Available online at www.sciencedirect.com
SCIENCE DIRECT®
Web Semantics: Science, Services and Agents on the World Wide Web xxx (2005) xxx–xxx
JOURNAL OF Web Semantics
www.elsevier.com/locate/websem

Semantic annotation for knowledge management: Requirements and a survey of the state of the art

Victoria Uren [a,*], Philipp Cimiano [b], José Iria [c], Siegfried Handschuh [d], Maria Vargas-Vera [a], Enrico Motta [a], Fabio Ciravegna [c]

[a] *Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK*
[b] *Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany*
[c] *Department of Computer Science, University of Sheffield, Regent Court, 21 1Portobello Street, Sheffield S1 4DP, UK*
[d] *FZI Forschungszentrum Informatik, Haid-und-Neu-Strasse 10-14, 76131 Karlsruhe, Germany*

Received 24 January 2005; accepted 18 October 2005

Abstract

While much of a company's knowledge can be found in text repositories, current content management systems have limited capabilities for structuring and interpreting documents. In the emerging Semantic Web, search, interpretation and aggregation can be addressed by ontology-based semantic mark-up. In this paper, we examine semantic annotation, identify a number of requirements, and review the current generation of semantic annotation systems. This analysis shows that, while there is still some way to go before semantic annotation tools will be able to address fully all the knowledge management needs, research in the area is active and making good progress.

Keywords: Semantic annotation; Knowledge management; Automation

1. Introduction

Does Semantic Web technology matter for knowledge management (KM)? We believe that it does because KM often centers on documents and the business processes that build on them. Documents provide a rich resource describing what an organization knows and account for 80–85% of the information stored by many companies. Indeed, for some professions documents are effectively the product they sell. Examples of these "product" documents include contracts, consultancy reports and consumer surveys. KM systems for handling this unstructured material are a large and growing sector of the software industry. IDC expects that content management and retrieval software spending will outpace the overall software market by 2007. They estimate the market at $6.46 billion in 2004 and a $9.72 billion by 2006 [1].

The Semantic Web envisages technologies, which can make possible the generation of the kind of "intelligent" documents imagined 10 years ago [2]. We define an intelligent document as a document which "knows about" its own content in order that automated processes can "know what to do" with it. Knowledge about documents has traditionally been managed through the use of metadata, which can concern the world around the document, e.g. the author, and often at least part of the content, e.g. keywords. The Semantic Web proposes annotating document content using semantic information from domain ontologies [3]. The result is Web pages with machine interpretable mark-up that provide the source material with which agents and Semantic Web services operate. The goal is to create annotations with well-defined semantics, however those semantics may be defined. The Semantic Web relies on a model-theoretic definition of meaning, but other types of semantics could be thought of [4]. In any case, for the sake of interoperability, a well-defined semantics is a must to ensure that annotator and annotation consumer actually share meaning. A key contribution of the Semantic Web is therefore to provide a set of worldwide standards. These open the possibility of operating with heterogenous resources by providing a bridge of common syntax, methods, etc.

* Corresponding author. Tel.: +44 1908 858516; fax: +44 1908 653169.
E-mail addresses: v.s.uren@open.ac.uk (V. Uren), cimiano@aifb.uni-karlsruhe.de (P. Cimiano), j.iria@dcs.shef.ac.uk (J. Iria), handschuh@acm.org (S. Handschuh), m.vargas-vera@open.ac.uk (M. Vargas-Vera), e.motta@open.ac.uk (E. Motta), f.ciravegna@dcs.shef.ac.uk (F. Ciravegna).

1570-8268/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2005.10.002

```
<kmi-basic-portal-ontology:kmi-planet-news-item
rdf:ID="planet-news-story358" >
  <aktive-portal-ontology:has-title>KMi successful at ISWC 2004</aktive-portal-
ontology:has-title>
  <aktive-portal-ontology:has-author rdf:resource="#martin-dzbor" />
  <aktive-portal-ontology:has-date  rdf:resource="#date-2004-11-23 />
  <aktive-portal-ontology:has-story-content>This year International Semantic Web
Conference ISWC 2004 was another successful occasion marking presence of KMi in the
Semantic Web research community - simultaneously on several different fronts. The
conference took place in a wonderful city of peace - Hiroshima, Japan. [•••] And
finally, the presence of KMi in the Semantic Web research community has been confirmed
by appointing Enrico Motta as a Programme Chair for the next year ISWC, which shall
take place in autumn 2005 in Ireland.  Well done!
  </aktive-portal-ontology:has-story-content>
  <aktive-portal-ontology:has-web-address>
http://news.kmi.open.ac.uk/rostra/news.php?r=11&t=2&id=698
</aktive-portal-ontology:has-web-address>
  <kmi-basic-portal-ontology:mentions-kmi-person
rdf:resource="#enrico-motta" />
  <kmi-basic-portal-ontology:mentions-kmi-person
rdf:resource="#john-domingue" />
  <kmi-basic-portal-ontology:mentions-kmi-person
rdf:resource="#martin-dzbor" />
  <kmi-basic-portal-ontology:mentions-kmi-person
rdf:resource="#liliana-cabral" />
  <kmi-basic-portal-ontology:mentions-kmi-person
rdf:resource="#arthur-stutt" />
  <kmi-basic-portal-ontology:mentions-non-kmi-person
rdf:resource="#jim-hendler" />
  <kmi-basic-portal-ontology:mentions-non-kmi-person
rdf:resource="#mark-musen" />
  <kmi-basic-portal-ontology:mentions-organization
rdf:resource="#lancaster-university" />
  <kmi-basic-portal-ontology:mentions-organization
rdf:resource="#the-international-semantic-web-conference" />
  <kmi-basic-portal-ontology:mentions-organization
rdf:resource="#workshop" />
  <kmi-basic-portal-ontology:mentions-project rdf:resource="#magpie" />
  <kmi-basic-portal-ontology:mentions-project rdf:resource="#buddyspace" />
</kmi-basic-portal-ontology:kmi-planet-news-item>
```

Fig. 1. Example of a document with semantic annotation from KMi's semantic website.

Semantic Web annotations go beyond familiar textual annotations about the content of the documents, such as "clause seven of this contract has been deleted because . . .", "the test results need to go in here". This kind of informal annotation is common in word processor applications and is intended primarily for use by document creators. Semantic annotation formally identifies concepts and relations between concepts in documents, and is intended primarily for use by machines. For example, a semantic annotation might relate "Paris" in a text to an ontology which both identifies it as the abstract concept "City" and links it to the instance "France" of the abstract concept "Country", thus removing any ambiguity about which "Paris" it refers to.

Semantic Web annotation brings benefits of two kinds over these systems, enhanced information retrieval and improved interoperability. Information retrieval is improved by the ability to perform searches, which exploit the ontology to make inferences about data from heterogeneous resources [5]. For example, consider the semantic mark-up shown in Fig. 1, which is taken from the Semantic Web site of the Knowledge Media Institute.[1]

The semantic annotations in the example identify people, organizations, and projects, which are mentioned in a web news story, as well as including traditional metadata, such as the author's name and date of publication. Since these statements are integrated with a large departmental ontology, we can then support queries like "give me all the stories which talk about projects on the Semantic Web". A query agent will exploit the semantic annotations to map stories to projects and then use information from the departmental project database to identify only projects related to the Semantic Web area. Ontology-based semantic annotations also allow us to resolve anomalies in searches, e.g. if a document collection were annotated using a geographical ontology, it would become easy to distinguish "Niger" the country from "Niger" the river in searches, because they would be annotated with references to different concepts in the ontology. Interoperability is particularly important for organizations, which have large legacy databases, often in different proprietary formats that do not easily interact. In these circumstances, annotations based on a common ontology can provide a common framework for the integration of information from heterogeneous sources.

As a motivating example of what can be achieved once documents are given semantic mark-up consider the Medical

[1] KMi, The Open University, Semantic website http://semanticweb.kmi.open.ac.uk/.

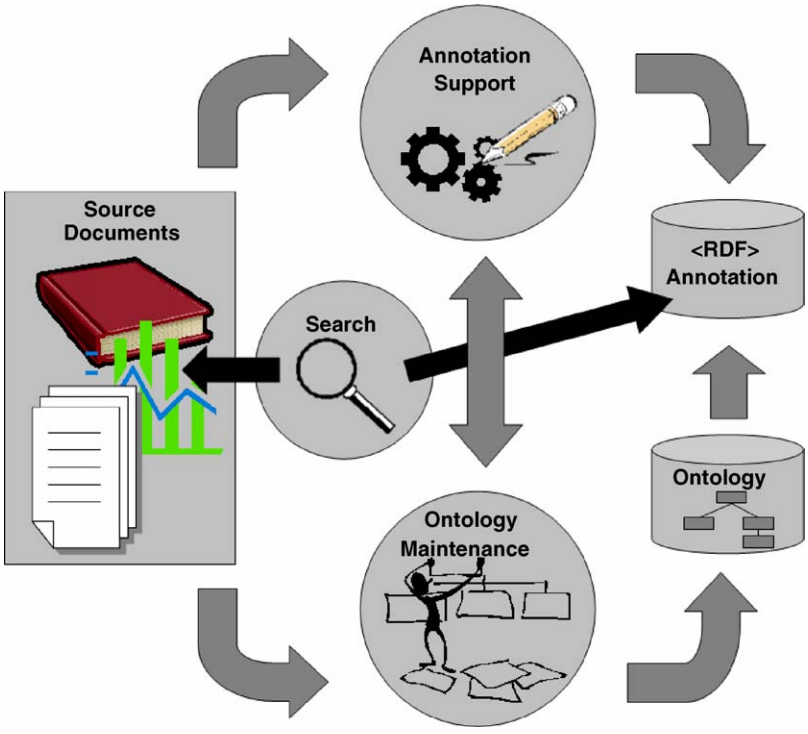

Fig. 2. The role of annotation in document centric KM. Annotations provide interoperability between different kinds of documents and support enhanced search services. Annotation tools draw on knowledge workers' domain knowledge and automatic analysis. Ontologies evolve to fit changing needs.

Imaging and Advanced Knowledge Technologies (MIAKT) project.[2] MIAKT has developed problem solving environments for use in the medical domain. Specifically, it is tackling triple assessment in symptomatic focal breast disease, which involves the interpretation of three different kinds of scan data by groups of medical professionals. In MIAKT the annotations make the knowledge contained in unstructured sources (medical images such as X-rays) available in a structured form, allowing both accurate and focused retrieval and knowledge sharing for a given patient's case. Moreover, the annotations can be used to provide automated services. For example, they can be processed using natural language generation software to automatically draft textual reports about the patient, the diagnostic information that is available and assessments made about the data by the medical team, a task which usually consumes doctors' valuable time [6].

Furthermore, this example is by no means an isolated case. The use of knowledge embodied in annotations is being investigated in domains as diverse as scientific knowledge [7], radio and television news [8], genomics [9], making web pages accessible to visually impaired people [10], employment information [11], online shopping [12] and the description of cultural artifacts in museums [13].

An intelligent, document centric KM process of the type we propose must handle three classes of data: ontologies, documents and annotations. As illustrated in Fig. 2, these need to be supported by new kinds of KM tools. Semantic search tools are needed to connect and exploit the information in annotations and documents. Ontology maintenance tools must support users in maintaining and evolving knowledge models to meet changing needs. Finally, tools are needed to facilitate the annotation of documents, which can detect changes in an ontology related to existing annotations. Annotation tools will, in their turn, need to give feedback to the ontology maintenance process when necessary. Strong coupling is needed between these systems to cope with the re-versioning and reuse of documents, the evolution of the ontologies used to describe them and a range of different users who may require different views on the data or have different access rights.

Annotation is, potentially, an additional burden in this model of KM. Human annotators are prone to error and non-trivial annotations usually require domain expertise, diverting technical staff from other tasks. Also, without maintenance, annotations can easily become obsolete. Therefore, unless annotation can be done cost-effectively the commercial future for the technology is limited. In this paper, we review the systems that currently exist to support the mark-up of documents and determine how well they fit the requirements of KM. Taking the document centric perspective described above, we have identified seven requirements for semantic annotation systems, which we use to assess the capabilities of existing annotation systems. We would like to emphasize that we are exclusively concerned with technical requirements and not with 'soft' requirements related to annotation. Social and psychological aspects are crucial for motivating people to annotate information. However, a discussion of these aspects is, though important, out of the scope of the paper as well as our competencies. For a detailed discussion of soft aspects

[2] Medical Imaging and Advanced Knowledge Technologies (MIAKT) project, http://www.aktors.org/miakt/, accessed on 22 July 2004.

related to the use of annotations for knowledge management the reader is referred to Ref. [14].

Our pool of 'systems' includes two Semantic Web annotation frameworks, which could be implemented differently by different tools, as well as the current generation of manual and automatic tools for semantic mark-up. In a fast developing field, such as this one, it is impossible to complete a comprehensive survey as new tools and new versions of existing tools are emerging constantly. We have tried to survey the general-purpose annotation tools that have some aspect of automation as completely as possible but have had to be more selective with examples of manual tools. We conclude that, while there is still some way to go before semantic annotation tools will be able to address fully all the knowledge management needs identified here, research in the area is very active and constant progress is being made. Semantic annotation tools suitable for large-scale knowledge management can be expected sooner rather than later.

2. Requirements

The document centric model of KM illustrated in Fig. 2 has led us to formulate seven requirements for semantic annotation systems. These overlap to some extent with the requirements set out by Handschuh et al. [15], but there are also differences. For example, we do not concern ourselves with issues such as efficiency and proper reference, although we acknowledge that these are important. Instead, we have considered four viewpoints on the task: the ontologies, the documents, the annotations that link ontologies to documents, and the users of the systems. Each viewpoint suggests one or more requirements, each of which normally brings together several associated needs. For instance, the ontology viewpoint suggests the need for tools to support multiple, evolving ontologies and the document viewpoint suggests the need to support the reuse and versioning of documents.

2.1. Requirement 1—standard formats

Using standard formats is preferred, wherever possible, because the investment in marking up resources is considerable and standardization builds in future proofing because new tools, services, etc., which were not envisaged when the original semantic annotation was performed may be developed. Compliance with standards also frees companies from the constraints of proprietary formats when choosing knowledge management software. These advantages can be applied to systems in general. For annotation systems in particular, standards can provide a bridging mechanism that allows heterogeneous resources to be accessed simultaneously and collaborating users and organizations to share annotations. It is the activity of the W3C in developing and promoting international standards for the Semantic Web that has convinced us that this route is worth following in knowledge management. Two types of standard are required, standards for describing ontologies such as the Web Ontology Language OWL [16] and standards for annotations such as the W3C's RDF annotation schema [17].

2.2. Requirement 2—user centered/collaborative design

Annotation can potentially become a bottleneck if it is done by knowledge workers with many demands on their time. Since few organizations have the capacity to employ professional annotators, it is crucial to provide knowledge workers with easy to use interfaces that simplify the annotation process and place it in the context of their everyday work. A good approach would be a single point of entry interface, so that the environment in which users annotate documents is integrated with the one in which they create, read, share and edit them. System design also needs to facilitate collaboration between users, which is a key facet of knowledge work with experts from different fields contributing to and reusing intelligent documents. We have already identified standard formats as a prerequisite for sharing annotations. Other issues for collaboration include implementing systems to control what to share with whom. For example, in a medical context, physicians might share all information about patients among themselves but only share anonymized information with planners. This brings us to issues related to trust, provenance and access rights. An intranet provides a more controlled environment for tracing the provenance of annotations than the wild Web but access policies are a critical issue to organizations, which are invariably concerned with confidentiality issues for client and staff data.

2.3. Requirement 3—ontology support (multiple ontologies and evolution)

In addition to supporting appropriate ontology formats, annotation tools need to be able to support multiple ontologies. For example, in a medical context, there may be one ontology for general metadata about a patient and other technical ontologies that deal with diagnosis and treatment. Either the ontologies must be merged or annotations must explicitly declare which ontology they refer to. In addition, systems will have to cope with changes made to ontologies over time, such as incorporating new classes or modifying existing ones. In this case, the problem is ensuring consistency between ontologies and annotations with respect to ontology changes. Multiple ontologies and evolving ontologies have been discussed elsewhere in the context of KM, e.g. [18]. Some of the important issues for the design of an annotation environment are to determine how changes should be reflected in the knowledge base of annotated documents and whether changes to ontologies create conflicts with existing annotations. There are also design implications for ontology support as knowledge workers may require facilities to help them explore and edit the ontologies they are using.

2.4. Requirement 4—support of heterogeneous document formats

Semantic Web standards for annotation tend to assume that the documents being annotated are in web-native formats such as HTML and XML. For example, the Annotea approach to locating an annotation at a particular point in a document uses XPointers. This approach will have limited usefulness for KM.

Documents will be in many different formats including word processor files, spreadsheets, graphics files and complex mixtures of different formats. This presents a technical challenge rather than a research challenge, but dealing with multiple document formats is a prerequisite for integrating annotation into existing work practices.

2.5. Requirement 5—document evolution (document and annotation consistency)

Ontologies change sometimes but some documents change many times. An example is W3C's specification documents, which go through multiple revisions. Requirement 3 concerns the problem of keeping ontologies and annotations consistent. This requirement concerns consistency from a textual point of view, i.e. maintaining correct pointers from the annotations to the surface representation in the text. What should happen to the annotations on a document when it is revised, poses both technical and application specific questions. If the anchor for an annotation in a shared document is removed during editing should the current document author be informed, so that they can re-anchor or delete the annotation, or is the original author of the annotation the only person with the right to do this? Is it even desirable, in general, to transfer annotations to a new version of a document, or do versions of annotations need to be maintained in parallel with document versions. For example, if a contract were prepared for a new client, annotations that referred to a legal ontology could be retained, but annotations, which referred to previous clients, could be removed. How can this selective transfer of annotations be achieved? Annotation environments need to help knowledge workers maintain appropriate annotations as documents change.

2.6. Requirement 6—annotation storage

The Semantic Web model assumes that annotations will be stored separately from the original document, whereas the "word processor" model assumes that comments are stored as an integral part of the document, which can be viewed or not as the reader prefers. The Semantic Web model, which decouples content and semantics, works particularly well for the Web environment in which the authors of annotations do not necessarily have any control over the documents they are annotating. In a KM environment, however, many annotators are more familiar with the document-centric, word processor model. They argue that, as they have control of documents, storing annotations as a part of those documents is preferable and helps them to keep annotations consistent with new document versions. We will consider both these storage models for annotations in KM.

2.7. Requirement 7—automation

Another aspect of easing the knowledge acquisition bottleneck is the provision of facilities for automatic mark-up of document collections to facilitate the economical annotation of large document collections. To achieve this, the integration of knowledge extraction technologies into the annotation environment is vital. These can automatically identify entities that are instances of a particular class and relations between the classes. Once again, HCI implications are important so that automated tools can be used effectively by knowledge workers without expertise in, say, natural language processing methods.

3. Annotation frameworks

Having specified the requirements, we now look at general frameworks for annotation, which could be implemented differently by different tools. We discuss two frameworks for annotation in the Semantic Web, the W3C annotation project Annotea [19], and CREAM [20], an annotation framework being developed at the University of Karlsruhe.

Annotea [19,21] is a W3C project, which specifies infrastructure for annotation of Web documents, with emphasis on the collaborative use of annotations. The use of open standards is a very important principle for all the work of W3C to promote interoperability and extensibility. The main format for Annotea is RDF and the kinds of documents that can be annotated are limited to HTML or XML-based documents. This is restrictive for KM, as much commercial data is in other formats. However, it provides in XPointer a method for locating annotations within a document. XPointer is a W3C recommendation for identifying fragments of URI resources. So long as the component of a document to which an XPointer refers is retained, the location of the associated annotation will be robust to changes in the detail of the document, but if large-scale revisions are made, annotations can easily come adrift from their anchor points. The Annotea approach concentrates on a semi-formal style of annotation, in which annotations are free text statements about documents. These statements must have metadata (author, creation time, etc.) and may be typed according to user-defined RDF schemata of arbitrary complexity. In this respect, Annotea is not quite as formal as would be ideal for the creation of intelligent documents. The storage model proposed is a mixed one with annotations being stored as RDF held either on local machines or on public RDF servers. The Annotea framework has been instantiated in a number of tools including Amaya, Annozilla and Vannotea (see Section 4.1).

The **CREAM** framework [15] looks at the context in which annotations could be made and used as well as the format of the annotations themselves. It specifies components required by an annotation system including the annotation interface, with automatic support for annotators, document management system and annotation inference server. Like Annotea, CREAM subscribes to W3C standard formats with annotations made in RDF or OWL and XPointers used to locate annotations in text, which restricts it to web-native formats such as XML and HTML. Unlike Annotea, the authors of CREAM have considered the possibility of annotating the deep web. This involves annotating the databases from which deep web pages are generated so that the annotations are generated automatically with the pages. As databases hold much of the legacy data in companies, this is a substantial addition. It is supported by a storage model that allows users to choose whether they want to store annotations separately on a server or embedded in a web page. This

assumes more user control of the document and recognizes that users may prefer to store annotations with the source material. The CREAM framework allows for relational metadata, defined as "annotations which contain relationship instances". Relational metadata is essential for constructing knowledge bases which can be used to provide semantic services. Examples of tools based on the CREAM framework are S-CREAM and M-OntoMat-Annotizer (see Section 4.1).

4. Semantic annotation tools

Having examined frameworks for annotation, which could be implemented in different ways, we now turn our attention to specific tools which can produce semantic annotations, i.e. annotations that reference an ontology. These are a first generation of tools which meet some of the requirements outlined above but which need further development to make a fully integrated annotation environment. Table 1 provides a summary and below we describe each system briefly.

4.1. Manual annotation

The most basic annotation tools allow users to manually create annotations. They have a great deal in common with purely textual annotation tools but provide some support for ontologies. There are several such programs which produce Annotea RDF mark-up. For example, the W3C Web browser and editor **Amaya** [22] can mark-up Web documents in XML or HTML. The user can make annotations in the same tool they use for browsing and for editing text, making Amaya a good example of a single point of access environment. It has facilities for manual annotation of web pages but does not contain any features to support automatic annotation. The **Annozilla**[3] browser aims to make all Amaya annotations readable in the Mozilla browser and to shadow Amaya developments. Teknowledge[4] produces a similar plug in for Internet Explorer.

The **Mangrove** system is another example of manual but user friendly annotation [23]. The aim of the system was to "entice" users into marking up their HTML by using the data created in a number of semantic services such as a departmental who's who and a calendar of events. The annotation tool itself is a straightforward GUI that allows users to associate a selection of tags to text that they highlight. Mangrove has recently been integrated with a semantic email service [24], which supports the initiation of semantic email processes, such as meeting scheduling, via text forms.

Multimedia annotation is the next phase of development for annotation, expanding the range files types that can be marked-up into images, video and audio. **Vannotea** [25] has been developed by the University of Brisbane for adding metadata to MPEG-2 (video), JPEG2000 (image) and Direct 3D (mesh) files, with the mesh being used to define regions of images. It is of particular interest from the view point of distributed knowledge management because it has been designed to allow input from distributed users. This has, for example, allowed it to be deployed to annotate cultural artifacts in a collaborative annotation exercise involving both museum curators and indigenous groups [13].

Some manual annotation tools have been developed to provide more sophisticated user support and a degree of semi-automatic or automatic annotation facilities. The **OntoMat** Annotizer is a tool for making annotations which is built on the principles of the CREAM framework. It has a Web browser to display the page which is being annotated and provides some reasonably user friendly functions for manual annotation, such as drag and drop creation of instances and the ability to mark-up pages while they are being created. OntoMat has been extended to include support for semi-automatic annotation. The first of these extensions was **S-CREAM** [15], which uses an information extraction (IE) system (Amilcare [26]). The user annotates and the system learns how to reproduce the user annotation, to be able to suggest annotations for new documents. OntoMat also incorporates methods for deep annotation [27], i.e. annotation for Web pages that are generated from databases. Other research in the CREAM family focuses on extending annotation to multimedia formats. **M-OntoMat-Annotizer** [28] supports manual annotation of image and video data by indexers with little multimedia experience by automatic extraction of low level features that describe objects in the content. A commercial version of OntoMat, called **OntoAnnotate**,[5] is available from Ontoprise.

The Mindswap lab at the University of Maryland has developed annotation systems for both Simple HTML Ontology Extensions (SHOE) and RDF. **SHOE Knowledge Annotator** [29] was an early system which allowed users to mark-up HTML pages in SHOE guided by ontologies available locally or via a URL. Users were assisted by being prompted for inputs. Unusually, the SHOE Knowledge Annotator did not have a browser to display Web pages, which could only be viewed as source code. **Running SHOE** [29] took a step towards automated mark-up by assisting users to build wrappers for Web pages that specify how to extract entities from lists and other pages with regular formats. Mindswap is continuing to develop a range of Semantic Web tools [30]. A recent addition of relevance to this survey is the RDF annotator **SMORE**[6] which allows mark-up of images and emails as well as HTML and text.

A tool with similar characteristics to SMORE is the **Open Ontology Forge** (OOF) [31]. OOF is seen by its creators at the national Institute of Informatics, Japan, as an ontology editor that supports annotation, taking it a step further towards an integrated environment to handle documents, ontologies and annotations.

The **COHSE** Annotator [32] produces annotations that are compatible with Annotea standards, although the annotations

[3] Annozilla annotator (http://annozilla.mozdev.org/index.html accessed on 3 August 2004).

[4] Teknowledge Annotation Applications (http://mr.teknowledge.com/DAML/ accessed on 3 August 2004).

[5] OntoAnnotate (http://www.ontoprise.de/products/ontoannotate accessed on 30 November 2004).

[6] SMORE: Semantic Markup, Ontology and RDF Editor (http://www.mindswap.org/~aditkal/editor.shtml accessed on 28 July 2004).

Table 1
Comparison of annotation tools for requirements 1–6

Annotation tool	Standard formats	User-centred design	Ontology support	Document formats	Document evolution	Annotation storage
Amaya	RDF(S) XLink, XPointer	Web browser & editor	Annotation server	HTML, XHTML and XML	XPointer	Local or annotation server
Mangrove	RDF	Graphical annotation tool		HTML, email		RDF database (Jena)
Vannotea	XML	Collaboration support		MPEG-2, JPEG2000, Direct3D		Annotation server
OntoMat	DAML + OIL, OWL, SQL, XPointer	Drag & drop, create & annotate	OntoBroker annotation inference server	HTML, Deep Web	XPointer, pattern matching	Annotation server, embedded in webpage, separate file
M-OntoMat-Annotizer	XML, RDF(S) DOLCE	Automatic extraction of visual descriptors		MPEG-7		Annotation server
SHOE Knowledge annotator	SHOE	Prompting	Ontology server	HTML		Embedded in Webpage
SMORE	RDF(S)	Web browser & editor	Ontology server and ontology editing	HTML, text, email and images		
Open Ontology Forge	RDF(S), XML, Xlink XPointer, Dublin Core	Web browser + drag & drop, create & annotate	Local, editable ontologies	HTML, text, images (SVG)	XPointer	Local RDF or XML file
COHSE annotator	DAML+OIL	Plug in for Mozilla & IE	Ontology server	HTML (via DOM)	XPointer	Annotation server, DLS
Lixto	Wrappers					
MnM	RDF(S), DAML+OIL, OCML	Web browser	Ontology server	HTML, text	Stores annotated page	Embedded in Webpage
Melita	RDF(S) DAML + OIL	Control of intrusiveness of IE	Local, editable ontologies	HTML, text	Regular expressions	
Parmenides	XML (CAS)		Clustering to suggest additions			
Armadillo	RDF(S)			HTML		RDF triple store
KnowItAll	HTML					
SmartWeb	RDF, RDF(S), OWL					RDF knowledge base
PANKOW	HTML	CREAM				
AeroSWARM (AeroDAML)	OWL	Web service	Local ontologies	HTML		
SemTag	RDF(S)			HTML		Label bureau (PICS)
KIM	RDF(S), OWL	Various plug-in front ends	KIMO	HTML		RDF(S) knowledge base
Rainbow Project	RDF WSDL/SOAP	AmphorA XHTML database	Shared upper level ontology	HTML		RDF repository (Sesame)
h-TechSight	DAML + OIL RDF	KM Portal	Ontology editor, dynamics metrics	HTML		Tagged HTML web server
WiCKOffice	Microsoft Smart Documents	Office applications, support for form filling		Microsoft Office		Annotation server (3 store)
AktivDoc	HTML RDF	Integrated editing environment		HTML		RDF triple store
SemanticWord	DAML+OIL	Microsoft Word GUIs		Word	Mark-up tied to text regions	
Magpie	HTML OCML	Web browser plug in		HTML		None, real time
Thresher	RDF	Web browser (Thresher)	Ontology personalization	HTML		None, real time

The comparison for requirement 7 (automation) is given in Table 2.

Table 2
Comparison of annotation tools for requirement 7 (automation)

Annotation tool	Automation	Type of analysis for automation	Learning in automation
Amaya	No		
Mangrove	No		
Vannotea	No		
OntoMat	Yes	PANKOW, Amilcare	Supervised learning
M-OntoMat-Annotizer	Yes	Extraction of spatial descriptors	Genetic algorithm
SHOE knowledge annotator	Yes	Running SHOE (wrappers)	No
SMORE	Yes	Screen scraper	No
Open Ontology Forge	Yes	String matching	No
COHSE annotator	Yes	Ontology string matching	No
Lixto	Yes	Wrappers	No
MnM	Yes	POS tagging, Named Entity Recognition	Supervised learning
Melita	Yes	String matching, POS tagging, Named Entity Recognition	Supervised learning
Parmenides	Yes	Text mining with constraints	Unsupervised learning
Armadillo	Yes	String matching, POS tagging, Named Entity Recognition	Unsupervised learning
KnowItAll	Yes	String matching, Hearst patterns	Unsupervised learning
SmartWeb	Yes	Shallow linguistic parsing	Unsupervised learning
PANKOW	Yes	Hearst patterns	Unsupervised learning
AeroSWARM (AeroDAML)	Yes	AeroText	No
SemTag	Yes	Seeker, similarity, TBD	Unsupervised learning
KIM	Yes	String matching, POS tagging, Named Entity Recognition	No
Rainbow project	Yes	Hidden Markov models, bit-map classification	Supervised learning
h-TechSight	Yes	Shallow linguistic analysis (POS tagging, Named Entity Recognition)	No
WiCKOffice	Yes	Named Entity Recognition	No
AktiveDoc	Yes	String matching, POS tagging, Named Entity Recognition	Supervised and unsupervised learning
SemanticWord	Yes	AeroDAML	No
Magpie	Yes	String-matching, Named Entity Recognition	No
Thresher	Yes	Screen scraping, wrappers	Supervised learning

are conceived as hyperlinks stored using the Distributed Links Service [33]. In this scenario, automatically applied hyperlinks are acceptable but only a word-matching service that highlights ontology terms in the text has been implemented so far. The annotator is provided as a plug-in suitable for use in Mozilla or Internet Explorer, giving the user a choice of working environment. The COHSE architecture has been used to support a number of domain applications, including the generation of semantic annotation for visually impaired users [10] and enriching a Java tutorial site [34].

4.2. Automatic annotation

In this group, we consider both annotation tools that include automation components which provide suggestions for annotations, but still require intervention by knowledge workers, and tools which acquire annotations automatically on a large scale. Some are still limited to usage by specialists while others are suitable for knowledge workers. Automated systems intended to support knowledge workers take into account user interface design issues related to minimizing intrusiveness while maximizing accuracy.

Automation can generally be regarded as falling into three categories. The most basic kind use rules or wrappers written by hand that try to capture known patterns for the annotations. Then there are two kinds of systems that learn how to annotate. Supervised systems learn from sample annotations marked up by the user. A problem with these methods is that picking enough good examples is a non-trivial and error-prone task. In order to tackle this problem unsupervised systems employ a variety of strategies to learn how to annotate without user supervision, but their accuracy is still limited. A summary of the automation aspects of the systems reviewed here is given in Table 2.

Lixto is a web information extraction system which allows wrappers to be defined for converting unstructured resources into structured ones. The tool allows users to create wrappers interactively and visually by selecting relevant pieces of information [35]. It was originally developed at the Technical University of Vienna by Gottlob and colleagues and is now distributed by the spin-off Lixto Software GmbH.[7]

MnM was designed to mark-up training data for IE tools rather than as an annotation tool per se [36]. This means that it stores marked up documents as tagged versions of the original, rather than the RDF formats used by the Semantic Web community. It has reasonable user support, with an HTML browser to display the document and ontology browser features. A strength of MnM is that it provides open APIs to link to ontology servers and for integrating information extraction tools, making it flexible about the formats and methods it uses.

Melita [37] is a user driven automated semantic annotation tool which makes two main strategies available to the user. On the one hand, it provides an underlying adaptive information

[7] Lixto Software GmbH (http://www.lixto.com accessed on 16th September 2005).

extraction system (Amilcare) that learns how to annotate the documents by generalizing on the user annotations. Annotation is therefore a process that starts by requiring full user annotation at early stages, but ends in having the user merely verify the correctness of suggestions made by the system. On the other hand, it provides facilities for rule writing (based on regular expressions) to allow sophisticated users to define their own rules. In Melita, documents are not selected randomly for annotation, but rather selected automatically based on the expected usefulness, to the IE system, of annotating the document. The Amilcare IE system has been incorporated in **K@**, a legal KM system with RDF based semantic capabilities produced by Quinary [38].

CAFETIERE is a rule-based system for generating XML annotations developed as part of the **Parmenides** project [39], which has been used, for example, to annotate the GENIA biomedical corpus [9]. Text mining techniques supplemented with slot based constraints are used to suggest annotations to analysts [40]. The Parmenides project also experimented with a clustering approach to suggest concepts and relations to extend ontologies [41].

Armadillo is a system for unsupervised creation of knowledge bases from large repositories (e.g. the Web) as well as document annotation [42]. It uses the redundancy of the information in repositories to bootstrap learning from a handful of seed examples selected by the user. Seeds are searched in the repository. Then Adaptive IE is used to generalize over the examples and find new facts. Confirmation by several sources (e.g. documents) is then required to check the quality of the newly acquired data. After confirmation, a new round of learning can be initiated. This bootstrapping process can be repeated until the user is satisfied with the quality of the learned information. Armadillo uses a number of techniques, from keyword based searches, to adaptive IE to information integration.

KnowItAll [43] automates extraction of large knowledge bases of facts from the Web in a similar fashion to Armadillo. The most notable difference is the way the system assesses the plausibility of candidate extractions. This is done using the pointwise mutual information (PMI) measure rather than weighing multiple evidence from domain-specific oracles. The PMI measure is roughly the ratio between the number of search engine hits obtained by querying with the discriminator phrase (e.g. "Liege is a city") by the number of hits obtained by querying with the extracted fact (e.g. "Liege"). Also, KnowItAll does not require any set of initial seeds. Besides the baseline system, the authors have provided three extensions to the system (pattern learning, subclass extraction and list extraction) which are shown to improve overall performance.

The **SmartWeb** project is also investigating unsupervised approaches for RDF knowledge base population [44]. Their approach resolves the issue of not having pre-existing mark-up to learn from by using class and subclass names from the ontology to construct examples. The context of these examples is then learnt. In this way, instances can be identified which have similar contexts, but which may use different terminology to the ontology. Smart Web is aimed at broadband mobile access and plans an initial demonstrator for the 2006 football world cup.

Another approach to learning annotations which exploits the sheer size of the Web is Pattern-based Annotation through Knowledge On the Web (**PANKOW**) [45]. PANKOW uses a range of relatively rare, but informative, syntactic patterns to mark-up candidate phrases in Web pages without having to manually produce an initial set of marked-up Web pages and go through a supervised learning step.

AeroSWARM[8] is an automatic tool for annotation using OWL ontologies based on the DAML annotator **AeroDAML** [46]. This has both a client server version and a Web enabled demonstrator in which the user enters a URI and the system automatically returns a file of annotations on another web page. To view this in context the user would have to save the RDF to an annotation server and view the results in an annotation friendly browser such as Amaya.

SemTag is another example of a tool which focuses only on automatic mark-up [47]. It is based on IBM's text analysis platform Seeker and uses similarity functions to recognize entities which occur in contexts similar to marked up examples. The key problem of large-scale automatic mark-up is identified as ambiguity, e.g. identical strings, such as "Niger" which can refer to different things, a river or a country. A Taxonomy Based Disambiguation (TBD) algorithm is proposed to tackle this problem. SemTag is proposed as a bootstrapping solution to get a semantically tagged collection off the ground. It is intended as a tool for specialists rather than one for knowledge workers.

KIM [48,49], uses information extraction techniques to build a large knowledge base of annotations. The annotations in KIM are metadata in the form of named entities (people, places, etc.) which are defined in the KIMO ontology and identified mainly from reference to extremely large gazetteers. This is restrictive, and it would be a significant research challenge to extend the KIM methodology to domain specific ontologies. However named entities are a class of metadata with broad usage, For example, in the **Rich News** application KIM has been used to help annotate television and radio news by exploiting the fact that Web news stories on the same topics are often published in parallel [8]. The KIM platform is well placed to showcase the kinds of retrieval and data analysis services that can be provided over large knowledge bases of annotations. For example, the KIM server is able to use a variety of plug in front ends, including one for Microsoft's Internet Explorer, a Web UI that provides different semantic search services, and a graph viewer for exploring the connections between entities. The development of KIM is set to continue in collaboration with DERI Galway and the GATE research team under the banner of **SWAN**[9], a Semantic Web Annotator.

The **Rainbow project**, based at the University of Economics, Prague, is taking a web-mining led approach to automating annotation. Rainbow is in fact a family of independent applications which share a common web-service front end and upper level ontology [50]. The applications include text mining from

[8] AeroSWARM project page (http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html accessed on 2 August 2004).

[9] http://www.deri.ie/projects/swan/ accessed August 2005.

product catalogues as well as more general pattern matching applications such as pornography recognition in bit-map image files. The generated RDF is stored in Sesame databases for semantic retrieval [12].

A traditional approach to information extraction is used by the **h-TechSight** Knowledge Management Platform, in which the GATE rule-based IE system is used to feed a semantic portal [51]. This work is of particular interest because the automatically generated annotations are monitored to produce metrics describing the "dynamics" of concepts and instances which can be fed back to end users [11]. It is envisaged that dynamics data will be used to inform the manual evolution of ontologies.

4.3. Integrated annotation environments

We emphasized in the requirements the need for single point of entry systems that incorporate the annotation process with knowledge workers' everyday tasks. In this section, we review systems that are aimed at integrating annotation into standard tools and making annotation simultaneous to writing.

WiCKOffice [52] explores this approach. It demonstrates how writing within a knowledge aware environment has useful support possibilities, such as automatic assistance for form filling using data extracted from knowledge bases.

AktiveDoc [53] enables annotation of documents at three levels: ontology based content annotation, free text statements and on-demand document enrichment. Support is provided during both editing and reading. Semi-automatic annotation of content is provided via Adaptive Information Extraction from text (using Amilcare). As AktiveDoc is designed for knowledge reuse, it is able to monitor editing actions and to provide automatic suggestions about relevant content. Support is not limited to filling forms and other pre-determined structures (as in WICK-Office), but it is extended to free text as well. This enables timely reuse of existing knowledge when available. Armadillo supports searches of relevant knowledge in large repositories; annotations in the document are used as context for searches. Annotations are saved in a separate database; levels of confidentiality are associated to annotations so to ensure confidentiality of knowledge when necessary.

AeroDAML can provide automation within authoring environments. For example, the **SemanticWord** annotator [54], which provides GUI based tools to help analysts annotate Microsoft Word documents with DAML ontologies as they write. A commercial annotation system for Microsoft Office applications called **OntoOffice**[10] is available from Ontoprise.

4.4. On-demand annotation

In this section, we describe two systems which are not strictly annotation tools. Instead, they produce annotation-like services on demand for users browsing un-annotated resources. In this way, they fill a niche for resources which it is either impossible to annotate, such as external web pages, documents which change rapidly, or those which might be annotated but with an unsuitable ontology.

Magpie [55], for example operates from within a web browser and does "real-time" annotation of web resources by highlighting text strings related to an ontology of the user's choice. Appropriate web services can be linked to highlighted strings. While the annotation of documents is automatic, Magpie currently has the disadvantage that subject specific parts of the lexicons of text strings for each ontology have to be produced manually (common named entities such as people's names and organizations can be highlighted with a Named Entity Recognition plug-in called ESpotter). Work on automating lexicon generation is in progress.

The **Thresher** system is similar to Magpie in that it uses wrappers to generate RDF on the fly as users browse deep web resources [56]. As with Magpie, the user can access semantic services for recognized objects. Writing wrappers is a complex task which Thresher tackles by providing facilities for non-technical users to mark-up examples of a particular class. These are then used to induce wrappers automatically. Because Thresher is part of the Haystack semantic browser [57] users can also personalize the ontologies they use.

5. Automation

Automation is a particularly important requirement because it is needed to ease the knowledge acquisition bottleneck, particularly for annotating large collections of legacy documents. The kinds of support provided for annotating text can be classified into four kinds, wrappers, IE systems incorporating supervised learning, IE systems that use some unsupervised machine learning, and natural language processing systems. Many of the systems we reviewed had one or more of these kinds of automatic support for annotators (see Table 2 for a summary).

The most common form of support in the current generation of tools is wrappers as originally developed by Kushmerick et al. [58], which exploit the structure of Web pages to identify nuggets of information for mark-up. Wrappers and rules are most useful when there are very regular patterns in the documents, such as standard tables of data. They require skill on the part of the user. Ciravegna et al. [37] give as an example of a typical user editable pattern for finding times of events in their Melita system:

$$\backslash d : \backslash d \backslash d \backslash W + (AM|PM|am|pm)$$

That this pattern or "regular expression" is intended to extract time expressions would be clear to most programmers and all information extraction specialists (it means a digit followed by symbol":" then 2 digits, a word and either AM or PM in capital letters or lowercase letters). The average knowledge worker, however, would certainly need support in deciphering the symbols and would probably prefer it to be translated into some form of natural language template that "looks like" the text it represents.

[10] OntoOffice tutorial (http://www.ontoprise.de/documents/tutorial_ontooffice.pdf accessed on 30 November 2004).

The Thresher system, which allows non-technical users to generate wrappers automatically from examples, is a good example of how progress is being made towards a more user-centric approach to wrapper generation.

Supervised IE systems (e.g. Amilcare, used by S-CREAM, MnM and Melita) learn how to recognize the objects that require annotation by learning from a collection of previously annotated documents. This usually requires the mark-up of a considerable collection of documents. The MnM system, for example, was built to investigate how this task could be facilitated for domain experts. Merely marking a number of documents is not sufficient; the items marked need to be good examples of the kinds of contexts in which the items are found. Finding the right mix of exemplar documents is a tougher challenge for non IE experts than the time-consuming work of marking up a sample of documents. Melita addressed this problem by suggesting the best mix of documents for annotation. Unsupervised systems, like Armadillo, are starting to tackle these challenges by exploiting unsupervised learning techniques. PANKOW (used in OntoMat), for example, demonstrates how the distribution of certain patterns on the Web can be used as evidence in order to approximate the formal annotation of entities in Web pages by a principle of 'annotation by maximal (syntactic) evidence'. For example, the number of times the phrase "cities such as Paris" occurs on web pages, would supply one piece of evidence that Paris is a city, which would be considered in the light of counts of other patterns containing "Paris". The successor C-PANKOW extends PANKOW by taking into account the local context of the web page the entity to be annotated appears in [59].

Users of automatic annotation systems need to be aware of their limitations. Broadly speaking these are missing annotations (known technically as low recall) and incorrect annotations (known as low precision), and they trade off against each other. However, for organizations with large collections of legacy data in particular, imperfect annotation may be preferable to no annotation.

Additional issues for IE in KM are discussed by Ciravegna [60]. Cimiano et al. [61] identify an additional problem, relation extraction, that we need to address here. This is critical to the mark-up of ontological information and the creation of intelligent documents. Most IE systems can recognize concept instances and values, but they are not able to establish explicit relations between entities. For this reason, if a document contains more than one instance of a concept, the system will not be able to allocate the correct properties to the correct instance because it is unable to differentiate among them. A typical example is a home page with several names and phone numbers. The IE system would not be able to match phone numbers to persons. The problem of relation detection is under active investigation in the information extraction community, e.g. through the ACE exercises,[11] and progress on this issue can be expected in the next few years.

[11] ACE exercises (http://www.itl.nist.gov/iad/894.01/tests/ace/ accessed on 30 November 2004).

6. Requirements revisited

In the above survey of annotation tools, we have examined a selection of tools that allow manual mark-up, plus a number of annotators supporting semi-automatic and automatic mark-up. Next, we look again at the seven requirements of annotation tools for KM to see how the tools measure up to them, where progress is being made, and where there are still challenges to be met.

6.1. Requirement 1—standard formats

We identified standardization of the format of annotations as essential to build in future proofing and compatibility of data with the widest possible range of systems. The survey shows that the W3C standards, particularly Annotea, are becoming dominant in this area. Systems like CAFETIERE, which uses its own XML based annotation scheme, are rare. This requirement has been fulfilled, although the standards may need to be augmented to tackle inadequacies in the existing standards (see discussion of requirement 5).

6.2. Requirement 2—user centered/collaborative design

Our ideal semantic annotation system would use a single point of entry approach in which annotation functionality, including access to maintain the underlying ontologies, would be seamlessly integrated with other tools routinely used by knowledge workers to author and read documents. This does not yet exist although there are signs of a trend towards integrated authoring environments, such as WickOffice and AktiveDoc. The most common home environment of the tools we have seen is a Web browser, a natural result of the fact that most of them were designed for the Semantic Web. Even for KM, this has the advantage of being a very familiar technology. The downside is that it both focuses development on native Web formats like HTML and XML and tends to divorce the annotation process from the process of document creation. More attention needs to be paid to build in or plug-in semantic annotation facilities in commonly used packages to encourage knowledge workers to view annotation as part of the authoring process not as an afterthought, and also to supporting annotation in collaborative environments, as for example in Vannotea. Most of the tools reviewed in this survey did not address issues of provenance or access rights. Concerning the specification of access policies, standard methods to restrict access to databases or the file system are available. Offering this kind of support for trust, provenance and access policies concerning annotations is an important issue which needs to be addressed to make Semantic Web annotations a viable knowledge management tool.

6.3. Requirement 3—ontology support (multiple ontologies and evolution)

Annotation tools have adapted rapidly to recent changes in ontology standards for the Web, with many of the more recent tools already supporting OWL. However, support for

doing anything more complex than searching and navigating an ontology browser is the exception. Ontology maintenance, which directly affects the maintenance of annotations, is poorly supported, or not supported at all, by the current generation of tools. This perhaps reflects the intended user groups; with the assumption being that knowledge workers will use existing ontologies rather than editing or creating them. However there are signs that annotation systems are giving users more control of ontologies. Melita allows users to split a concept and then view all the instances that have been created for the old concept and reassign them. The COHSE architecture includes a component for maintaining the ontology but this does not appear to be available from the annotator. The Open Ontology Forge supports the creation of new classes from a root class. Much more is still required. A genuinely integrated semantic annotation environment should give the user automatic support for ontology maintenance, for example, using text mining methods to suggest new classes as they emerge in documents and spotting inconsistencies between new and existing annotations. h-TechSight has made a start in this direction by monitoring the dynamics of instances and concepts to assist end-users in manual ontology evolution. Parmenides has gone rather further and experimented with clustering methods to suggest ontology changes. However there is still a long way to go and we believe that ontology maintenance represents a significant research challenge.

6.4. Requirement 4—support of heterogeneous document formats

Satisfying this requirement is a prerequisite for producing integrated annotation environments and our survey suggests that the range of document types that can be handled is expanding, though few individual systems handle many different formats. Most of the annotation tools we looked at supported only HTML and XML. WICKOffice and OntoOffice provided annotation for word processor files. Mangrove and SMORE provided facilities for handling emails. Open Ontology Forge, SMORE, Vannotea and M-OntoMat-Annotizer provided means to annotate images and image regions. The Rich News application of KIM provided an interesting example of how an area where automation expertise exists (text based IE) can be used to support the automatic annotation of more difficult (audio visual) media. This kind of cross media approach is likely to prove fertile ground for the development of environments that take a more integrated approach to handling heterogeneous formats.

6.5. Requirement 5—document evolution (document and annotation consistency)

We have observed that keeping annotations synchronized with changes to documents is challenging and this is one area in which the current annotation standards are inadequate. The Annotea approach adopted by many of the tools, stores annotations separately from the document and uses XPointer to locate them in the document. There are strong arguments in favour of separate storage of annotations and documents, some which we will discuss in requirement 6, but the problem with the XPointer approach is that connections are one way from annotations to documents and, therefore, too easily broken by edits at the document end. An environment in which documents and annotations are stored separately, but closely coordinated is required. A number of practical fixes have been implemented in OntoMat, including the ability to search for similar documents that have already been annotated, and a proposal to use pattern matching to help relocate annotations in suitable places in the new document. However, these are ways of coping with the problem. For KM applications a coordinated approach is needed to tackle the issues of versioning annotations as documents evolve. These include determining who has permission to edit annotations, at which points in the document life cycle it is appropriate to update the annotations, what automatic interventions are possible to reduce the burden on users, etc. Our survey did not discover any concerted work on these lines.

6.6. Requirement 6—annotation storage options

In the Semantic Web, documents and their annotations are stored separately. This is unavoidable since documents and annotations are likely to be owned by different people or organizations and stored in different places. A variety of approaches to separate storage were seen in the tools we examined. The Annotea approach calls for RDF servers. Web storage technologies that have been used are RDF triplestore (Armadillo and AktiveDoc), Label Bureaus (SemTag) and DLS (COHSE).

In an organizational setting, with greater control of documents, an alternative model is to store annotations directly in the document. This is familiar to knowledge workers from the current text comment facilities for word processors, spreadsheets, etc. We have also seen it used for example in SemanticWord and MnM. This approach is appealing, not just because of its familiarity, but because users believe it avoids the problem of keeping annotations and documents consistent (in fact it just places all responsibility for consistency on the human editors of the document). However, separate storage of annotations has advantages for KM. The resulting decoupling of semantics and content facilitates document reuse because it is possible to set up rules which control and automate which kinds of annotations are transferred to new documents and which are not. It allows information from heterogeneous resources to be queried centrally as a knowledge base. It also makes it easy to produce different views of a document for users with different roles in an organization or different access rights, thus facilitating knowledge sharing and collaboration. We therefore argue that separate storage is the better model, even when extra overheads are required to maintain links between a document and its annotations.

6.7. Requirement 7—automation

Automation is vital to ease the knowledge acquisition bottleneck, as discussed above. Many of the systems we examined had some kind of automatic and semi-automatic support for annotation. Most of these handled just text, using mainly wrappers, IE and natural language processing although there are some systems, notably M-OntoMat-Annotizer and parts of the Rainbow

Project looking to automate the handling of other media. Language technologies present usability challenges when deployed for knowledge workers since most are research tools or designed for use by specialists. A first step in this direction is Melita, where attention has been paid in finding ways to enable a seamless user interaction with the underlying IE system. In addition to the usability challenges there are also research challenges, among which we have highlighted the extraction of relations as important for semantic annotation.

7. Summing up

Documents are central to KM, but intelligent documents, created by semantic annotation, would bring the advantages of semantic search and interoperability. These benefits, however, come at the cost of increased authoring effort. We have, therefore, argued that integrated systems are needed which support users in dealing with the documents, the ontologies and the annotations that link documents to ontologies within familiar document authoring environments. These systems need automation to support annotation, automation to support ontology maintenance, and automation to help maintain the consistency of documents, ontologies and annotations. They also need to support collaborative working. In this area we have identified the lack of access controls as a limitation of the Semantic Web approach for knowledge management.

The two Semantic Web frameworks Annotea and CREAM favour different aspects of annotation activity. Annotea, with its emphasis on collaboration has influenced the development of a number of excellent systems with good user interfaces that are well suited to distributed knowledge sharing. CREAM, with its greater emphasis on the deep Web and the annotation of legacy resources has pushed the development of annotation systems more aimed towards corporate KM.

Our review of existing annotation systems indicates that research is very active and there are many systems which provide some of the requirements, but that fully integrated environments are still some way off. WiCKOffice and AktiveDoc present exemplars of what integrated authoring environments might look like but are currently limited with respect to their degree of automation and range of documents covered. Technical challenges to development include supporting multi-media document formats, particularly automating the annotation of media other than text, addressing issues of trust, provenance and access rights and resolving the problems of storage. Problems around keeping annotations consistent with evolving documents, present a considerable challenge, particularly in combination with evolving ontologies and the field needs to draw on ontology maintenance research to tackle this challenge.

Acknowledgements

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), the Designing Adaptive Information Extraction for Knowledge Management (Dot.Kom) project, the 6th Framework EU IST project “aceMedia”, the DARPA DAML project “OntoAgents” (01IN901C0) and the SmartWeb project, funded by the German Ministry of Research. AKT is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. Dot.Kom is sponsored by the European Commission as part of the Information Society Technologies (IST) programme under grant number IST-2001034038.

References

[1] S. Olsen, IBM sets out to make sense of the Web, 2004, CNET News.com (http://news.com.com/2100-1032_3-5153627.html accessed on 14 September 2005).
[2] Delphi Group, The document is the process, White Paper, Delphi Consulting Group Inc., 1994, http://www.delphigroup.com/research/whitepapers/DocIsProcess.pdf.
[3] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Sci. Am. (2001) 34–43.
[4] P. Gardenförs, How to make the Semantic Web more semantic, in: Proceedings of the 3rd International Conference on Formal Ontology in Information Systems, 2004, IOS Press, 2004.
[5] C. Welty, N. Ide, Using the right tools: enhancing retrieval from marked-up documents, J. Comput. Humanit. 33 (10) (1999) 59–84.
[6] K. Bontcheva, Y. Wilks, Automatic report generation from ontologies: the MIAKT approach, in: Proceedings of the 9th International Conference on Applications of Natural Language to Information Sys-tems (NLDB’2004), 2004, Manchester, UK, 2004.
[7] N.S. Friedland, P.G. Allen, G. Matthews, M. Witbrock, D. Baxter, J. Curtis, B. Shepard, P. Miraglia, J. Angele, S. Staab, E. Moench, H. Oppermann, D. Wenke, D. Israel, V. Chaudhri, B. Porter, K. Barker, J. Fan, S.Y. Chaw, P. Yeh, D. Tecuci, Project halo: towards a digital Aristotle, AI Magazine, Winter 2004, 2004.
[8] M. Dowman, V. Tablan, H. Cunningham, B. Popov, Web-assisted annotation, semantic indexing and search of television and radio news, in: Proceedings of the 14th International World Wide Web Conference (WWW2005), May 10–14, Chiba, Japan, 2005, pp. 225–234.
[9] F. Rinaldi, G. Schneider, K. Kaljurand, J. Dowdall, A. Persidis, O. Konstanti, Mining relations in the GENIA corpus, Second European Workshop on Data Mining and Text Mining for Bioin-formatics, 24 September 2004, Pisa, Italy, 2004.
[10] P. Plessers, S. Casteleyn, Y. Yesilada, O. De Troyer, R. Stevens, S. Harper, C. Goble, Accessibility: a web engineering approach, in: Proceedings of the 14th International World Wide Web Conference (WWW2005), May 10–14, Chiba, Japan, 2005, pp. 353–362.
[11] D. Maynard, M. Yankova, N. Aswani, H. Cunningham, Automatic creation and monitoring of semantic metadata in a dynamic knowledge portal, in: Proceedings on the Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2004), LNAI 3192, 2004, pp. 65–74.
[12] O. Svab, M. Labsky, V. Svatek, RDF-based retrieval of information extracted from web product catalogues, in: Proceedings of the SIGIR’04 Semantic Web Workshop, Sheffield, 2004.
[13] J. Hunter, R. Schroeter, B. Koopman, M. Henderson, Using the semantic grid to build bridges between museums and indigenous communities, in: Proceedings of the GGF11—Semantic Grid Applications Workshop, Honolulu, June 10, 2004, 2004.
[14] Harvard Business School, Harvard Business Review on Knowledge Management, Harvard Business School Publishing, Boston, MA, USA, 1998.
[15] S. Handschuh, S. Staab, R. Studer, Leveraging metadata creation for the Semantic Web with CREAM, KI ’2003—advances in artificial intelligence, in: Proceedings of the Annual German Conference on AI, September 2003, 2003.
[16] D.L. McGuinness, F. van Harmelen, OWL Web Ontology Language Overview, 2004 (http://www.w3.org/TR/owl-features/ accessed on 23 July 2004).
[17] R. Swick, E. Prud’hommeaux, M.-R. Koivunen, J. Kahan, Anno tea Protocols, 2004 (http://www.w3.org/2001/Annotea/User/Protocol, accessed on 23 July 2004).

[18] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz, Ontologies for Enterprise KM, IEEE Intell. Syst. 18 (2) (2003) 26–33.
[19] J. Kahan, M.-J. Koivunen, E. Prud'Hommeaux, R. Swick, Annotea: an open RDF infrastructure for shared web annotations, in: Proceedings of the 10th International World Wide Web Conference (WWW 2001), Hong Kong, 2001.
[20] S. Handschuh, S. Staab, Authoring and annotation of web pages in CREAM, in: Proceedings of the 11th International World Wide Web Conference (WWW 2002), 7–11 May 2002 Honolulu, Hawaii, USA, 2002.
[21] M.-R. Koivunen, Annotea and Semantic Web supported collaboration, Invited talk at Workshop on User Aspects of the Semantic Web (UserSWeb) at European Semantic Web Conference (ESWC 2005), 29 May 2005, Heraklion, Greece, 2005.
[22] V. Quint, I. Vatton, An Introduction to Amaya, W3C NOTE 20-February-1997, 1997 (http://www.w3.org/TR/NOTE-amaya-970220.html accessed on 28 July 2004).
[23] L. McDowell, O. Etzioni, S. Gribble, A. Halevy, H. Levy, W. Pentney, D. Verma, S. Vlasseva, Enticing ordinary people onto the Semantic Web via instant gratification, in: Proceedings of the 2nd International Semantic Web Conference (ISWC 2003), October 2003, 2003.
[24] L. McDowell, O. Etzioni, A. Halevy, Semantic email: theory and applications, J. Web Semantics 2 (2) (2004) 153–183.
[25] R. Schroeter, J. Hunter, D. Kosovic, Vannotea, A collaborative video indexing, annotation and discussion system for broadband networks, in: Proceedings of the K-CAP 2003 Workshop on "Knowledge Markup and Semantic Annotation", October 2003, Florida, 2003.
[26] F. Ciravegna, Y. Wilks, Designing adaptive information extraction for the Semantic Web in amilcare, in: S. Handschuh, S. Staab (Eds.), Annotation for the Semantic Web, in the Series Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, 2003.
[27] R. Volz, S. Handschuh, S. Staab, L. Stojanovic, N. Stojanovic, Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the Semantic Web, J. Web Semantics 1 (2) (2004) 187–206.
[28] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, M.G. Strintzis, Semantic annotation of images and videos for multimedia analysis, in: Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), 29 May–1 June 2005, Heraklion, Greece, 2005.
[29] J. Heflin, J. Hendler, A portrait of the Semantic Web in action, IEEE Intell. Syst. 16 (2) (2001) 54–59.
[30] J. Golbeck, M. Grove, B. Parsia, A. Kalyanpur, J. Hendler, New tools for the Semantic Web, in knowledge engineering and knowledge management (ontologies and the Semantic Web), in: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), 1–4 October 2002, Sigüenza (Spain), LNAI 2473, Springer Verlag, 2002, pp. 392–400.
[31] N. Collier, A. Kawazoe, A.A. Kitamoto, T. Wattarujeekrit, T.Y. Mizuta, A. Mullen, Integrating deep and shallow semantic structures in open ontology forge, in: Proceedings of the Special Interest Group on Semantic Web and Ontology, JSAI (Japanese Society for Artificial Intelligence), vol. SIG-SWO-A402-05, 2004.
[32] S. Bechhofer, C. Goble, Towards annotation using DAML + OIL, in: Proceedings of the Workshop on Semantic Markup and Annotation at 1st International Conference on Knowledge Capture (K-CAP 2001),Victoria, B.C., Canada.
[33] L. Carr, D. de Roure, W. Hall, G. Hill, The distributed link service: a tool for publishers, authors and readers, World Wide Web J. 1 (1) (1995) 647–656.
[34] S. Bechhofer, C. Goble, L. Carr, W. Hall, S. Kampa, D. De Roure, COHSE: conceptual open hypermedia service, in: S. Handschuh, S. Staab (Eds.), Annotation for the Semantic Web, IOS Press, Amsterdam, 2003.
[35] R. Baumgartner, R. Flesca, Gottlob G, Visual web information extraction with Lixto, in: Proceedings of the International Conference on Very Large Data Bases (VLDB), 2001.
[36] Vargas-Vera M., E. Motta, J. Domingue, M. Lanzoni, A. Stutt, F. Ciravegna, MnM: A tool for automatic support on semantic markup, KMi Technical Report, September 2003, TR Number 133, 2003.
[37] F. Ciravegna, A. Dingli, D. Petrelli, Y. Wilks, User-system cooperation in document annotation based on information, in: Proceedings of the 13th International Conference on Knowledge Engineering and KM (EKAW02), 1–4 October 2002, Sigüenza, Spain, 2002.
[38] L. Gilardoni, M. Biasuzzi, M. Ferraro, R. Fonti, P. Slavazza, Machine learning for the Semantic Web: putting the user in the cycle, in: Proceedings of the Dagstuhl Seminar, Machine Learning for the Semantic Web, 13–18 February 2005, 2005.
[39] W.J. Black, J. McNaught, A. Vasilakopoulos, K. Zervanou, B. Theodoulidis, F. Rinaldi, CAFETIERE conceptual annotations for facts, events, terms, individual entities, and relations, Parmenides Technical Report, 11 Jan, 2005, TR-U4.3.1, 2005.
[40] A. Vasilakopoulos, M. Bersani, W.J. Black, A Suite of Tools for Marking Up Textual Data for Temporal Text Mining Scenarios, in: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004), 24–30 May 2004, Lisbon, 2004.
[41] M. Siliopoulou, F. Rinaldi, W.J. Black, G.P. Zarri, R.M. Mueller, M. Brunzel, B. Theodoulidis, G. Orphanos, M. Hess, J. Dowdall, J. McNaught, M. King, A. Persidis, L. Bernard, Coupling information extraction and data mining for ontology learning in PARMENIDES, in: Proceedings of the Recherche d'Information Assistée par Ordinateur (RIAO'2004), 2004.
[42] F. Ciravegna, S. Chapman, A. Dingli, Y. Wilks, Learning to harvest information for the Semantic Web, in: Proceedings of the 1st European Semantic Web Symposium, May 10–12, 2004, Heraklion, Greece, 2004.
[43] O. Etzioni, M.J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, A. Yates, Unsupervised named-entity extraction from the Web: an experimental study, Artif. Intell. 165 (1) (2005) 91–134.
[44] P. Buitelaar, S. Ramaka, Unsupervised ontology based semantic tagging for knowledge markup, in: Proceedings of the Workshop on Learning in Web Search at the International Conference on Machine Learning, August 2005, Bonn, Germany, 2005.
[45] P. Cimiano, S. Handschuh, S. Staab, Towards the self-annotating web, in: Proceedings of the 13th International World Wide Web Conference (WWW 2004), May 17–22, 2004, New York, NY, 2004.
[46] P. Kogut, W. Holmes, AeroDAML: applying information extraction to generate DAML annotations from web pages, in: Proceedings of the Workshop on Knowledge Markup and Semantic Annotation at 1st International Conference on Knowledge Capture (K-CAP 2001), Victoria, B.C., Canada, 2001.
[47] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K.S. McCurley, S. Rajagopalan, A. Tomkins, J.A. Tomlin, J.Y. Zienberer, A case for automated large scale semantic annotation, J. Web Semantics 1 (1) (December 2003).
[48] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov, Towards Semantic Web information extraction, in: Proceedings of the Human Language Technologies Workshop at 2nd International Semantic Web Conference (ISWC2003), 20 October 2003, Florida, USA, 2003.
[49] B. Popov, A. Kirayakov, D. Ognyanoff, D. Manov, A. Kirilov, KIM—a semantic platform fo information extraction and retrieval, Nat. Lang. Eng. 10 (3/4) (2004) 375–392.
[50] V. Svatek, M. Labsky, M. Vacura, Knowledge modelling for deductive web mining, in: Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004), Whittlebury Hall, Northamptonshire, UK, 2004.
[51] D. Maynard, M. Yankova, A. Kourakis, A. Kokossis, Ontology-based information extraction for market monitoring and technology watch, Proceedings of the Workshop on User Aspects of the Semantic Web (UserSWeb) at European Semantic Web Conference (ESWC 2005), 29 May 2005, Heraklion, Greece, 2005.
[52] L. Carr, T. Miles-Board, A. Woukeu, G. Wills, W. Hall, The case for explicit knowledge in documents, in: Proceedings of the ACM Sympo-

sium on Document Engineering (DocEng '04), Oct 28–30, Milwaukee, Wisconsin, USA, 2004, pp. 90–98.
[53] V. Lanfranchi, F. Ciravegna, D. Petrelli, Semantic Web-based document: editing and browsing in AktiveDoc, in: Proceedings of the 2nd European Semantic Web Conference, May 29–June 1, 2005, Heraklion, Greece, 2005.
[54] M. Tallis, SemanticWord processing for content authors, in: Proceedings of the Knowledge Markup and Semantic Annotation Workshop (SEMANNOT 2003) at 2nd International Conference on Knowledge Capture (K-CAP 2003), October 26, 2003. Sanibel, Florida, USA, 2003.
[55] M. Dzbor, E. Motta, J. Domingue, Opening up magpie via semantic services, in: Proceedings of the 3rd International Semantic Web Conference, November 2004, Hiroshima, Japan, 2004.
[56] A. Hogue, D. Karger, Thresher: automating the unwrapping of semantic content from the world wide web, in: Proceedings of the 14th International World Wide Web Conference (WWW2005), May 10–14, Chiba, Japan, 2005, pp. 86–95.
[57] D. Huynh, D. Kerger, D. Quan, Haystack: a platform for creating, organizing and visualizing information using RDF, in: Proceedings of the 11th International World Wide Web Conference (WWW2002), Hawaii, USA, 2002.
[58] N. Kushmerick, D. Weld, B. Doorenbos, Wrapper induction for information extraction, in: Proeedings of the International Joint Conerence on Artificial Intelligence, 1997.
[59] P. Cimiano, G. Ladwig, S. Staab, Gimme' the context: context-driven automatic semantic annotation with C-PANKOW, in: Proceedings of the 14th International World Wide Web Conference (WWW 2005), Tokyo, Japan, 2005.
[60] F. Ciravegna, Challenges in information extraction from text for KM, Proceedings of the IEEE Intelligent Systems and Their Applications, November 2001 (Trend and Controversies), 2001.
[61] P. Cimiano, F. Ciravegna, J. Domingue, S. Handschuh, A. Lavelli, S. Staab, M. Stevenson, Requirements for information extraction for KM, in: Proceedings of the KM and Semantic Annotation Workshop at 2nd International Conference on Knowledge Capture (KCAP-2003), 2003.

Victoria Uren is a Research Fellow at the Knowledge Media Institute (Open University). Her research concerns the application of natural language technologies for knowledge acquisition and management. She has worked on several projects within KMi including ScholOnto, AKT and Dot.Kom.

Philipp Cimiano is a researcher at the Institute AIFB (University of Karlsruhe). He is currently working in the German BMBF project SmartWeb and co-organizing an ontology learning challenge within the PASCAL framework. He has worked on several projects including the Dot.Kom and Genome Information Extraction (GenIE) projects. His main interests include natural language processing and understanding as well as knowledge acquisition from texts.

Jose Iria is a Research Associate in the Department of Computer Science at the University of Sheffield. He has worked on several projects including Dot.Kom and Abraxas. His research interests include the application of Machine Learning techniques to Information Extraction for automated extraction of entities and relations from text.

Siegfried Handschuh is a researcher at the FZI (Research Center for Information Technology) Karlsruhe. His current research interests include 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". He is currently working at FZI for the Ontoprise team on the HALO 2 project. Previously he worked at the Institute AIFB (University of Karlsruhe) in the Onto Agents project in the DARPA DAML program.

Maria Vargas-Vera is a researcher at the Knowledge Media Institute (Open University) currently working on the AKT project. She is interested in the use of Ontologies in Natural Language Processing (NLP). In particular Ontology-Based Information Extraction, Question Answering, Automated Assessment of Student Essays and Semi-Automatic Construction of Ontologies from text.

Enrico Motta is Professor in Knowledge Technologies and Director of the Knowledge Media Institute (Open University). His current research focuses primarily on the integration of semantic, web and language technologies to support knowledge acquisition and management and problem solving.

Fabio Ciravegna is Professor of Language and Knowledge Technologies at the University of Sheffield. He coordinates the Web Intelligence Technologies Lab, part of the Natural Language Processing group. His current research focuses primarily on Language Technologies for Knowledge Management and in particular on techniques for acquisition, sharing and reuse of information and knowledge in Semantic Web oriented environments.