotate molecular biology documents with semantics in terms of event frame style following an extensible version of PropBank’s predicate-argument structure [8] description.<sup>2</sup> However, we also take into account other two popular event frame styles among IE research groups (i.e. MUC-7’s template relation [5] and FrameNet’s semantic frame [1]) in forming the scheme proposed in this paper. We believe that the scheme which incorporates the key features of these three projects will provide OOF the flexibility to be used by other research groups.

## 2 The Event Frame Annotation Scheme

Annotation of event frames will give web pages some machine readable information describing a set of relationships between entities existing in each proposition. For example, if the event *buying* in the expression “John is buying flowers for Mary.” is annotated in an appropriate way, then not only can a machine understand that “John” and “Mary” are persons, but also “John” plays role as “buyer”, “flowers” plays role as “bought object” and “Mary” plays role as “receiver” in this *buying* event. So, various applications such as IE can extract these important facts for users. Similarly, the event frame annotation is capable of representing molecular events such as protein-protein interaction also.

In general, different projects have their own perspective on how to define their event frame or how a set of relationships in an event should be represented. Some special distinctions<sup>3</sup> of the event frame’s descriptions in three main projects of our focus are illustrated in Fig. 1. However, all styles can be thought of as the general frame-like styles which a relation or a set of relations is specified in an event frame by a related predicate or set of predicates and related arguments or entities.



**Fig. 1.** The abstract view of an event frame from different projects

<sup>2</sup> Reasons why we choose PropBank’s event frame style are out of scope of this paper. They will be reported elsewhere.

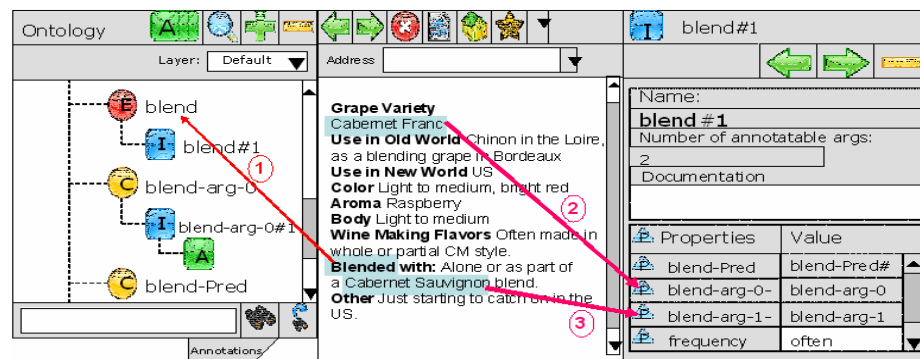
<sup>3</sup> Due to space limitation, only particular scheme’s points will be explained.

## 2.1 Knowledge Model Issues

The knowledge model of OOF has several similarities to other ontology editors such as Protégé-2000. An OOF ontology is centred around a frame-based knowledge model consisting of classes, properties (slots) and annotations. Classes are related through subsumption in a simple taxonomy. An event frame is managed as a subclass of a root class, called an Event class. The argument participated in an event are represented in the form of Event class's property. Basically, property slot in OOF is a binary relation between a domain (a class) and a range (a value data type). As an event's argument require being filled by more than one value, property of Event class is necessary to be managed as a class rather than just a binary relation. A main predicate of an event is modelled on a property of Event class as well. Moreover, instances of basic class types in OOF are not abstract concept, but a surface-level representation of the concept appearing in the document, in the form of texts or images. Contrary to other class types, event instances are abstract entities but the predicate itself and the arguments are realized as annotations in the text.

## 2.2 User Interface

As shown in the planned design of the new version of OOF in Fig. 2, OOF provides a capability to view ontology, a Web page and annotated information concurrently.



**Fig. 2.** Event annotation screen shots

For event annotation, a user has to create a particular Event class first (cf. the red E-icon). This process leads to the automatically construction of classes for the predicate and arguments which are defined as properties of the Event class. Then, a user can start annotating an event existing in a text by dragging and dropping the main predicate to the Event class for that event. As shown in Fig. 2, the text “Blended” is highlighted and assigned to the Event class *blend* (cf. arrow #1) to create an instance *blend#1* representing the event. Next, instances for event’s arguments can be captured by highlighting some text elements and then using a hot key combination. In this example, the text “Cabernet Franc” and “Cabernet Sauvignon” are captured for filling argument slots *blend-arg-0* and *blend-arg-1*, respectively (cf. arrow #2 and arrow #3).

### 3 Discussion

There have been several annotation tools such as GATE of which rather focusing on the annotation process embedded with language processing tools (e.g. POS tagger, tokeniser) than the ontology editing; MnM of which distinct property is the supporting of various representation language (e.g. DAML+OIL, RDF, WebOnto); and OntoMat which provides many of the same features as OOF including ontology editing. In contrast to these tools, the focus of our design is highlighting the role of the predicate occurred in the text as the centre of the occurring event. The event itself is represented as an individual object class rather than represented as a property of participated entity. We believe that our thinking of predicate which is much closer to linguistic perspective would allow OOF to be flexible for various event annotation styles.

The OOF has progressed forward in concerning more flexible scheme for event annotation. However, OOF still requires the extension of event annotation scheme in order to support nontrivial aspects such as to represent sequences of events.

### 4 Conclusion

We briefly presented the main scheme for semantic annotation of event frame being integrated in Open Ontology Forge (OOF) tool, with the design to cover various styles of event frames. The capability both to create ontology and to annotate texts as well as to provide the linkage from the ontology instance to where it exists in texts makes the OOF annotation tool worthwhile for Semantic Web applications. Current version is downloadable from <http://research.nii.ac.jp/~collier/OOF/index.htm>. We plan on releasing new version included event frame annotation capability in January 2005.

### References

1. Baker, C., Fillmore, C., and Lowe, J.: The Berkeley FrameNet project. In *Proc. of COLING-ACL*, Montreal (1998)
2. Berners-Lee, T., Fischetti, M., and Dertouzos, M.: Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web. Harper, San Francisco, September (1999)
3. Collier, N., and Takeuchi, K.: PIA-Core: Semantic Annotation through Example-based Learning. In *Proc. of the 3<sup>rd</sup> LREC*, Las Palmas, Spain (2002)
4. Collier, N., Takeuchi, K., Kawazoe, A., Mullen, T., and Wattarujeeekrit, T.: A Framework for Integrating Deep and Shallow Semantic Structures in Text Mining. In *Proc. of the 7<sup>th</sup> KES*, UK (2003)
5. DARPA. In *Proc. of MUC-7*, Fairfax, VA, USA (1998)
6. GATE: <http://gate.ac.uk/>
7. Kawazoe, A. and Collier, N.: An Ontologically-motivated Annotation Scheme for Coreference. In *Proc. of SWFAT*, Nara, Japan (2003)
8. Kingsbury, P. and Palmer, M.: From Treebank to PropBank. In *Proc. of the 3<sup>rd</sup> LREC*, Las Palmas (2002)
9. MnM: <http://kmi.open.ac.uk/projects/akt/MnM/>
10. OntoMat: <http://annotation.semanticweb.org/ontomat/>

---

# Position Papers

---



# Annotating OWL Ontologies

Bijan Parsia and Aditya Kalyanpur

University of Maryland, MIND Lab, 8400 Baltimore Ave,  
College Park MD 20742, USA  
{bparsia}@isr.umd.edu  
{aditya}@cs.umd.edu

## 1 Introduction

One key function of the Semantic Web is to support knowledge representation based annotations of Web “resources”. The linguistic apparatus (e.g., OWL<sup>1</sup>) used to express these annotations (e.g., OWL classes, or properties) are themselves Web resources and hence subject to annotation. To this end, we have integrated Annotea<sup>2</sup> based human and machine oriented annotations into our OWL ontology editor SWOOP.<sup>3</sup>

## 2 OWL Entities and Annotation Properties

OWL has four sorts of entity that are of primary interest to an annotator or ontology user: ontologies themselves, classes, properties, and individuals. In the DL species of OWL, the expressiveness of the language for describing each sort of entity varies greatly, from the wide range of constructors and sorts of axioms for defining classes to the minimal vocabulary (with minimal logical impact) for describing ontologies themselves. By contrast, the Full species of OWL allows for the entire language to be used for any sort of entity.

OWL DL does have one sort of assertion which can be uniformly applied to all sorts of OWL entity: owl:AnnotationProperty<sup>45</sup> based assertions. From the model theoretic point of view, such assertions are mere comments. All annotation properties are ignored by the reasoner, and they may not themselves be structured by further axioms.<sup>6</sup> owl:AnnotationProperty assertions can have as objects either individuals or datavalues, including rdf:XMLLiterals, thus can embed arbitrary XML, including RDF/XML (e.g., Annotea comments), XHTML, or SVG. The built in annotation properties rdfs:label and rdfs:comment are already extensively used in user interfaces (e.g., tool tips) and in other end user displays.

---

<sup>1</sup> The W3C’s Web Ontology Language: <http://www.w3.org/2001/sw/WebOnt/>

<sup>2</sup> <http://www.w3.org/2001/Annotea/>

<sup>3</sup> <http://www.mindswap.org/2004/SWOOP/>

<sup>4</sup> In this paper, the ‘rdf’, ‘rdfs’, and ‘owl’ are mapped to the obvious namespaces.

<sup>5</sup> <http://www.w3.org/TR/owl-ref/#Header>

<sup>6</sup> OWL DL is probably more restricted than it needs to be. For example, some form of subproperty reasoning over annotation properties is quite feasible.



### 3 Annotea for OWL

The Annotea project developed an infrastructure for the creation and sharing of out of band, fine grained, extensible, Web based annotations. The Annotea framework has two parts: an RDF based annotation format and a protocol for publishing, retrieving, and managing those annotations. We have extended the Annotea format to support machine oriented annotations for collaborative editing. We have also investigated other distribution mechanisms such as OWL annotation properties and RSS 1.0 feeds.

We have written an Annotea client plugin for SWOOP. The SWOOP plugin can publish and display annotations using the standard Annotea vocabulary (including support for various annotation types such as questions, explanations, examples, etc.) to the Annotea server or to an annotation property in an OWL document.

We have also defined an OWL ontology for a new class of annotations — ontology changes. The Annotea project did define a “Change” class of annotations, but it is designed to indicate a *proposed* change to the annotated document, with the proposal described by a chunk of HTML-marked-up natural language. In our ontology, annotations correspond to specific *edits* (assertions, deletions, modifications) made in SWOOP and SWOOP can read and *apply* those edits .

SWOOP uses the OWL API<sup>7</sup> to model ontologies and their associated entities. The OWL API separates the representation of changes from the application of changes. Each possible change type has a corresponding Java class in the API, which are subsequently applied to the ontology (essentially, the Command design pattern). These classes allow for the rich representation of changes, including metadata about the changes. We have used these classes as the basis for our changes annotations. Using this change’ ontology, SWOOP can externalize and export annotated change sets, which can then be browsed, filtered, endorsed, recommended, and selectively accepted. Thus, it is possible to define “virtual versions” of an ontology by specifying a base ontology and a set of changes to apply to it.<sup>8</sup>

### 4 Granularity

Annotea uses XPointer<sup>9</sup> to associate annotations with fine grained parts of documents. For classes, properties, and individuals, there isn’t a lot of further granularity to be had. The URI of a class gets you that class and classes do not have subranges. Class *descriptions*, as collections of axioms, have an interesting and fine grain, while as collections of RDF *triples* they have an even finer grain, though of disputed use. We plan to explore XPointer schemes<sup>10</sup> which address the definitions. It is likely that we will focus above the triple layer, i.e., on axioms.

<sup>7</sup> <http://owl.man.ac.uk/api.shtml>

<sup>8</sup> This mechanism is modeled on Smalltalk’s change records and sets.

<sup>9</sup> <http://www.w3.org/TR/xptr/>

<sup>10</sup> <http://www.mindswap.org/papers/swrp-iswc04.pdf>

# OntoSelect: Towards the Integration of an Ontology Library, Ontology Selection and Knowledge Markup

Paul Buitelaar

DFKI Language Technology  
Stuhlsatzenhausweg 3,  
D-66123 Saarbrücken, Germany  
[paulb@dfki.de](mailto:paulb@dfki.de)

A central task in the Semantic Web effort is the annotation of data and documents with appropriate semantic information (i.e. *knowledge markup* or *ontology population*) derived from one or more ontologies published on the Semantic Web. The added knowledge allows automatic procedures (agents, web services, etc.) to interpret the underlying data and/or documents in a unique, formally specified way, thereby enabling autonomous information processing.

Most of the current work in knowledge markup is concerned with annotation of concepts relative to a particular ontology that is typically developed specifically for the task at hand. Instead, a more realistic approach would be to access an ontology library and to select one or more appropriate ontologies. Although the large-scale development and publishing of ontologies is still only in a beginning phase, many are already available (see e.g. the DAML ontology library<sup>1</sup>, OWL ontology library<sup>2</sup>, or SchemaWeb<sup>3</sup>). To select the most appropriate ontology (or a combination of complementary ontologies) will therefore be an increasingly important subtask of knowledge markup.

Here we present an approach towards an integration of the collection and classification of ontologies in a dynamic web-based ontology library, methods for the selection of an ontology from this library and its use in knowledge markup. Building on the idea of the DAML and SchemaWeb ontology libraries, we aim to take this to its fullest consequence through the construction of a fully dynamic ontology library (OntoSelect) that will be updated continuously, organized in a meaningful way and with automatic support for ontology selection in knowledge markup.

The OntoSelect approach aims at providing an access point for ontologies on any possible topic or domain. However, unlike these libraries, OntoSelect is not based on a static registration of published ontologies, but instead includes a dynamic ontology crawling procedure that monitors the web for any newly published ontologies in the representation formats: RDF/S, DAML or OWL.

Collected ontologies are analyzed using the OWL API<sup>4</sup> that allows for the extraction of structure and content of any RDF/S, DAML or OWL ontology. There are cur-

---

<sup>1</sup> <http://www.daml.org/ontologies/>

<sup>2</sup> <http://protege.stanford.edu/plugins/owl/ontologies.html>

<sup>3</sup> <http://www.schemaweb.info/>

<sup>4</sup> <http://owl.man.ac.uk/api.shtml>

rently around 800 ontologies in the OntoSelect library, covering a wide range of topics and domains. Ontologies are stored in a database and are organized according to: format; ontology-, class- and property-names; class- and property-labels. The assignment of labels is unfortunately not so wide spread. However, specifically from the semantic annotation and knowledge markup perspective this is an important aspect, as automatic annotation or markup of documents crucially depends on the availability of terminology for classes and/or properties.

OntoSelect provides a dynamically updated library of ontologies that may be used in a knowledge markup process. However, as there is a rapidly increasing number of published ontologies available, it is becoming a more and more difficult task to select the most appropriate one(s). To provide semi-automatic support for this, OntoSelect includes a functionality for selecting ontologies for a given knowledge markup task, based on the following criteria that address ontology content and structure:

- *Coverage*: How many of the terms in the document collection of the particular knowledge markup task are covered by the classes and properties in the ontology?
- *Structure*: How detailed is the knowledge structure that the ontology represents?
- *Connectedness*: *Is the ontology connected to other ontologies and how well established are these?*

After selection of an appropriate ontology from the OntoSelect ontology library, a document collection under consideration will be marked up with the knowledge from this ontology. We are currently working towards an instance-based learning approach that considers knowledge markup as a classification task. Classifiers for the knowledge markup process will be generated by collecting occurrences (i.e. linguistic realizations of classes and properties: labels or class-/property-names with their linguistic contexts) from relevant text collections that are to be associated with each of the ontologies in the OntoSelect library.

A central problem to be addressed in this is the extraction of relevant terms in text and their appropriate classification by the constructed classifier. Additional problems that are to be addressed include multilinguality (e.g. the use of an English-based ontology in knowledge markup of German documents) and ambiguity (e.g. multiple definitions of the same concept in several ontologies or multiple use of the same label for different concepts within one ontology).

## Acknowledgements

This research has been supported by research grants for the SmartWeb and VieWs projects. Thanks to Thomas Eigner and Srikanth Rmaka for their work on the OntoSelect framework.

# Bootstrapping Domain Ontologies for Rapid Semantic Annotation of User-Friendly Semantic Web Content

Yuri A. Tijerino, President & CTO ([yuri@ontolligence.com](mailto:yuri@ontolligence.com))  
*Ontolligence Corp.*

## 1. Introduction

In attempting to develop tools, mechanisms and content for the Semantic Web we have to keep in mind that the requirement for machine understandability [2] is not a constraint and should not preclude usability requirements by end users. In essence, the Semantic Web should remain as distributed, self-evolving, ad-hoc, and easy to build, navigate and maintain as the World Wide Web (Web) is today. However, Semantic Web ontology languages such as XML, RDF, RDF-S, DAML+OIL, OWL, OWL-S and others require specialized expertise to understand and use. This raises three interesting problems: 1) How to bootstrap enough domain ontologies that are capable of representing the intricacies of Web information in the form of knowledge in the Semantic Web, 2) how to bootstrap enough Semantic Web content by using those ontologies to annotate Web content in a resilient manner, and 3) how to provide easy to use Semantic Web applications that are easy to use and understand by the average Web user.

Ontolligence Corp., attempts to address these problems head-on. We create tools, techniques and processes that make it possible to quickly create Semantic Web ontologies from sample ontologies and from domain specific sample Web pages. At the same time, we create automated and semi-automated tools that quickly annotate Web pages with Semantic Web ontologies in a manner that makes the pages understandable by computational mechanisms. Furthermore, we implement applications that enable both users and computational mechanisms to interactively collaborate in performing problem solving tasks that reap the benefits of Semantic Web content.

## 2. Technical Objectives

The technical objectives of Ontolligence Corp., a spin off of Brigham Young University's Data Extraction Group [4], are threefold: First, we are investigating how to technologically expand our existing mechanisms [3] to incorporate support for OWL data-extraction ontologies in a commercially viable manner. Currently we support conversion of DAML ontologies into OSM data-extraction ontologies [6]. The technical objective here is to make it easier for companies and organizations to make a transition to the Semantic Web while supporting the OWL standard.

Second, we are investigating the construction of a mechanism that supports automatic or semi-automatic generation of data-extraction ontologies in the OWL standard. Currently this is a tedious manual process that does not fit

well in commercial applications due to a high cost to benefit ratio. We have already experimented with the automated generation of OSM data-extraction ontologies [5] and are working on new techniques which takes advantage of structured data found in the Web such as tables [8, 7], to make the process of ontology generation more cost effective. The technical objective here is to make it possible for companies and organizations to create their own internal conceptual domain models (i.e. ontologies) in a timely manner without the need for specialized and costly ontology engineering expertise, which is one of the main factors preventing industry-wide investment in this area.

Third, we are developing user friendly mechanisms that allow Semantic Web users to interact with computational mechanisms to browse, search, reason and perform problem solving on the Semantic Web. We have obtained positive experimental data that indicates that it is possible to allow agents to collaborate with other agents without requiring them to share the same ontology [1]. The technical objective here is to allow humans to communicate seamlessly with agents through simple, ontology-generated Web forms to specify problems, resolve conflicts and clarify requests [9].

We are interested in discussing with, learning from and collaborating with other researchers and partners about current cutting-edge research, techniques, approaches and mechanisms that address these three particular areas.

## References

1. Al-Muhammed, M., ‘Dynamic Matchmaking between Messages and Services in Multi-Agent Systems’. Technical report, Brigham Young University, Provo, Utah.
2. Berners-Lee, T., J. Hendler, and O. Lassila: 2001, ‘The Semantic Web’. *Scientific American* **36**(25).
3. Chartrand, T.: 2003, ‘Ontology-Based Extraction of RDF Data from the World Wide Web’. Master’s thesis, Brigham Young University, Provo, Utah.
4. DEG: 2004, ‘Homepage for BYU Data Extraction Research Group’. URL: <http://www.deg.byu.edu/>.
5. Ding, Y.: 2003, ‘Semiautomatic Generation of Relilient Data-Extraction Ontologies’. Master’s thesis, Brigham Young University, Provo, Utah.
6. Embley, D., D. Campbell, Y. Jiang, Y.-K. Ng, R. Smith, S. Liddle, and D. Quass: 1998, ‘A Conceptual-Modeling Approach to Extracting Data from the Web’. In: *Proceedings of the 17th International Conference on Conceptual Modeling (ER’98)*. Singapore, pp. 78–91.
7. Tijerino, Y., D. Embley, D. Lonsdale, and G. Nagy, ‘Ontology Generation from Tables’. *Journal of World Wide Web Internet and Web Information Systems*. Submitted.
8. Tijerino, Y., D. Embley, D. Lonsdale, and G. Nagy: 2003, ‘Ontology Generation from Tables’. In: *Proceedings of the 4th International Conference on Web Information Systems Engineering*. Rome, Italy. 242–249.
9. Tijerino, Y. and M. A.-M. Embley: 2004, ‘Toward a Flexible Human-Agent Collaboration Framework with Mediating Domain Ontologies for the Semantic Web’. In: *Proceedings of ISWC’04 Workshop on Meaning Coordination and Negotiation*. Hiroshima, Japan.

# Towards an Integrated Corpus for the Evaluation of Named Entity Recognition and Object Consolidation

Knud Möller<sup>1</sup>, Alexander Schutz<sup>2</sup> and Stefan Decker<sup>1</sup>

<sup>1</sup> Digital Enterprise Research Institute, National University of Ireland, Galway  
knud.moeller@deri.ie, stefan.decker@deri.ie

<sup>2</sup> Institut für Allgemeine Linguistik, Universität des Saarlandes, Saarbrücken  
schutz@coli.uni-sb.de

## 1 Introduction

When faced with the task of incorporating legacy web data from existing HTML pages into the Semantic Web (SW), a widespread approach is to use Information Extraction (IE) and Named Entity Recognition (NER) techniques. Natural language texts are annotated automatically or semi-automatically, and thus formal data is extracted from the texts. While this allows to add new sets of data to the SW, the process cannot stop there. It is necessary to integrate the newly created formal data with existing formal data, i.e. to identify entities which are identical in both sets. To summarize, two main problems have to be tackled to allow the integration of information from unstructured data into the SW:

1. Find the set of entities  $E_D$  in a document (NER), and probably detect co-reference chains within the document.
2. Find matches between the elements of  $E_D$  and entities in a pre-existing knowledge base.

In order to evaluate any system trying to tackle both of these problems (e.g. KIM [1] or Sementag and Seeker [2]), conventional corpora are not suitable, since these are mostly tailored towards IE and NER only. These corpora can be used to evaluate a system's performance on an inner-document basis, i.e. how well it can detect entities in a document and probably chains of co-reference between them. However, what is needed is a means of evaluating a system with respect to how well it is able to match between the entities in a document and corresponding entities in a database. This problem falls into the area of Object Consolidation. We therefore propose a novel kind of corpus, which we will call an **Integrated Corpus for Named Entity Recognition and Object Consolidation**. The first incentive for proposing such a corpus came when we were looking for a way to evaluate the Geco project [3].

## 2 An Integrated Corpus

Our proposed integrated corpus consists of two interrelated parts:

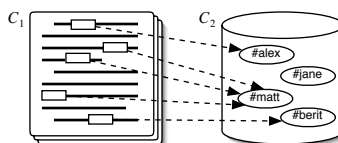
- An annotated textual corpus  $C_1$  for the evaluation of IE/NER components. This part of the corpus will be very similar to traditional corpora like MUC<sup>3</sup> or ACE<sup>4</sup>.

<sup>3</sup> MUC6 see <http://wave.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T13>, MUC7 see <http://wave.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T02>

<sup>4</sup> see <http://www.ldc.upenn.edu/Projects/ACE>

- A knowledge base (KB)  $C_2$  containing objects corresponding to the entities mentioned in  $C_1$ .

These two parts are integrated by linking the annotated entities in  $C_1$  to the corresponding objects in  $C_2$ , as Figure 2 illustrates.



**Fig. 1.** The Integrated Corpus

For the first version of our corpus, we defined a set of 40 documents with approximately 100 words. These documents were excerpts from Wikipedia<sup>5</sup> biographies of various politicians, actors, scientists, bands, fictional and non-fictional characters, etc. We compiled the corpus with the aim of including challenging problems for both the NER and the object consolidation task, such as different forms of the same name (e.g. “Bill Clinton”, “Clinton”, “Billy”), potentially ambiguous tokens (e.g. “Hope”: location/verb) and pseudonyms (e.g. “Ringo Starr”, “Richard Starkey”). The corpus was then annotated by one human annotator, currently only with respect to three different annotation types: PERSON, LOCATION and JOBTITLE.

In order to allow the integration of the textual corpus and the KB, the latter would have to contain the same entities as mentioned in the text. Of the 205 PERSON annotations in the textual corpus, 95 referred to individual entities. For each of these entities, we included a corresponding entity in the KB. Within the Geco project, we were working with FOAF<sup>6</sup> representations of people. For this reason, we chose to build a KB of `foaf:Person` instances.

Having completed both parts of the corpus, they had to be tied together. This was achieved by referencing the `Person` instances in the knowledge base from the annotations in the textual corpus. In FOAF, the assumption is made that each person can be uniquely identified by her email address. We therefore used email addresses (both real and made-up) as the referencing scheme. Once both parts of the corpus had been related in that way, the Integrated Corpus was complete.

### 3 Future Work

In this paper, we proposed a novel kind of evaluation corpus, which we called an **Integrated Corpus for the Evaluation of Named Entity Recognition and Object Consolidation**. It can be used for both the evaluation of NER systems and systems trying to solve object consolidation problems. We are aware of the fact that future versions of the textual part of our corpus will have to be extended in both size and depth. We will have to extend the size of the corpus, its scope and the number of annotation types. Another important task for a future version of our corpus is the development of a suitable kind of evaluation metrics. The conventional recall, precision and F-measure metrics could be applied individually to the textual part of the corpus and the linking between the annotations and the instances in the knowledge base. However, it would be desirable to provide a combined measure in order to rate the overall performance of a system with respect to our corpus.

<sup>5</sup> see <http://en.wikipedia.org>

<sup>6</sup> see <http://xmlns.org/foaf/0.1>

## Bibliography

- [1] Popov, B., Kiryakov, A., Manov, D., Ognyanoff, D., Goranov, M.: Kim - semantic annotation platform. *Lecture Notes in Computer Science* **124** (2003) 834–849
- [2] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Sementag and seeker: bootstrapping the semantic web via automated semantic annotation. In: *Proceedings of the twelfth international conference on World Wide Web*, ACM Press (2003) 178–186
- [3] Möller, K.: Geco - using human language technology to enhance semantic web browsing. In: *Proceedings of the Faculty of Engineering Research Day 2004*, National University of Ireland, Galway. (2004)