

**Title:** Building Semantic Intranets: What is Needed in the Annotation Toolbox?

**Corresponding Author:**

Victoria Uren,  
Knowledge Media Institute,  
The Open University,  
Walton Hall,  
Milton Keynes,  
MK7 6AA, UK.

[v.s.uren@open.ac.uk](mailto:v.s.uren@open.ac.uk)

Fax: (01908) 653169

**Other Authors:**

Philipp Cimiano, Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany. [cimiano@aifb.uni-karlsruhe.de](mailto:cimiano@aifb.uni-karlsruhe.de)

José Iria, Department of Computer Science, University of Sheffield, Regent Court, 211Portobello Street, Sheffield S1 4DP, UK. [j.iria@dcs.shef.ac.uk](mailto:j.iria@dcs.shef.ac.uk)

Siegfried Handschuh, Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany. [sha@aifb.uni-karlsruhe.de](mailto:sha@aifb.uni-karlsruhe.de)

Maria Vargas-Vera, Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. [m.vargas-vera@open.ac.uk](mailto:m.vargas-vera@open.ac.uk)

Enrico Motta, Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. [e.motta@open.ac.uk](mailto:e.motta@open.ac.uk)

Fabio Ciravegna, Department of Computer Science, University of Sheffield, Regent Court, 211Portobello Street, Sheffield S1 4DP, UK. [f.ciravegna@dcs.shef.ac.uk](mailto:f.ciravegna@dcs.shef.ac.uk)



# Building Semantic Intranets: What is Needed in the Annotation Toolbox?

Victoria Uren<sup>1</sup>, Philipp Cimiano<sup>2</sup>, José Iria<sup>3</sup>, Siegfried Handschuh<sup>2</sup>,  
Maria Vargas-Vera<sup>1</sup>, Enrico Motta<sup>1</sup>, and Fabio Ciravegna<sup>3</sup>

<sup>1</sup> Knowledge Media Institute,  
The Open University,  
Walton Hall, Milton Keynes, MK7 6AA, UK  
{e.motta; v.s.uren; m.vargas-vera}@open.ac.uk

<sup>2</sup> Institute AIFB  
University of Karlsruhe  
D-76128 Karlsruhe  
Germany  
{cimiano; sha}@aifb.uni-karlsruhe.de

<sup>3</sup> Department of Computer Science,  
University of Sheffield,  
Regent Court, 211 Portobello Street,  
Sheffield S1 4DP, UK  
{f.ciravegna; j.iria}@dcs.shef.ac.uk

**ABSTRACT.** While much of a company's explicit knowledge can be found in text repositories, current content management systems only provide limited capabilities for structuring and making sense of documents. In the emerging Semantic Web, however, some of the traditional document search, interpretation and aggregation problems can be addressed by ontology-based semantic markup, which enables intelligent search and information extraction. In this paper, we examine semantic annotation in detail, we identify a number of requirements which need to be fulfilled if semantic web solutions are to address knowledge management needs, and we review the current generation of semantic annotation systems. This analysis shows that there is still some way to go before semantic annotation tools will be able to address fully all the knowledge management needs identified by our analysis.

**KEYWORDS.** Knowledge Management; Annotation; Annotation systems; Automatic annotation

## 1 INTRODUCTION

Does Semantic Web technology matter for Knowledge Management (KM)? KM is often seen as centering on documents. These provide an explicit representation of 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 [25].

Therefore the Semantic Web does matter because it envisages technologies which can make possible the generation of the kind of “intelligent” documents imagined ten years ago [13]. 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 [2]. The result is Web pages with machine interpretable mark up that provide the source material with which agents and Semantic Web services operate.

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, which may, for example, formally identify concepts and relations between concepts in documents, is intended primarily for use by machines.

Semantic Web annotation brings benefits of two kinds, enhanced information retrieval and improved interoperability. Information retrieval is improved by the ability to perform annotation-based search. For example, consider the semantic markup shown in figure 1, which is taken from the semantic web site of the Knowledge Media Institute<sup>1</sup>. 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 in the departmental project database to identify only projects related to the semantic web

---

<sup>1</sup>KMi, The Open University, semantic website <http://plainmoor.open.ac.uk:8080/ksw/index.html>

area. Please note that we are not saying anything here about the provenance of this information, e.g., whether it has been manually encoded, or automatically extracted.

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 contained in documents with other sources of information, e.g. legacy databases.

As a motivating example of what can be achieved once documents are given semantic markup consider the MIAKT (Medical Imaging and Advanced Knowledge Technologies) project<sup>2</sup>. MIAKT is developing 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 among doctors examining the same patient 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 assessment made about the data by the medical team, a task which usually consumes doctors’ valuable time [4].

An intelligent, document centric KM process of the type we propose must handle three classes of data: ontologies, documents and annotations. As illustrated in figure 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 imposed by this model of KM. Therefore unless annotation can be done cost-effectively the commercial future for the technology is limited. In this paper, we examine the systems that currently exist to support the markup of documents and determine how well they fit the requirements of KM. Taking the document centric perspective described above, we have

---

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

identified seven requirements for semantic annotation systems, which we use to assess the capabilities of existing annotation systems. 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. As discussed later, our analysis shows that there is still some way to go before semantic annotation tools will be able to address fully all the knowledge management needs identified by our study.

## **2 REQUIREMENTS**

The document centric model of KM illustrated in figure 2 has led us to formulate seven requirements for semantic annotation systems. These overlap to some extent with the requirements set out by Handschuh and Studer [18], 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.

### **REQUIREMENT 1 - Standard formats**

Using standard formats for both annotations and the ontologies they refer to is preferred, wherever possible, because the investment in marking up resources is considerable and standardization builds in future proofing. The resources created are more likely to be compatible with tools, services etc., which were not envisaged when the original semantic annotation was performed. Standardization also facilitates interoperability of disparate resources. W3C has been active in developing and promoting standards such as RDF annotation schema [29], and the Web ontology language OWL [24].

### **REQUIREMENT 2 - User centered design**

Annotation can potentially become a bottleneck if it is done by knowledge workers with many demands on their time. An expensive alternative is to employ professional annotators. Hence 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 every day work. A good approach would be a single point of entry interface, so that the environment in which users annotate documents is the same as the one in which they create, read 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.

### **REQUIREMENT 3 Ontology support**

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. Multiple ontologies and evolving ontologies have been discussed elsewhere in the context of KM, e.g. [23]. 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.

#### **REQUIREMENT 4 - 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 (though not a complete solution to) integrating annotation into existing work practices.

#### **REQUIREMENT 5 - Document evolution**

Ontologies change sometimes but some documents change many times. Keeping annotations synchronized with changes to documents is challenging. An example is given by W3C's specification documents which go through multiple revisions. 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 document is removed during editing does the document author need to know or the author of the annotation? While it might be desirable, in general, to transfer annotations to a new document, if a contract were prepared for a new client, annotations that referred to a legal ontology should be retained, but annotations which referred to previous clients should be removed. How can this selective transfer of annotations be achieved? Annotation environments need to help knowledge workers maintain the consistency of annotations as documents change.

#### **REQUIREMENT 6 – Annotation storage**

The Semantic Web model assumes that annotations will be stored separately from the original document, whereas models such as commenting in word processor documents generally store the comments 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.

### **REQUIREMENT 7 - Automation**

Another aspect of easing the knowledge acquisition bottleneck is the provision of facilities for automatic markup 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 in text 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 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 [21], and CREAM [8], an annotation framework being developed at the University of Karlsruhe.

**Annotea** [21] is a W3C project which specifies infrastructure for annotation of Web documents. 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 ready made structure 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 **CREAM** framework [19] 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 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, embedded in a web page or on a separate server. 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.

## 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 annotators 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 markup. For example, the W3C Web browser and editor **Amaya** [27] 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**<sup>3</sup> browser aims to make all Amaya annotations readable in the Mozilla browser and to shadow Amaya developments. Teknowledge<sup>4</sup> produces a similar plug in for Internet Explorer.

Some fundamentally manual annotation tools 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**, [19], which uses an information extraction (IE) system (Amilcare [10]). The user annotates and the system learns how to reproduce the user annotation, to be able to suggest

---

<sup>3</sup> Annozilla annotator (<http://annozilla.mozdev.org/index.html> accessed on 3 Aug. 2004)

<sup>4</sup> Teknowledge Annotation Applications (<http://mr.teknowledge.com/DAML/> accessed on 3 Aug. 2004)

annotations for new documents. Work is now underway on **PANKOW** (Pattern-based Annotation through Knowledge On the Web) [7] an approach which exploits a range of 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. OntoMat also incorporates methods for deep annotation, i.e. annotation for Web pages that are generated from databases. A commercial version of OntoMat, called **OntoAnnotate**<sup>5</sup>, is available from Ontoprise, which also produces an annotation system for Microsoft Office applications called **OntoOffice**<sup>6</sup>.

The University of Maryland has developed annotation systems for both SHOE (Simple HTML Ontology Extensions) and RDF. **SHOE Knowledge Annotator** [20] 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** [20] 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. A more recent contribution from the University of Maryland is the RDF annotator **SMORE**<sup>7</sup>. SMORE allows markup of images and emails as well as HTML and text, which is an important step towards annotation of a greater range of document formats. A tool with similar characteristics to SMORE is the **Open Ontology Forge** (OOF) [12]. 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 [1] produces annotations that are compatible with Annotea standards, although the annotations are conceived as hyperlinks stored using the Distributed Links Service [5]. In this scenario automatically applied hyperlinks are acceptable but only a word-matching services 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.

## 4.2 Semi-automatic Annotation

The next group of annotation tools that we will discuss is semi-automatic, they have automatic components but assume intervention by knowledge workers in the annotation process. **MnM** was designed to mark up training data for IE tools rather than as an annotation tool *per se* [31]. 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

---

<sup>5</sup> OntoAnnotate (<http://www.ontoprise.de/products/ontoannotate> accessed on 30 Nov. 2004)

<sup>6</sup> OntoOffice tutorial ([http://www.ontoprise.de/documents/tutorial\\_ontooffice.pdf](http://www.ontoprise.de/documents/tutorial_ontooffice.pdf) accessed on 30 Nov. 2004)

<sup>7</sup> SMORE: Semantic Markup, Ontology and RDF Editor (<http://www.mindswap.org/~aditkal/editor.shtml> accessed on 28 July 2004)

ontology servers and for integrating information extraction tools, making it flexible about the formats and methods it uses.

**Melita** [9] 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 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.

### 4.3 Automatic Annotation

The most automated group of tools annotate automatically on a large scale to bootstrap the production of annotated collections. They are often intended for use by specialists rather than by knowledge workers.

The **Armadillo** system tackles the problem of getting annotated examples to learn from by learning in an unsupervised way from a handful of examples selected by the user [11]. It uses these to search a large data source, e.g., the Web, for further examples. Confirmation by several sources is then required to check the quality of the new 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 rules.

**AeroSWARM**<sup>8</sup> is an automatic tool for annotation using OWL ontologies based on the DAML annotator **AeroDAML** [22]. This has both a client server version and a Web enabled demonstrator in which 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. AeroDAML is therefore not in itself an annotation environment. However, it can be integrated into annotation systems, e.g., the **SemanticWord** annotator [30]. This annotator for Microsoft Word provides a sophisticated set of GUI based tools to help analysts annotate Word documents with DAML ontologies as they write.

**SemTag** is another example of a tool which focuses only on automatic mark-up [14]. 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 markup 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.

---

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

**KIM** [26] 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 and 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.

**Magpie** [15] is a semantic web browser which does "real-time" annotation of web resources by highlighting text strings related to an ontology of the users 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.

#### **4.4 Task specific Annotation**

Finally, we will look at a couple of examples of annotation tools which have been produced with particular applications in mind. **TRELLIS** [16] is designed to support argument analysis in decision making scenarios. It demonstrates the additional support that can be given to user when an annotation environment is designed for a specific purpose. For example, annotations in TRELLIS are in the form of free text statements. This presents a problem since statements about the same thing can be phrased differently and consequently not matched up by the user. Therefore a component called ACE has been built which helps users to formulate statements in ways which are consistent with terms in the ontology [3]. The annotations in TRELLIS can be output as RDF. However, perhaps because it is designed as a tool for analyzing a wide range of document formats, the authors do not discuss whether it is possible to anchor annotations to a particular part of a text.

**CAFETIERE** is the annotation editor for the Parmenides Common Annotation Scheme, an XML based schema for events and temporal information which has the advantage of being "layered" so that semantic annotations can be clearly distinguished from structural and lexical ones. It uses text mining techniques supplemented with slot based constraints to suggest annotations to analysts [32]. Parmenides has been used, for example, to annotate the GENIA biomedical corpus [28].

## **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, 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. [9] give as an example of a typical user editable pattern for finding times of events in their Melita system:

$$\backslashd:\backslashd\backslashW+(AM|PM|am|pm)$$

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.

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), demonstrates how searching for simple patterns on the Web can yield extra data can be drawn to find out more about entities in local documents which are to be annotated. The previous sentence, for example, would fit the pattern “class *such as* example”, to yield the information that PANKOW is a natural language system.

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. Particularly for organizations with large collections of legacy data, imperfect annotation may be preferable to no annotation.

Additional issues for IE in KM are discussed by Ciravegna [8]. Cimiano et al. [6] 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 assign phone numbers to persons. The problem of relation detection is under active investigation in the information extraction community, e.g., through the ACE exercises<sup>9</sup>, and progress on this issue can be expected in the next few years.

## **6 UNFULFILLED REQUIREMENTS**

In the above survey of annotation tools we saw that in addition to general tools that allow manual markup, a number of annotators supporting semi-automatic and automatic mark-up have been developed plus some special purpose tools. Next we will look at how far these tools go to meet the seven requirements of annotation tools for KM.

### **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 the discussion of requirement 5).

### **REQUIREMENT 2 - User centered 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. 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. There are some honorable exceptions, such as SemanticWord, but these are rare. More attention needs to be paid to building semantic annotation facilities into 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, which is rarely mentioned.

### **REQUIREMENT 3 Ontology support**

---

<sup>9</sup> ACE exercises (<http://www.itl.nist.gov/iad/894.01/tests/ace/> accessed on 30 Nov. 2004)

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. To our knowledge, no annotation system is currently capable of this level of support. We believe that producing such a system represents a significant research challenge, which we are tackling within the Dot.Kom consortium.

#### **REQUIREMENT 4 - Document formats**

Most of the annotation tools we looked at supported only HTML and XML. Semantic Word and OntoOffice provided annotation for word processor files and SMORE could annotate some kinds of graphics files and emails. Open Ontology Forge also provides means to annotate images and image regions (SVG). TRELLIS is neutral about the format of the documents it describes, but it does not provide links to the context being annotated only metadata describing it. Satisfying this requirement is a prerequisite for producing integrated annotation environments. While there remains a lot of work to do to extend annotation to the many formats in commercial use this is a technical challenge rather than a research challenge.

#### **REQUIREMENT 5 - Document evolution**

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 many 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. Making the changes is required to resolve it is a challenge for the developers of Semantic Web standards.

#### **REQUIREMENT 6 – Annotation storage options**

The problem of handling change in documents is related intimately to the problem of how annotations are stored. 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), Label Bureaus (SemTag) and DLS (COHSE). The application specific tool TRELIS, on the other hand, uses a local knowledge base.

In an organizational setting, with greater control of documents, an alternative model is to store annotation 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. 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 which kinds of annotations are transferred to new documents and which are not. It also makes it easy to produce different views of a document for users with different roles in an organization. We therefore argue that separate storage is the better model, even when extra over-heads are required to maintain links between a document and its annotations.

#### **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 using mainly wrappers, IE and natural language processing. All of these present usability challenges when deploying them for knowledge workers since most are research tools designed for use by specialists. A first step in this direction is Melita, where attention has been paid in finding ways to enable the user to control 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.

Our review of existing annotation systems indicates that such integrated environments are still some way off. Technical challenges to their development include supporting different ontology formats and document formats, and resolving the problems of storage, in particular, problems around keeping annotations consistent with evolving documents, which have not been addressed by the Semantic Web community. The human computer interaction challenges inherent in building integrated systems of this complexity are also very interesting. Research challenges include further improvements to automatic annotation components, such as relation extraction, and developing support systems for ontology evolution.

## ACKNOWLEDGEMENTS

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), the Designing Information Extraction for KM (Dot.Kom) project, the 6th Framework EU IST project "aceMedia", and the DARPA DAML project "OntoAgents" (01IN901C0). 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] Bechhofer S., Goble C. (2001) *Towards Annotation Using DAML+OIL*. In Workshop on Semantic Markup and Annotation. At 1st International Conference on Knowledge Capture (K-CAP 2001). Victoria, B.C., Canada.
- [2] Berners-Lee T., Hendler J., Lassila O. (2001) *The Semantic Web*, Scientific American, 34-43.
- [3] Blythe J., Gil. Y (2004) *Incremental Formalization of Document Annotations through Ontology-Based Paraphrasing*, In Proceedings 13th International World Wide Web Conference, (WWW 2004), May 17-22, 2004, New York, NY.
- [4] Bontcheva K., Wilks Y. (2004) *Automatic Report Generation from Ontologies: the MIAKT approach*. In Proceedings 9th International Conference on Applications of Natural Language to Information Systems (NLDB'2004). Manchester, UK 2004.
- [5] Carr L., de Roure D., Hall W., Hill G. (1995) *The Distributed Link Service: a tool for publishers, authors and readers*. World Wide Web Journal, 1(1): 647-656

- [6] Cimiano P., Ciravegna F., Domingue J., Handschuh S., Lavelli A., Staab S., Stevenson M. (2003) *Requirements for Information Extraction for KM*, In KM and Semantic Annotation Workshop, At 2nd International Conference on Knowledge Capture (KCAP-2003).
- [7] Cimiano P., Handschuh S., Staab S. (2004) *Towards the self-annotating web*. In Proceedings 13th International World Wide Web Conference, (WWW 2004), May 17-22, 2004, New York, NY.
- [8] Ciravegna F. (2001) *Challenges in Information Extraction from Text for KM*. IEEE Intelligent Systems and Their Applications, November 2001, (Trend and Controversies).
- [9] Ciravegna F., Dingli A., Petrelli D., Wilks Y. (2002) *User-System Cooperation in Document Annotation based on Information*, In 13th International Conference on Knowledge Engineering and KM (EKAW02), 1-4 October 2002 – Sigüenza, Spain.
- [10] Ciravegna F., Wilks Y. (2003) *Designing Adaptive Information Extraction for the Semantic Web in Amilcare*, in S. Handschuh and S. Staab (eds), Annotation for the Semantic Web, in the Series Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, 2003.
- [11] Ciravegna F., Chapman S., Dingli A., Wilks, Y. (2004) *Learning to Harvest Information for the Semantic Web*, In Proceedings 1st European Semantic Web Symposium, Heraklion, Greece, May 10-12, 2004.
- [12] Collier N., Kawazoe A., Kitamoto A.A., Wattarujeekrit T., Mizuta T.Y., Mullen A. (2004) *Integrating Deep and Shallow Semantic Structures in Open Ontology Forge*, Special Interest Group on Semantic Web and Ontology, JSAI (Japanese Society for Artificial Intelligence), Vol. SIG-SWO-A402-05.
- [13] Delphi Group, (1994) *The document is the process*, White Paper, Publ: Delphi Consulting Group Inc., <http://www.delphigroup.com/research/whitepapers/DocIsProcess.pdf>.
- [14] Dill S., Eiron N., Gibson D., Daniel D., Guha R., Jhingran A., Kanungo T., Rajagopalan S., Tomkins A., Tomlin J.A., Zien J.Y. (2003) *SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation*, In Proceedings 12th International World Wide Web Conference (WWW 2003), Budapest, Hungary, 20-24 May, 2003.
- [15] Dzbor M., Motta E., Domingue J. (2004) *Opening Up Magpie via Semantic Services*, In Proceedings 3rd International Semantic Web Conference, Hiroshima, Japan, November 2004.
- [16] Gil Y., Ratnakar V. (2002) *TRELLIS: an interactive tool for capturing information analysis and decision making*. EKAW 2002, LNAI 2473, 37-42.
- [17] Handschuh S. and Staab S. (2002) *Authoring and annotation of web pages in CREAM*, In Proceedings 11th International World Wide Web Conference (WWW 2002), Honolulu, Hawaii, USA 7-11 May, 2002.

- [18] Handschuh S., Staab S. (2003) *CREAM: CREATing Metadata for the Semantic Web*, Computer Networks 42(5): 579-598
- [19] Handschuh S., Staab S., Studer R. (2003) *Leveraging metadata creation for the Semantic Web with CREAM*, KI '2003 - Advances in Artificial Intelligence. In Proceedings Annual German Conference on AI, Sept. 2003
- [20] Heflin J., Hendler J. (2001) *A Portrait of the Semantic Web in Action*. IEEE Intelligent Systems, 16(2), 54-59.
- [21] Kahan J., Koivunen M-J., Prud'Hommeaux E., Swick R. (2001) *Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In Proceedings 10th International World Wide Web Conference (WWW 2001) Hong Kong.
- [22] Kogut P., Holmes W. (2001) *AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages*. In Workshop on Knowledge Markup and Semantic Annotation. At 1st International Conference on Knowledge Capture (K-CAP 2001), Victoria, B.C., Canada.
- [23] Maedche A., Motik B., Stojanovic L., Studer R., Volz R., (2003) *Ontologies for Enterprise KM*, IEEE Intelligent Systems 2003 18(2), 26-33
- [24] McGuinness D.L., van Harmelen F. (2004) *OWL Web Ontology Language Overview*, (<http://www.w3.org/TR/owl-features/> accessed on 23 July 2004)
- [25] Olsen S. (2004) *IBM sets out to make sense of the Web*, CNET News.com ([http://news.com.com/2100-1032\\_3-5153627.html?tag=prntfr](http://news.com.com/2100-1032_3-5153627.html?tag=prntfr) accessed on 30 No. 2004)
- [26] Popov B., Kiryakov A., Ognyanoff D., Manov D., Kirilov A., Goranov M. (2003) *Towards Semantic Web Information Extraction*, In Human Language Technologies Workshop. At 2nd International Semantic Web Conference (ISWC2003), 20 October 2003, Florida, USA.
- [27] Quint V., Vatton I. (1997) *An Introduction to Amaya*, W3C NOTE 20-February-1997, (<http://www.w3.org/TR/NOTE-amaya-970220.html> accessed on 28 July 2004)
- [28] Rinaldi F., Schneider G., Kaljurand K., Dowdall J., Persidis A., Konstanti O. (2004) *Mining relations in the GENIA corpus*. Second European Workshop on Data Mining and Text Mining for Bioinformatics, 24 September 2004, Pisa, Italy
- [29] Swick R., Prud'hommeaux E., Koivunen M-R, Kahan J., (2001) *Annotea Protocols* (<http://www.w3.org/2001/Annotea/User/Protocol>, accessed on 23 July 2004)
- [30] Tallis M. (2003) *Semantic Word Processing for Content Authors*, In Knowledge Markup and Semantic Annotation Workshop (SEMANNOT 2003). At 2nd International Conference on Knowledge Capture (K-CAP 2003), October 26, 2003, Sanibel, Florida, USA.
- [31] Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F. (2003) *MnM: A Tool for Automatic Support on Semantic Markup*, KMi Technical Report, TR Number133, Sept. 2003.

[32] Vasilakopoulos A., Bersani M., Black W.J. (2004) *A Suite of Tools for Marking Up Textual Data for Temporal Text Mining Scenarios*. In Proceedings 4th International Conference on Language Resources and Evaluation (LREC-2004), Lisbon, 24-30 May 2004.

```

<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="#maggie" />
  <kmi-basic-portal-ontology:mentions-project rdf:resource="#buddyspace" />
</kmi-basic-portal-ontology:kmi-planet-news-item>

```

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

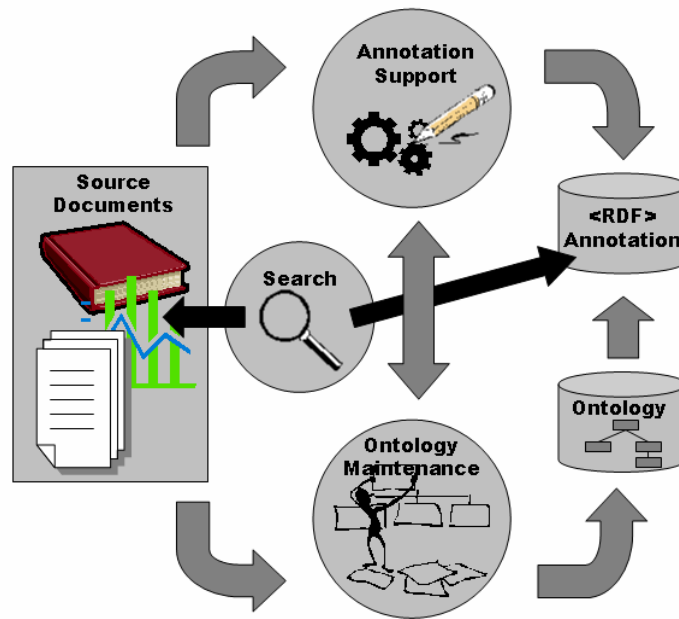


Figure 2: The role of annotation in document centric KM. Annotations provide interoperability between different kinds of documents and support enhanced search services. Annotation tools draw on knowledge workers' domain knowledge and automatic analysis. Ontologies evolve to fit changing needs.

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
OntoMat	DAML+OIL, OWL, SQL, XPointer	drag & drop, create & annotate	OntoBroker annotation inference server	HTML, Deep Web	XPointer, pattern matching	annotation server or embedded in webpage, or separatefile
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 and image regions (SVG)	XPointer	Local RDF or XML file
MnM	RDF(S), DAML+OIL, OCML	Web Browser	ontology server	HTML, text	stores annotated page	embedded in Webpage
COHSE Annotator	DAML+OIL	plug in for Mozilla & IE	Ontology Server	HTML (via DOM)	XPointer	Annotation server or DLS
AeroSWARM (AeroDAML)	OWL	Web service	Local Ontologies	HTML	?	?
SemanticWord	DAML+OIL	Microsoft Word GUIs		Word	Markup tied to text regions	?
SemTag	RDF(S)	?	?	HTML	?	Label bureau (PICS)
Melita	RDF(S) DAML+OIL	Text viewer control of intrusiveness of IE	Local, editable ontologies	HTML, text	Regular expressions	?
Armadillo	RDF(S)	?	?	HTML	?	RDF triple store
KIM	RDF(S), OWL	various plug-in front ends	KIMO	HTML	?	RDF(S) knowledge base
Magpie	Real-time	Web browser	?	HTML	?	none
TRELLIS	XML, RDF DAML/OIL OWL	Assisted statement formulation	Local ontology	Neutral	?	Knowledge base
CAFETIERE	XML (CAS)	?	Supports any Ontology API	?	?	?

**Table 1** Comparison of annotation tools for requirements 1-6.

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

<b>Annotation tool</b>	<b>Wrappers and string matching</b>	<b>Supervised learning</b>	<b>Unsupervised learning</b>	<b>Natural Language Processing</b>
Amaya	?	?	?	?
OntoMat	?	Adaptive IE	?	PANKOW (patterns)
SHOE Knowledge annotator	Running SHOE (wrappers)	?	?	?
SMORE	Screen scraper	?	?	?
Open Ontology Forge	String matching	?	?	?
MnM	?	IE plug-ins	?	?
COHSE Annotator	Ontology string matching	?	?	?
AeroSWARM (AeroDAML)	?	?	?	AeroText
SemanticWord	?	?	?	AeroDAML
SemTag	Seeker, similarity, TBD	?	?	?
Melita	Wrappers, string-matching	Adaptive IE	?	Named Entity recognition, part of speech
Armadillo	Wrappers, string-matching	Adaptive IE	Adaptive IE (modified)	?
KIM	?	?	?	Named Entity Recognition
Magpie	String-matching	?	?	Named Entity Recognition
TRELLIS	?	?	?	?
CAFETIERE	?	?	Text Mining with constraints	?

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