

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

UPGRADE is the anchor point for UPENET (UPGRADE European Network), the network of CEPIs member societies' publications, that currently includes the following ones:

- **Mondo Digitale**, digital journal from the Italian CEPIs society AICA
- **Novática**, journal from the Spanish CEPIs society ATI
- **OCG Journal**, journal from the Austrian CEPIs society OCG
- **Pliroforiki**, journal from the Cyprus CEPIs society CCS
- **Pro Dialog**, journal from the Polish CEPIs society PTI-PIPS

Publisher

UPGRADE is published on behalf of CEPIs (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** (<http://www.ati.es/novatica/>), journal of the Spanish CEPIs society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**, and in Italian (summary, abstracts and some articles online) by the Italian CEPIs society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*, <http://www.alsi.it/>) and the Italian IT portal *Tecnoteca* (<http://www.tecnoteca.it/>)

UPGRADE was created in October 2000 by CEPIs and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

Editorial Team

Chief Editor: Rafael Fernández Calvo, Spain, rfcvalvo@ati.es
 Associate Editors:
 François Louis Nicolet, Switzerland, nicolet@acm.org
 Roberto Carniel, Italy, carniel@dgt.uniud.it
 Zakaria Maamar, Arab Emirates, Zakaria.Maamar@zu.ac.ae
 Soraya Kouadri Mostéfaoui, Switzerland, soraya.kouadrimostefaoui@unifr.ch

Editorial Board

Prof. Wolfried Stucky, Former President of CEPIs
 Prof. Nello Scarabottolo, CEPIs Vice President
 Fernando Piera Gómez and
 Rafael Fernández Calvo, ATI (Spain)
 François Louis Nicolet, SI (Switzerland)
 Roberto Carniel, ALSI – Tecnoteca (Italy)

UPENET Advisory Board

Franco Filippazzi (Mondo Digitale, Italy)
 Rafael Fernández Calvo (Novática, Spain)
 Veith Risak (OCG Journal, Austria)
 Panicos Masouras (Pliroforiki, Cyprus)
 Andrzej Marciniak (Pro Dialog, Poland)

English Editors: Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by Antonio Crespo Foix, © ATI 2005

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Rafael Fernández Calvo rfcvalvo@ati.es

Advertising correspondence: novatica@ati.es

UPGRADE Newsletter available at

<http://www.upgrade-cepis.org/pages/editinfo.html#newsletter>

Copyright

© Novática 2005 (for the monograph and the cover page)

© CEPIs 2005 (for the sections MOSAIC and UPENET)

All rights reserved. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (February 2006):

Key Success Factors in Software Engineering
 (The full schedule of UPGRADE is available at our website)

Monograph: The Semantic Web (published jointly with Novática*)

Guest Editors: *Luis Sánchez-Fernández, Michael Sintek, and Stefan Decker*

- 2 Presentation. The Semantic Web or The Next Web Wave – *Luis Sánchez-Fernández, Michael Sintek, and Stefan Decker*
- 5 The Semantic Web: Fundamentals and A Brief State-of-the-Art – *Luis Sánchez-Fernández and Norberto Fernández-García*
- 12 Leveraging Metadata Creation by Annotation for The Semantic Web – *Siegfried Handschuh*
- 19 The Quest for Information Retrieval on The Semantic Web – *David Vallet-Weadon, Miriam Fernández-Sánchez, and Pablo Castells-Azpilicuenta*
- 24 Functional RuleML: From Horn Logic with Equality to Lambda Calculus – *Harold Boley*
- 30 Towards Semantic Desktop Wikis – *Malte Kiesel and Leo Sauerermann*
- 35 Towards Semantically-Interlinked Online Communities – *Uldis Bojars, John G. Breslin, Andreas Harth, and Stefan Decker*
- 41 A Semantic Search Engine for the International Relation Sector – *Luis Rodrigo-Aguado, V. Richard Benjamins, Jesús Contreras-Cino, Diego-Javier Patón-Villahermosa, David Navarro-Arno, Robert Salla-Figuerol, Mercedes Blázquez-Cívico, Pilar Tena-García, and Isabel Martos-Laborde*
- 48 Semantic Search in Digital Image Archives: A Case Study – *Julio Villena-Román, José-Carlos González-Cristóbal, Cristina Moreno-García, and José- Luis Martínez-Fernández.*
- 55 Configuring e-Government Services Using Ontologies – *Dimitris Apostolou, Ljiljana Stojanovic, Tomás Pariente-Lobo, Joan Battle-Montserrat, and Andreas E. Papadakis*

UPENET (UPGRADE European Network)

- 63 From **Novática** (ATI, Spain)
 ICT for Education
 An Initiative for Educational Modernization: The Ponte dos Brozos Project – *Simón Neira-Dueñas and Felipe Gómez-Pallete Rivas*
- 71 From **Pro Dialog** (PIPS, Poland)
 ICT for Education
 On The Superiority of Internet-Based Mass Enrolment to High Schools over Traditional – *Andrzej P. Urbanski*

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIs society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>, and in Italian (online edition only, containing summary, abstracts, and some articles) by the Italian CEPIs society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*) and the Italian IT portal *Tecnoteca* at <http://www.tecnoteca.it/>.

Leveraging Metadata Creation by Annotation for The Semantic Web

Siegfried Handschuh

The success of the Semantic Web crucially depends on the easy creation of ontology-based metadata by semantic annotation. We provide a framework, CREAM, that allows for the creation of semantic metadata about static and dynamic Web pages, i.e. for semantic annotation of the Shallow and the Deep Web. CREAM supports the manual and the semi-automatic annotation of static Web pages, the authoring of new web pages with the simultaneous creation of metadata, and the deep annotation of Web pages defined dynamically by database queries.

Keywords: Metadata, Semantic Annotation, Semantic Web.

1 Introduction

Like all truly great ideas, Tim Berners-Lee's principle idea of the *Semantic Web* may be easily summarized: when computers not only retrieve, but also understand what data is available on the Web, we will have a new kind of Web and new types of intelligent applications in the Web. In the foreseeable future, however, machines will be too dumb to understand what people have put on the Web. Therefore, let us put *computer-understandable data* next to human-understandable data. Then, the computers will be smarter.

In order to make the vision of the Semantic Web come true, we need a number of building blocks, some of them elaborated on in recent writings [17][9][21]. For instance, we need standardized languages to describe semantic data, i.e. data that is as computer-understandable as it is semantically self-describing, and we need programmes and protocols to actually exchange and understand semantic data. As a priority, however, we need semantic data.

This paper is about providing semantic data, a process often referred to as "Semantic Annotation" because it frequently involves the embellishment of existing data, e.g. plain text, that is only understandable for the human, with semantic metadata that describes, e.g., the text. Understandably, Semantic Annotation is now one of the core challenges for building the Semantic Web.

Since Semantic Annotation is the key notion of this paper we shall define it in more detail.

Definition: *the term Semantic Annotation describes a process as well as the outcome of the process*¹. Hence it describes: i) the process of addition of semantic data or metadata to the content given an agreed ontology and ii) it describes the semantic data or metadata itself as a result of this process.

The Semantic Web supports its users to find accurate information, to combine related pieces of information into an overarching picture and to compose new applications without programming knowledge.

To achieve these objectives not only human readers have to understand the content of on a web page, software agents also must be able to interpret existing information. This is only possible when the relevant information is represented in a declarative and semantically precise way and when it is thus understandable for the computer. This need creates the necessity to provide semantically accurate, ontology-based metadata. We describe how the problem is tackled by means of our annotation framework, CREAM (CREating Metadata for the Semantic Web). CREAM comprises methods for:

Manual annotation: the transformation of existing syntactic resources (*viz.* textual documents) into interlinked knowledge structures that represent relevant underlying information.

Authoring of documents: in addition to the annotation of existing documents the authoring mode lets authors create metadata - almost for free - while putting together the content of a document [12].

Semi-automatic annotation: efficient semi-automatic annotation based on information extraction that is trained to handle structurally and/or linguistically similar documents [13]. Another approach for semi-automatic annotation is the self-annotating Web. The principle idea of the self-annotating Web is that it uses globally available Web data and structure to semantically annotate - or at least facilitate annotation of - local resources [3].

Siegfried Handschuh is a Research Fellow at the National University of Ireland, Galway, and working at the Digital Enterprise Research Institute (DERI) in the Semantic Web Cluster. Previously he worked at the FZI (*Forschungszentrum Informatik*) Karlsruhe, Germany, for the Ontoprise team on the HALO 2 project - research effort towards the development of Digital Aristotle - and before that at the Institute AIFB (University of Karlsruhe) in the OntoAgents project in the DARPA DAML (Defense Advanced Research Projects Agency - DARPA Agent Markup Language) program. He has obtained his degree in Information Science from the University of Constance, Germany, and his PhD from the University of Karlsruhe. His current research interests include Semantic Desktop, Annotations in the Semantic Web and Knowledge Acquisition. He has chaired several workshops on Semantic Annotation and is co-editor of the book "Annotation for the Semantic Web". <ha@aifb.uni-karlsruhe.de>

¹ Cf. with term "drawing".

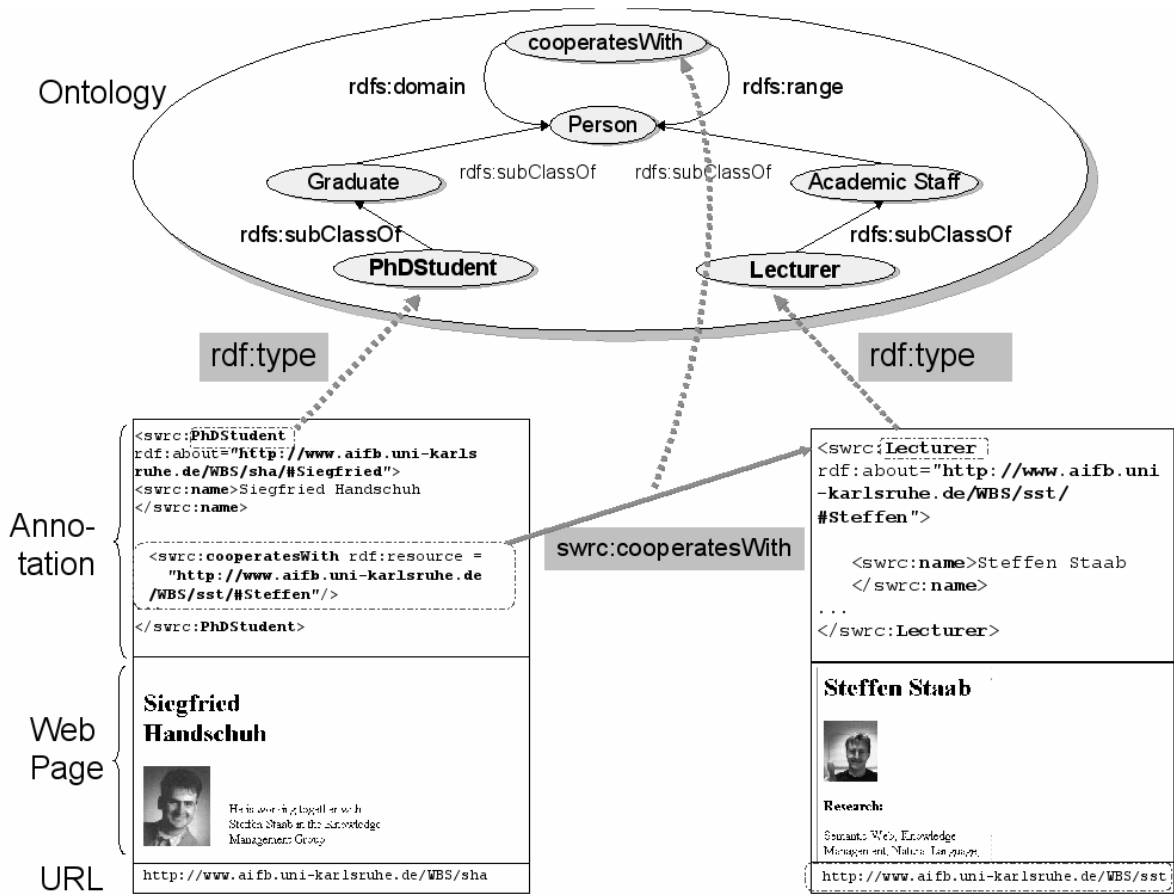


Figure 1: Annotation Example.

Deep annotation: deep annotation results in a semantic mapping to the underlying database if the database owner cooperates in the Semantic Web and allows for direct access to the database [14].

In the following we wrap up [15] and afore mentioned contributions.

For a more concise description, we first define our terminology and give an example of the kind of metadata the generation of which we support by CREAM (Section 2). In Section 3 we elaborate the requirements for an annotation framework and in 4 we derive the design of it. Eventually, we will discuss related works and conclude.

2 Relational Metadata

We elaborate the terminology here because many of the terms that are used with regard to metadata creation tools carry several, ambiguous connotations that imply conceptually important differences:

An **ontology** is a formal, explicit specification of a shared conceptualization of a domain of interest [11]. In our case, an ontology is defined in RDF(S) (Resource Description

Framework Schema) or OWL (Web Ontology Language). Hence, an ontology is constituted by statements expressing definitions of OWL classes - RDF(S) resources, respectively - and properties ([23][2]).

An **annotation** in our context is a set of instantiations attached to an HTML (HyperText Markup Language) document. We distinguish (i) instantiations of OWL classes, (ii) instantiated properties from one class instance to a datatype instance - henceforth called attribute instance (of the class instance), and (iii) instantiated properties from one class instance to another class instance - henceforth called relationship instance.

Class instances should have unique URIs (Uniform Resource Identifier)², e.g. like 'http://www.uni.de/WBS/sst/#Steffen'. Note, that uniqueness of identifier is an issue of *proper referencing*, which is elaborated in Section 3. Class instances frequently come with attribute instances, such as a human-readable label like 'Steffen'.

Metadata are data about data. In our context the annotations are metadata about the HTML documents. We use the term **relational metadata** to denote the annotations that contain relationship instances.

Often, the term "annotation" is used to mean something like "private or shared note", "comment" or "Dublin Core metadata". This alternative meaning of annotation may be emulated in our approach by modeling these notes with at-

²For a clarification of the relationship between URIs, URLs, and URNs (Uniform Resource Name), see <http://www.w3.org/TR/uri-clarification/>.

tribute instances. For instance, a comment note "I like this paper" would be related to the URL (Universal Resource Locator) of the paper via an attribute instance 'hasComment'.

In contrast, relational metadata also contain statements like 'Siegfried cooperates with Steffen', i.e. relational metadata contain relationships between class instances rather than only textual notes.

Figure 1 illustrates our use of the terms "ontology", "annotation" and "relational metadata". It depicts some part of the SWRC (Semantic Web Research Community, <<http://ontobroker.semanticweb.org/ontos/swrc.html>>) ontology. Furthermore it shows two homepages, viz. pages about Siegfried and Steffen (<<http://www.uni.de/WBS/sha>> and <<http://www.uni.de/WBS/sst>>, respectively) with annotations given in an XML (eXtensible Markup Language) serialization of RDF facts. For the two persons there are instances denoted by corresponding URIs (<<http://www.uni.de/WBS/sst/#Steffen>> and <<http://www.uni.de/WBS/sha/#Siegfried>>). The swrc: name of <<http://www.uni.de/WBS/sha/#Siegfried>> is "Siegfried Handschuh". In addition, there is a relationship instance between the two persons, viz. they cooperate. This cooperation information 'spans' the two pages.

3 Requirements

CREAM is an annotation and authoring framework suited for the easy and comfortable creation of relational metadata. OntoMat-Annotizer (OntoMat for short) is its concrete implementation.

Given the problems with syntax, semantics and pragmatics in earlier experiences, e.g. KA2 [1], we list here a set of requirements. Thereby, the principal requirements apply for *a-posteriori* annotation as well as for the *integration of web page authoring with metadata creation* as follows:

Consistency: semantic structures should adhere to a given ontology in order to allow for better sharing of knowledge. For example, it should be avoided that annotators use an attribute instance, whereas the ontology requires a concept instance.

Proper Reference: identifiers of instances, e.g. of persons, institutes or companies, should be unique. In fact, in most real-world situations the same object will be given many URIs, since people create them independently. For instance, the metadata generated in the KA2 case study contained three different identifiers for our colleague Dieter Fensel. Thus, knowledge about him could not be grasped with a straightforward query. To remedy this problem, there are technical ('smushing') and logical (e.g. OWL:sameAs) solutions.

Avoid Redundancy: decentralized knowledge provisioning should be possible. However, when annotators collaborate, it should be possible for them to identify

(parts of) sources that have already been annotated and to reuse previously captured knowledge in order to avoid laborious redundant annotations.

Relational Metadata: like HTML information, which is spread on the Web, but related by HTML links, knowledge markup may be distributed, but it should be semantically related. Current annotation tools tend to generate template-like metadata, which is hardly connected, if at all. For example, annotation environments often support Dublin Core [6][7], providing means to state, e.g., the name of authors of a document, but not their Ids³. Thus, the only possibility to query for all publications of a certain person requires the querying for some attribute like *fullname* - which is very unsatisfying for frequent names like "John Smith".

Dynamic Documents: a large percentage of the web pages are not static documents. For dynamic web pages (e.g. ones that are generated from a database) it does not seem to be useful to annotate every single page. Rather one wants to "annotate the database" in order to reuse it for its own Semantic Web purpose.

Maintenance: knowledge markup needs to be maintained. An annotation tool should support the maintenance task.

Ease of Use: it is obvious that an annotation environment should be easy to use in order to be really useful. However, this objective is not easily achieved, because metadata creation involves intricate navigation of semantic structures, e.g. taxonomies, properties and concepts.

Efficiency: the effort for the production of metadata is an important restraining threshold. The more efficiently a tool supports metadata creation, the more metadata users tend to produce. This requirement is related to the ease of use. It also depends on the automation of the metadata creation process, e.g. on the preprocessing of the document.

Multiple Ontologies: HTML documents in the semantic web may contain information that is related to different ontologies. Therefore the annotation framework should cater for concurrent annotations with multiple ontologies.

Our framework CREAM that is presented here, targets a comprehensive solution for metadata creation during web page authoring and *a-posteriori* annotation. The objective is pursued by combining advanced mechanisms for inferencing, fact crawling, document management, meta ontology definitions, metadata re-recognition, content generation, and information extraction. These components are explained in the subsequent sections.

4 Design of CREAM

The difficulties sketched before directly feed into the design rationale of CREAM. The design rationale links the requirements with the CREAM modules. This results in an N:M mapping (neither functional nor injective). A tabular overview of the matrix can be found in [12].

Document Editor: the document editor may be conceptually - though not practically - distinguished into a viewing component and the component for generating content: The *document viewer* visualizes the document contents.

³ In the web context one typically uses the term URI (Uniform Resource Identifier) to speak of "unique identifier".

The annotator may easily provide new metadata by selecting pieces of text and aligning it with parts of the ontology. The document viewer should support various formats (HTML, PDF - Portable Document Format, XML, etc.). For some formats the following component for content generation may not be available.

The document viewer highlights the existing semantic annotation and server-side markup of the web page. It distinguishes visually between semantic annotation and markup that describes the information structure of an underlying database.

The editor also allows the conventional authoring of documents, viz. the *content generation*. In addition, instances already available may be dragged from a visualization of the content of the annotation inference server and dropped into the document. Thereby, some piece of text and/or a link is produced taking into account the information from the meta ontology (cf. module meta ontology).

The newly generated content is already annotated and the meta ontology guides the construction of further information, e.g. further XPointers [5][10] are attached to instances.

Ontology Guidance and Fact Browser: the framework needs guidance from the ontology. In order to allow for sharing of knowledge, newly created annotations must be consistent with a community's ontology. If metadata creators instantiate arbitrary classes and properties the semantics of these properties remains void. Of course the frame-

work must be able to adapt to multiple ontologies in order to reflect different foci (or focuses) of the metadata creators. In the case of concurrent annotation with multiple ontologies there is an ontology guidance/fact browser for each ontology.

Crawler: the creation of relational metadata must take place *within* the Semantic Web. During metadata creation, subjects must be aware of which entities already exist in their part of the Semantic Web. This is only possible if a crawler makes relevant entities immediately available.

Annotation Inference Server: relational metadata, proper reference and avoidance of redundant annotation require querying for instances, i.e. querying whether and which instances exist. For this purpose as well as for checking of consistency, we provide an annotation inference server. The annotation inference server reasons on crawled and newly created instances and on the ontology. It also serves the ontological guidance and fact browser, because it allows to query for existing classes, instances and properties.

Meta Ontology: the purpose of the meta ontology is the separation of ontology design and use. It is needed to describe how classes, attributes and relationships from the domain ontology should be used by the CREAM environment. Thus, the ontology describes how the semantic data should look like and the meta ontology connected to the ontology describes how the ontology is used by the annotation environment to actually create semantic data. It is specifically explained in [12].

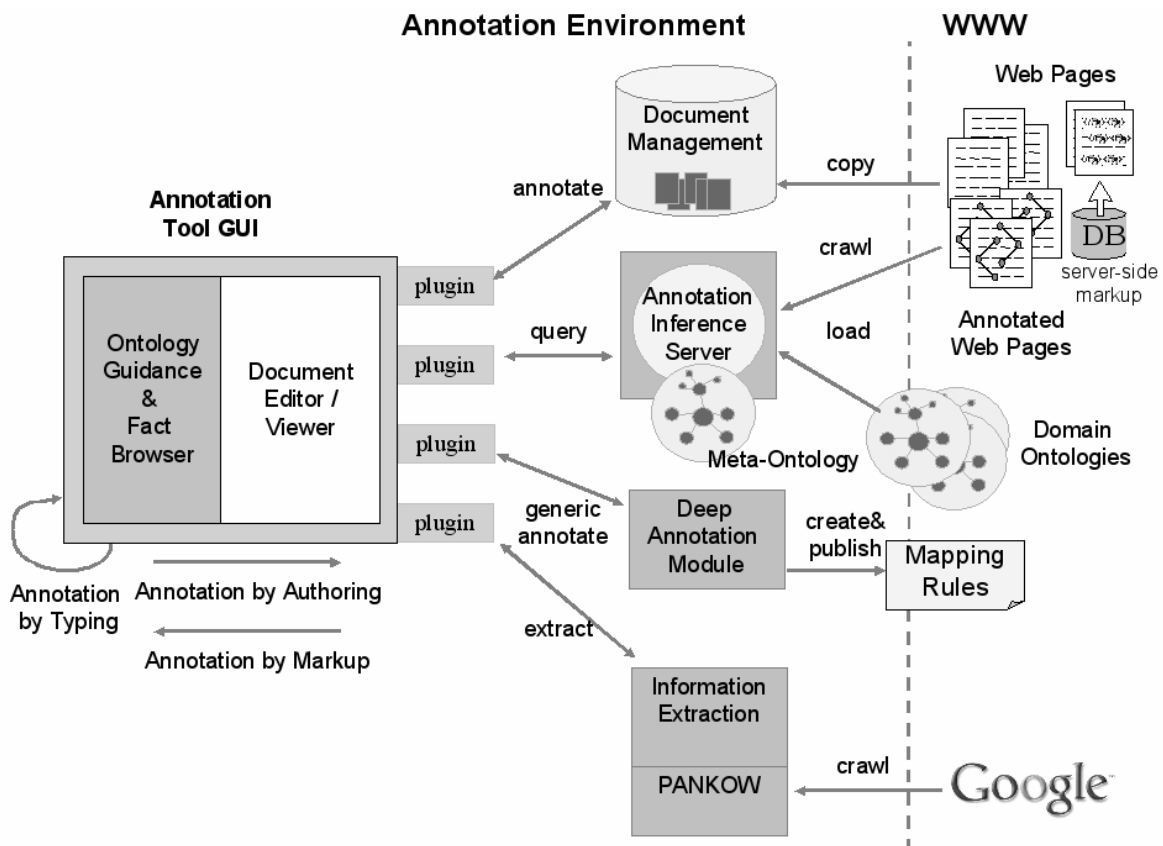


Figure 2: Architecture of CREAM.

Deep Annotation Module: this module enables the deep annotation scenario. It manages the generation of mapping rules between the database and the client ontology. For this purpose, it combines the generic annotation stored in the annotation inference server and the server-side markup provided with the content. [14]. On demand it publishes the mapping rules derived from the generic annotations.

Document Management: considering the dynamics of HTML pages on the web, it is desirable to store foreign web pages one has annotated together with their annotations. Foreign documents for which modification is not possible may be remotely annotated by using XPointer as a addressing mechanism.

Metadata Re-recognition & Information Extraction: even with sophisticated tools it is laborious to provide semantic annotations. A major goal thus is semi-automatic metadata creation taking advantage of information extraction techniques to propose annotations to metadata creators and, thus, to facilitate the metadata creation task. Concerning our environment we envisage three major techniques:

Firstly, metadata re-recognition compares existing metadata literals with newly typed or existing text. Thus, the mentioning of the name "Siegfried Handschuh" in the document triggers the proposal that URI, <http://www.aifb.uni.de/WBS/sha/#Siegfried>, is co-referenced at this point. Secondly, "Wrappers" may be learned from given markup in order to automatically annotate similarly structured pages. Thirdly, message extraction systems may be used to recognize named entities, propose co-reference, and extract some relationship from texts (cf., e.g., [22][24]).

This component has been realized by using the Amilcare information extraction system (cf. [13], <http://www.dcs.shef.ac.uk/~fabio/Amilcare.htm>), but it is not yet available in the download version of OntoMat.

Besides the requirements that constitute single modules, one may identify functions that cross module boundaries, such as storage and replication. CREAM supports two different ways of *storage*. The annotations will be stored inside the document that is in the document management component. Alternatively or simultaneously it is also possible to store them in the annotation inference server. We provide a simple *replication* mechanism by crawling annotations into our annotation inference server. Then inferencing can be used to rule out formal inconsistencies.

4.1 Architecture of CREAM

The architecture of CREAM is depicted in Figure 2. The Design of the CREAM framework pursues the idea to be flexible and open. Therefore, OntoMat, the implementation of the framework, comprises a plug-in structure, which is flexible with regard to adding or replacing modules.

The core OntoMat (see screenshot in Figure 3), which is downloadable, consists of an Ontology Guidance and Fact browser (left hand side), a document viewer/editor (right hand side), and a internal memory data-structure for the ontology and metadata. However, one only gets the full-fledged semantic capabilities (e.g. datalog reasoning or

subsumption reasoning) when one uses a plug-in connection to a corresponding annotation inference server.

5 Comparison with Related Work

In the following we mention only briefly some of the related work, for a more elaborate comparison the interested reader may turn to [15]. We know of three major systems that use knowledge markup intensively in the Semantic Web, viz. SHOE [16], Ontobroker [4], and WebKB [20]. All three rely on markup in HTML pages. They all started with providing manual markup by editors. However, our experiences (cf. [8]) have shown that text-editing knowledge markup yields extremely poor results, viz. syntactic mistakes and improper references.

The approaches from this line of research that are closest to CREAM are the SHOE Knowledge Annotator <http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html>, WebKB [20], and the MnM [25] annotation tool.

Annotea (Amaya) [19][18] is a tool that shares the idea of creating a kind of user comment about Web pages. The term "annotation" in these frameworks is understood as a remark to an existing document. But Annotea actually goes one step further. It allows reliance on an RDF schema as a kind of template that is filled in by the annotator. For instance, Annotea users may use a schema for Dublin Core and fill the author-slot of a particular document with a name. The user may also decide to use complex RDF descriptions instead of simple strings for filling in such a template.

6 Conclusion

The main contributions made by this work is the **design of a comprehensive and pioneering annotation framework** that reduces the complexity of Semantic Annotation for the annotator. The framework employs a comprehensive set of modules including inference services, crawler, document management system, ontology guidance/fact browser, and document editors/viewers. Process issues pertaining to the annotation/authoring task are modularized from content descriptions by a meta ontology. The framework has been **prototypically implemented** in the open source project OntoMat, <http://projects.semwebcentral.org/projects/ontomat/>, hosted by the DARPA DAML (Defense Advanced Research Projects Agency - DARPA Agent Markup Language) program. OntoMat is the reference implementation of the CREAM framework. It is Java-based and provides a plug-in interface for extensions for further applications. It has been used in several cases, e.g. the annotation of paper abstracts for the International Conferences on Semantic Web (ISWC 2002, 2003, 2004) by some of the authors. Also it is in use on class room machines in an obligatory Semantic Web course, <http://nb.vse.cz/~svatek/modz.htm>, for informatics students in Prague, on which some 250 people enrol every year.

The annotation framework comprised methods especially for manual annotation, authoring of documents, semi-automatic annotation and deep annotation.

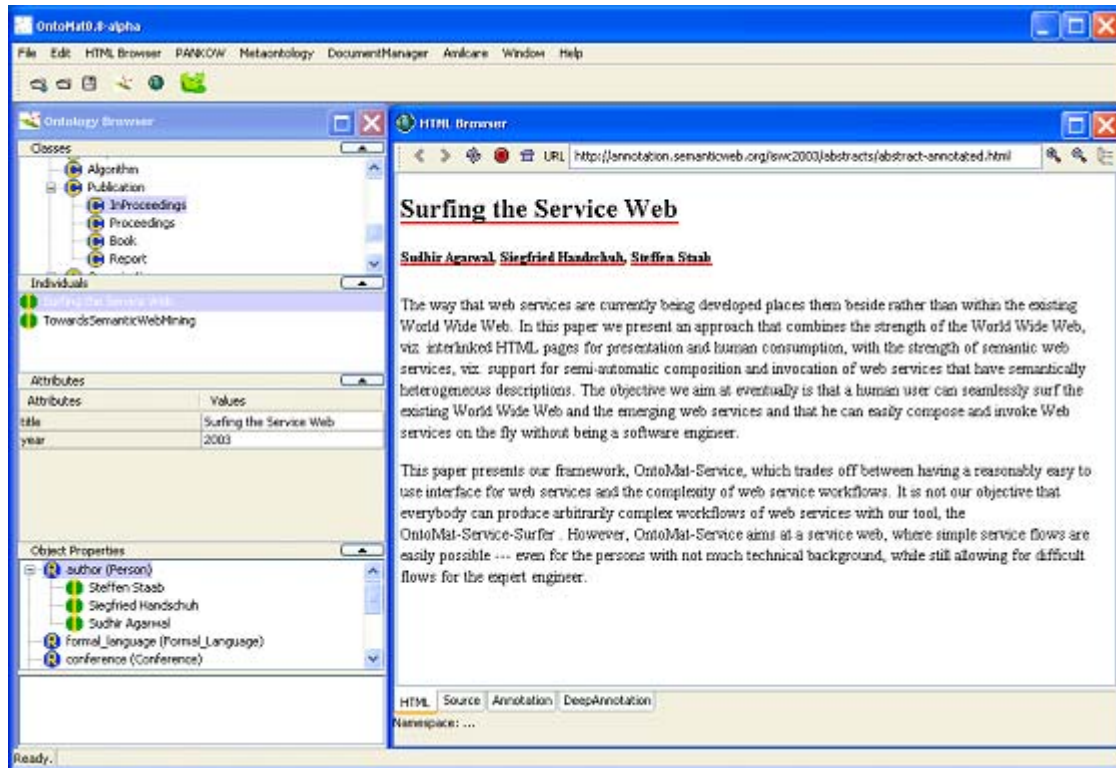


Figure 3: Screenshot of OntoMat.

Insights

In addition, this research provides several additional insights into Semantic Annotation, which are described in the following. Semantic Metadata are, simply expressed, facts that are related to a domain ontology. Though this may appear trivial at first, however this easily conflicts with several other requirements. We also need a *meta ontology* describing how the domain ontology should be used by the annotation framework. Furthermore, there is the requirement for remote storage of annotation, which leads to the need for a robust referencing scheme, viz. *XPointer*. Also, there is the need for the provision of *metametadata*, e.g. author, date, time and location of annotation. In addition, different requirements exist for different *semantics* of Semantic Annotation as well as the need to express different aspects of the content in metadata viz. a *layering* of the annotation (e.g. Structural Annotation, Lexical Annotation, Semantic Annotation).

Automatization is vital to ease the knowledge acquisition bottleneck. To achieve this, the integration of knowledge extraction technologies into the annotation environment has been undertaken. This is used to semi-automatically identify entities in text that are instances of a particular class and relations between the classes. As the evaluation in [15] showed, HCI implications are also important here, so that a semi-automated tool can be used effectively by Web users without expertise in natural language processing methods. Annotation is a potential knowledge acquisition bottleneck as discussed above. To ease the constriction, annotation has to be carried out by people who are not

specialist annotators. To facilitate the annotation task is especially important for the success of the Semantic Web. The Annotation interfaces must, therefore, bridge the gap between formal descriptions of knowledge and Web users understanding their domains of interest. A good approach is therefore a semantic authoring environment, so that the environment in which users annotate documents is the same as the one in which they create, read and edit them.

Open Questions

The general problem of metadata creation remains interesting. In the following the open questions that are not answered by our work are identified:

Firstly, the question of **scalability** to more and larger dimensions. Like "*what happens if there are 100,000 people known in your annotation inference server?*". Even for the evaluation we had to prune the ontology in order to make it feasible for the annotation task. Secondly, Semantic Annotation takes place within the Semantic Web. For the proper creation of relational metadata we need **unique identifiers** of persons, institutes or companies. While crawling of existing metadata helps to reduce this problem it is not solved and possibly may never be. Thirdly, we are still in the early stages with respect to providing **methodological guidelines** for the purposes of Semantic Annotation. Fourthly, probably the most important for the Semantic Web. How to create **incentives** for annotation?

Summary

Documents created by Semantic Annotation bring the

advantages of semantic search and interoperability. These benefits, however, come at the cost of an increased authoring effort. In this paper we have, therefore, presented a comprehensive framework which support users in dealing with the documents, the ontologies and the annotations that link documents to ontologies.

Future research challenges include further improvements to automatic annotation components, such as relation extraction, and developing support systems for ontology evolution. There are also important human computer interaction challenges inherent in building integrated systems of this complexity.

References

- [1] R. Benjamins, D. Fensel, and S. Decker. KA2: Building Ontologies for the Internet: A Midterm Report. *International Journal of Human Computer Studies*, 51(3):687-713, 1999.
- [2] D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, W3C, Feb. 2004. W3C Working Draft, <<http://www.w3.org/TR/rdf-schema/>>.
- [3] P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web, in Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills (eds.), *Proc. of the WWW2004*, pages 462-471, New-York, USA, May 2004. ACM.
- [4] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information, in R. Meersman et al. (eds.), *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351-369. Kluwer Academic Publisher, 1999.
- [5] S. DeRose, E. Maler, and R. Daniel. XML Pointer Language (XPointer). Technical report, W3C, 2001. Working Draft 16 August 2002.
- [6] Dublin core metadata initiative, April 2001, <<http://purl.oclc.org/dc/>>.
- [7] Dublin Core Metadata Template, 2001. http://www.ub2.lu.se/metadata/DC_creator.html.
- [8] M. Erdmann, A. Maedche, H.-P. Schnurr, and S. Staab. From Manual to Semi-Automatic Semantic Annotation: About Ontology-Based Text Annotation Tools, in P. Buitelaar & K. Hasida (eds). *Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg, August 2000.
- [9] D. Fensel, K. P. Sycara, and J. Mylopoulos, editors. *ISWC-2003 - Proceedings of the Second International Semantic Web Conference*, LNCS 2870. Springer, 2003.
- [10] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall. Conceptual Open Hypermedia = The Semantic Web?, in S. Staab, S. Decker, D. Fensel, and A. Sheth (eds.), *The Second International Workshop on the Semantic Web*, CEUR Proceedings, Volume 40, <<http://www.ceur-ws.org>>, pages 44-50, Hong Kong, May 2001.
- [11] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 6(2):199-221, 1993.
- [12] S. Handschuh and S. Staab. Authoring and annotation of web pages in cream, in *Proc. of the 11th WWW 2002*, Honolulu, Hawaii, May 7-11, 2002, pages 462-473. ACM Press, 2002.
- [13] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREATION of Metadata, in *Proc. of EKAW02, LNCS/LNAI 2473*, pages 358-372, Sigüenza, Spain, October 2002. Springer.
- [14] S. Handschuh, S. Staab, and R. Volz. On deep annotation, in *Proc. of the WWW2003*, Budapest, HUNGARY, May 2003.
- [15] Siegfried Handschuh. *Creating Ontology-based Metadata by Annotation for the Semantic Web*. Dissertation, Universität Fridericiana zu Karlsruhe, 2005.
- [16] J. Heflin and J. Hendler. Dynamic Ontologies on the Web, in *AAAI-2000 - Proceedings of the National Conference on Artificial Intelligence*. Austin, TX, USA, August 2000, 2000.
- [17] J. Hendler and I. Horrocks (eds.). *ISWC-2002 - Proceedings of the First International Semantic Web Conference*, LNCS 2342. Springer, 2002.
- [18] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations, in *Proc. of the Tenth International World Wide Web Conference, WWW 10*, Hong Kong, China, May 1-5, 2001, pages 623-632. ACM Press, 2001.
- [19] Marja-Riitta Koivunen and Ralph R. Swick. Collaboration through Annotations in the Semantic Web. In this book, 2003.
- [20] P. Martin and P. Eklund. Embedding Knowledge in Web Documents, in *Proceedings of the 8th Int. World Wide Web Conf. (WWW'8)*, Toronto, May 1999, pages 1403-1419. Elsevier Science B.V., 1999.
- [21] S. A. McIlraith, D. Plexousakis, and F. van Harmelen (eds.). *ISWC-2004 - Proceedings of the Third International Semantic Web Conference*, LNCS 3298. Springer, 2004.
- [22] MUC-7 - *Proc. of the 7th Message Understanding Conference*, 1998, <<http://www.muc.saic.com/>>.
- [23] OWL Web Ontology Language Reference, <<http://www.w3.org/TR/owl-ref/>>, 2004.
- [24] M. Vargas-Vera, E. Motta, J. Domingue, S. Buckingham Shum, and M. Lanzoni. Knowledge Extraction by using an Ontology-based Annotation Tool, in *Proc. of the Knowledge Markup and Semantic Annotation Workshop 2001 (at K-CAP 2001)*, pages 5-12, Victoria, BC, Canada, October 2001.
- [25] M. Vargas-Vera, E. Motta, J. Domingue, S. Buckingham Shum, and M. Lanzoni. Knowledge Extraction by Using an Ontology-Based Annotation Tool, in *Proceedings of the Knowledge Markup and Semantic Annotation Workshop 2001 (at K-CAP 2001)*, pages 5-12, Victoria, BC, Canada, October 2001.