Deliverable D16.2 MPA2

Conceptual Search

DOCUMENT IDENTIFIER	PS_WP16_USFD_D16.2_Conceptual Search
DATE	15/01/2007
ABSTRACT	The current deliverable introduces the necessary background information for the domain, definition of the limitations of contemporary types of search, as well as an overview of advanced search techniques. In the end it presents the solution chosen as a basis for the conceptual search in the context of PrestoSpace: the KIM Platform for semantic annotation, indexing and retrieval, providing semantically-enabled kinds of search, formal knowledge navigation and the opportunity for even more sophisticated search paradigms to be layered on top.
KEYWORDS	
WORKPACKAGE / TASK	WP16 / MPA2
AUTHOR, COMPANY	Atanas Kiryakov (SAI), Kalina Bontcheva (USFD), Valentin Tablan (USFD), Borislav Popov (SAI)
NATURE	Report
DISSEMINATION	Public

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.1	03 – 03 - 2004	First Draft	Living	Confidential
0.2	08 – 03 - 2004	Second Draft	Living	Confidential
0.3	15 – 12 - 2006	Advanced draft	Living	Restricted
1.0	15 – 01 - 2007	Final version	Complete	Public

Table of Contents

1. Executive Summary

The current deliverable introduces the necessary background information for the domain, definition of the limitations of contemporary types of search, as well as an overview of advanced search techniques. In the end it presents the solution chosen as a basis for the conceptual search in the context of PrestoSpace: the KIM Platform for semantic annotation, indexing and retrieval, providing semantically-enabled kinds of search, formal knowledge navigation and the opportunity for even more sophisticated search paradigms to be layered on top.

2. Overview

The current deliverable provides background information about the current search technology exploited widely over large scales of data (e.g. the web) and its limitations. It also introduces the more sophisticated approach of involving formal knowledge representation and in-depth preliminary analysis of the content in order to achieve semantic or conceptual search. The document describes the basic standards in the Semantic Web movement aiming at the next-generation web. Another related aspect described considers metadata and annotations and then continues with semantic annotation. Information Extraction is explained in its ontology-based variation as a main constituent of the mentioned pre-processing. Different tools, libraries and platforms forming the current technological landscape in the domain are also overviewed. Another aspect covered is the gathering and enrichment of background knowledge used to model the domain and to aid the information extraction processing. A bit more emphasis is placed on the KIM Platform for semantic annotation and search as the chosen technology for the PrestoSpace project needs.

3. Background

3.1. Limitations of Current Search Technology

In general, when specifying a search, users enter a small number of terms in the query. The query describes the information need, and is commonly based on the words that people expect to occur in the types of document they seek. This gives rise to a fundamental problem, known as “index term *synonymy*”: not all documents will use the same words to refer to the same concept. Therefore, not all the documents that discuss the concept will be retrieved by a simple keyword-based search. Furthermore, query terms may of course have multiple meanings; this problem can be called “query term *polysemy*“. As conventional search engines cannot interpret the sense of the user's search, the ambiguity of the query leads to the retrieval of irrelevant information.

Technically, the above two problems can be explained as follows: search engines that match query terms against a keyword-based index will fail to match relevant information when the keywords used in the query are different from those used in the index, despite having the same meaning. This problem can be overcome to some extent through thesaurus-based expansion of the query; this approach increases the level of document recall, but it may result in significant precision decay, i.e. the search engine returning too many results for the user to be able to process realistically.

Users can partly overcome query ambiguity by careful choice of additional query terms. However, there is evidence to suggest that many people may not be prepared to do this. For example, an analysis of the transaction logs of the Excite WWW search engine [Jansen et al., 2000] showed that web search engine queries contain

on average 2.2 terms.

In addition to difficulties in handling synonymy and polysemy, conventional search engines are of course unaware of any other semantic links between terms (or, more precisely, the concepts which the terms represent). A major limitation of non-semantic IR approaches is that they cannot handle queries which either require knowledge and data which are not available in the documents; or require extraction, explicit structuring, and reasoning about some data. Consider for example, the following query:

> *"telecom company" Europe "John Smith" director*

The information need appears to be for documents concerning a telecom company in Europe, a person called John Smith, and a board appointment. Note, however, that a document containing the following sentence would not be returned using conventional search techniques:

> *"At its meeting on the 10th of May, the board of London-based O2 appointed John Smith as CTO"*

In order to be able to return this document, the search engine would need to be aware of the following semantic relations:

1. O2 is a mobile operator, which is a kind of telecom company;
2. London is located in the UK, which is a part of Europe;
3. A CTO is a kind of director.

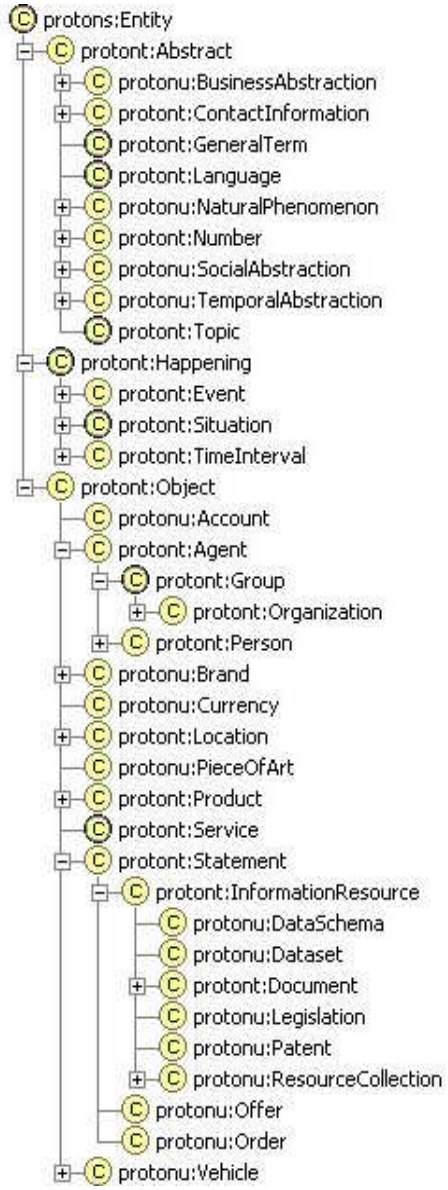

Figure 3.1: A view of the top part of the PROTON class hierarchy

These are precisely the kinds of relations which need to be represented and reasoned over in order to enable conceptual searching.

3.2. Ontologies

Formal knowledge representation (KR) is about building models[1] of the world (of a particular state of affairs, situation, domain or problem), which allow for automatic reasoning and interpretation. Such formal models are called *ontologies,* whenever they (are intended to) represent a shared conceptualization (e.g. a basic theory, a schema, or a classification). Ontologies can be used to provide formal semantics (i.e. machine-interpretable meaning) to any sort of information: databases, catalogues,

[1] The typical modelling paradigm is mathematical logic, but there are also other approaches, rooted in information and library science. KR is a very broad term; here we only refer to one of its main streams.

documents, web pages, etc. Ontologies can be used as semantic frameworks: the association of information with ontologies makes such information much more amenable to machine processing and interpretation. This is because formal ontologies are represented in logical formalisms, such as OWL, [Dean et al. 2004], which allow automatic inferencing over them and over datasets aligned to them. An important role of ontologies is to serve as schemata or "intelligent" views over information resources[2]. Thus they can be used for indexing, querying, and reference purposes over non-ontological datasets and systems, such as databases, document and catalogue management systems. Because ontological languages have a formal semantics, ontologies allow a wider interpretation of data, i.e. inference of facts which are not explicitly stated. In this way, they can improve the interoperability and the efficiency of the usage of arbitrary datasets.

An ontology can be characterized as comprising a 4-tuple[3]:

O = <C,R,I,A>

Where C is a set of **classes** representing *concepts* we wish to reason about in the given domain (invoices, payments, products, prices,…); R is a set of **relations** (also referred to as **properties** or **predicates**) holding between (instances of) those classes (`Product` *`hasPrice`* `Price`); I is a set of **instances**, where each instance can be an instance of one or more classes and can be linked to other instances or to **literal** values (strings, numbers, …) by relations (*`product23`* `compatibleWith` `product348`; *`product23`* `hasPrice` `€170`); A is a set of **axioms** (if a product has a price greater than €200, then shipping is free).

The ontologies can be classified as *light-weight* or *heavy-weight* according to the complexity of the KR language used. Light-weight ontologies allow for more efficient and scalable reasoning, but do not possess the high predictive (or restrictive) power of the full-bodied concept definitions of heavy-weight ontologies. The ontologies can be further differentiated according to the sort of conceptualization that they formalize: *upper-level* ontologies model general knowledge, while *domain-* and *application-ontologies* represent knowledge about a specific domain (e.g. medicine or sport) or a type of applications (e.g. knowledge management systems). Basic definitions regarding ontologies can be found in [Gruber 1992, 1993] and [Guarino 1995, 1998].

Finally, ontologies can be distinguished according to the sort of semantics being modelled and their intended usage. The major categories from this perspective are:

1. *Schema-ontologies*: ontologies which are close in purpose and nature to database and object-oriented schemata. They define classes of objects, their appropriate attributes and relationships to objects of other classes. A typical usage of such an ontology is that large sets of instances of the classes are defined and managed. Intuitively, a class in a schema ontology corresponds to a table in an RDBMS; a relation – to a column; an instance – to a row in the table for the corresponding class.
2. *Topic-ontologies*: taxonomies which define hierarchies of topics, subjects,

[2] Comments in the same spirit are provided in (Gruber 1992) also. This is also the role of ontologies on the semantic web..

[3] A more formal and extensive mathematical definition of an ontology is given, for example, in [Ehrig et al., 2005]. The characterization offered here is suitable for the purposes of our discussion, however.

categories, or designators. These have a wide range of applications related to classification of different things (entities, information resources, files, web-pages, etc.) The most popular examples are library classification systems and taxonomies, which are widely used in the KM field. Yahoo and DMoz[4] are popular large scale incarnations of this approach in the context of the web. A number of the most popular taxonomies are listed as encoding schemata in Dublin Core [DCMI 2005].

3. *Lexical ontologies*: lexicons with formal semantics, which define lexical concepts[5], word-senses and terms. These can be considered as semantic thesaurus or dictionaries. The concepts defined in such ontologies are not instantiated, rather they are directly used as reference, e.g. for annotation of the corresponding terms in text. WordNet is the most popular general purpose (i.e. upper-level) lexical ontology.

This documents is mostly concentrated on annotation, indexing and retrieval with respect to schema-ontologies and KBs built with respect to them.

PROTON [Terziev 2005] is a light-weight upper-level schema-ontology developed in the scope of the SEKT project [Davies et al. 2005]. It is used in the KIM system (see section 4.2.4.4) for semantic annotation, indexing and retrieval. We will also use it for ontology-related examples within this section. PROTON is encoded in OWL Lite (see sub-section 3.4.1) and defines about 300 classes and 100 properties, providing good coverage of named entity types and concrete domains (i.e. modelling of concepts such as people, organizations, locations, numbers, dates, addresses, etc.) A snapshot of the PROTON class hierarchy is given in Figure 3.1.

3.3. Knowledge Bases and Semantic Repositories

Knowledge base (KB) is a broader term than ontology. Similarly to an ontology, a KB is represented in a KR formalism, which allows automatic inference. It could include multiple axioms, definitions, rules, facts, statements, and any other primitives. In contrast to ontologies, however, KBs are not intended to represent a shared or consensual conceptualization. Thus, ontologies are a specific sort of KB. Many KBs can be split into ontology and instance data parts, in a way analogous to the splitting of schemata and concrete data in databases. A broader discussion on the different terms related to ontology and semantics can be found in section 3 of (Kiryakov 2006).

Semantic repositories[6] allow for storage, querying, and management of structured data with respect to formal semantics; in other words, they provide KB management infrastructure. Semantic repositories can serve as a replacement for database management systems (DBMS), offering easier integration of diverse data and more analytical power. In a nutshell, a semantic repository can dynamically interpret metadata schemata and ontologies, which determine the structure and the

[4] http://www.yahoo.com and http://www.dmoz.org respectively.

[5] We use 'lexical concept' here as some kind of a formal representation of the meaning of a word or a phrase. In Wordnet, for example, lexical concepts are modelled as synsets (synonym sets), while word-sense is the relation between a word and a synset.

[6] "Semantic repository" is not a well-established term. A more elaborate introduction can be found at http://www.ontotext.com/inference/semantic_repository.html.

semantics of data and of queries against that data.

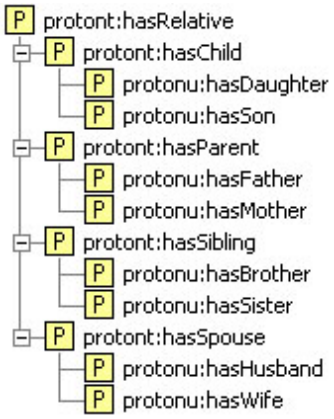

Figure 3.2: A Hierarchy of Family Relationships

Compared to the approach taken in relational DBMSs, this allows for (i) easier changes to and combinations of data schemata and (ii) automated interpretation of the data. As an example, let us imagine a typical database populated with the information that John is a son of Mary. It will be able to "answer" just a couple of questions: *Who are the son(s) of Mary?* and *Of whom is John the son?* Given simple family-relationships ontology (as the one in PROTON, see Figure 3.2), a semantic repository could handle much bigger set of questions. It will be able infer the more general fact that John is a child of Mary (because `hasSon` is a sub-property of `hasChild`) and, even more generally, that Mary and John are relatives (which is true in both directions, because `hasRelative` is defined to be symmetric in the ontology). Further, if it is known that Mary is a woman, a semantic repository will infer that Mary is the mother of John, which is a more specific inverse relation. Although simple for a human to infer, the above facts would remain unknown to a typical DBMS and indeed to any other information system, for which the model of the world is limited to data-structures of strings and numbers with no automatically interpretable semantics.

3.4. The Semantic Web

Research in the Semantic Web has already produced a variety of tools, standards and best practices which are of real benefit when working toward implementing conceptual search. This section gives a brief presentation of the Semantic Web and associated ontology representation standards.

> "The Semantic Web is a web of data. There is lots of data we all use every day, and its not part of the web. I can see my bank statements on the web, and my photographs, and I can see my appointments in a calendar. But can I see my photos in a calendar to see what I was doing when I took them? Can I see bank statement lines in a calendar?
>
> Why not? Because we don't have a web of data. Because data is controlled by applications, and each application keeps it to itself.
>
> The Semantic Web is about two things. It is about common formats for interchange of data, where on the original Web we only had interchange of documents. Also it is about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing."
>
> [from http://www.w3.org/2001/sw/]

3.4.1. Basic Semantic Web Standards: RDF(S) and OWL

A family of mark-up and KR standards were developed, under W3C-driven community processes, as a basis for the Semantic Web. RDF, [Klyne and Carroll 2004], is a metadata representation language, which serves as a basic data-model for the Semantic Web. It allows resources to be described through relationships to

other resources and literals. The resources are defined through URIs (unified resource identifiers, as in XML; e.g. URL). The notion of resource is virtually

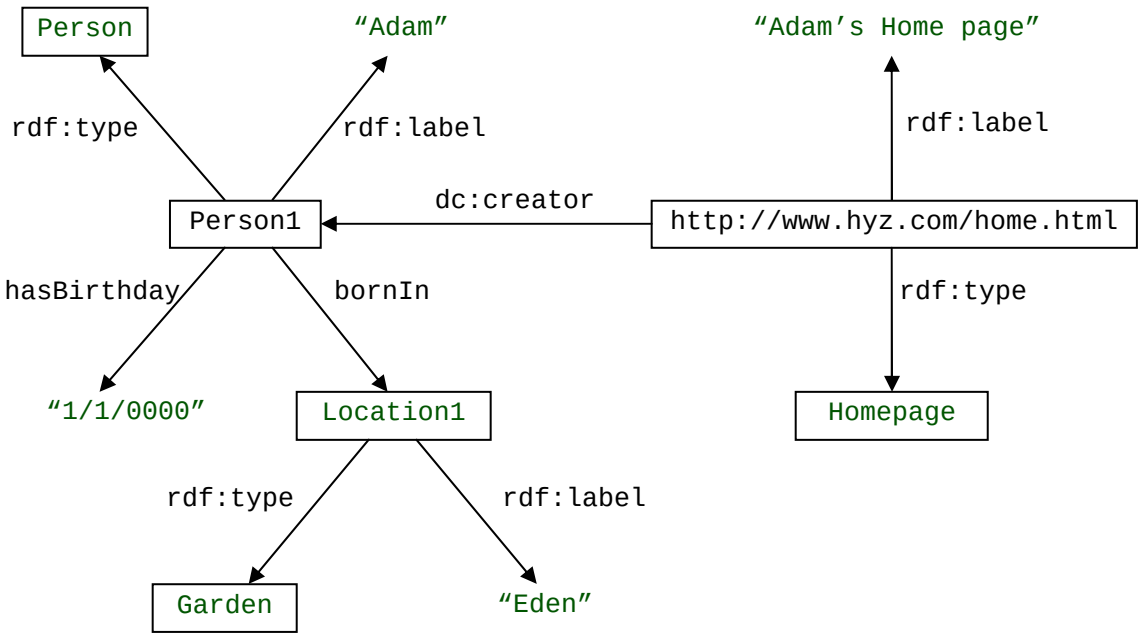

Figure 3.3: RDF Graph Describing Adam and His Home Page

unrestricted; anything can be considered as a resource and described in RDF: from a web page or a picture published on a web to concrete entities in the real world (e.g. people, organisations) or abstract notions (e.g. the number Pi and the musical genre Jazz). Literals (again as in XML) are any concrete data values e.g. strings, dates, numbers, etc. The main modelling block in RDF is the statement – a triple `<Subject, Predicate, Object>`, where:

1. `Subject` is the resource, which is being described;
2. `Predicate` is a resource, which determines the type of the relationship;
3. `Object` is a resource or a literal, which represents the “value” of the attribute.

A set of RDF triples can be seen as a graph, where resources and literals are nodes and each statement is represented by a labelled arc (the Predicate or relation),

directed from the Subject to the Object. So-called blank nodes can also appear in the graph, representing unique anonymous resources, used as auxiliary nodes. A sample graph, which describes a web page, created by a person called Adam, can be seen in Figure 3.3.

Resources can belong to (formally, be *instances* of) classes – this can be expressed as a statement through the `rdf:type` system property as follows: `<resource, rdf:type, class>`. Two of the system classes in RDF are `rdfs:Class` and `rdf:Property`. The instances of `rdf:Class` are resources which represent classes, i.e. those resources which can have other resources as instances. The instances of `rdf:Property` are resources which can be used as predicates (relations) in triple statements.

The most popular format for encoding RDF is its XML syntax, [Becket 2004]. However, RDF can also be encoded in a variety of other syntaxes. The main difference between XML and RDF is that the underlying model of XML is a tree of nested elements, which is rather different from the graph of resources and literals in RDF.

RDFS, [Brickley and Guha 2000], is a schema language, which allows for definition of new classes and properties. OWL, [Dean et al. 2004], is an ontology language, which extends RDF(S)[7] with means for more comprehensive ontology definitions. OWL has three dialects: OWL-Lite, OWL-DL, and OWL-Full. Owl-Lite is the least expressive of these dialects but the most amenable to efficient reasoning. Conversely, OWL-Full provides maximal expressivity but is undecidable[8]. OWL-DL can be seen as a decidable sub-language inspired by the so-called description logics. These dialects are nested such that every OWL-Lite ontology is a legal OWL-DL ontology and every OWL-DL ontology is a legal OWL-Full ontology.

Below we briefly present the principal modelling primitives typically used in SW applications; this comprises most of the RDFS constructs and few of the simplest ones from OWL:

- All resources, including classes and properties, may have titles (literals[9], linked through property `rdf:label`) and descriptions or glosses (literals linked through `rdf:comment`);
- Classes can be defined as sub-classes, i.e. specializations, of other classes (via `rdf:subClassOf`). This means that all instances of the class are also instances of its super class. For example, in PROTON `City` is a sub-class of `Location`.
- In OWL, properties are distinguished into object- and data-properties (instances respectively of `owl:ObjectProperty` and `owl:DataProperty`). The object-properties are binary relationships, relating entities to other entities. The data-properties can be considered as attributes – they relate entities to literals.
- Domains and ranges of properties can be defined. A domain (`rdfs:domain`) specifies the classes of entities to which this property is applicable. A range (`rdfs:range`) specifies the classes of entities (for object-properties) or data-types of the literal values (in case of data-properties), which can serve as objects in statements predicated by this property. For instance, the property `hasSister` might typically have the class `Person` as its domain and `Woman` as its range. Whenever multiple classes are provided as domain or range for a single property, the intersection of those classes is used.
- Properties can be defined as sub-properties, i.e. specializations of other properties (via `rdf:subPropertyOf`). Imagine that there are two properties, `p1` and `p2`, for which `<p1,subPropertyOf,p2>`. The formal meaning of this statement is that for all pairs for which `p1` takes place, i.e. `<x,p1,y>`, `p2` also

[7] RDF(S) is a short name for the combination of RDF and RDFS.

[8] An undecidable logical language is one for which it is a theoretical impossibility to build a reasoner which can prove all the valid inferences from any theory expressed in that language.

[9] Recall that literals are values such as strings or numbers.

takes place, i.e. <x,p2,y> is also true. The hierarchy of family relationships discussed above (see Figure 3.2) provides a number of intuitive examples of sub-properties.

- Properties can be defined as a symmetric (via owl:SymmtericProperty) and transitive (via owl:TransitiveProperty) ones. If p1 is a symmetric property than whenever <x,p1,y> is true, <y,p1,x> is also true. If p2 is a transitive property and <x,p2,y> and <y,p2,z> are true, it can be concluded that <x,p2,z> is also true. hasRelative is an example of a property which is both symmetric and transitive.
- Object-properties can be defined to be inverse to each other (via owl:inverseOf). This means that if <p1,inverseOf,p2> then, whenever <x,p1,y> holds, <y,p2,x> can be inferred and vice versa. An obvious example is <hasChild, owl:inverseOf, hasParent>.

4. Metadata and Annotations

Metadata is a term of wide and sometimes controversial or misleading usage. From its etymology, metadata is "data about data". Thus, metadata is a role that certain (pieces of) data could play with respect to other data. Such an example could be a particular specification of the author of a document, provided independently from the content of the document, say, according to a standard like Dublin Core (DC), [DCMI 2005]. RDF has been introduced as a simple language that is to be used for the assignment of semantic descriptions to information resources on the web. Therefore an RDF description of a web page represents metadata. However, an RDF description of a person, independent from any particular documents (e.g., as a part of an RDF(S)-encoded dataset), is not metadata – this is data about a person, not about other data. In this case, RDF(S) is used as a KR language. Finally, the RDFS definition of the class Person, will typically be part of an ontology, which can be used to structure datasets and metadata, but which is again not a piece of metadata itself.

A term which is often used as a synonym for metadata, particularly in the natural language processing (NLP) community, is *annotation.* In this section, we discuss annotation of documents in general, while the next section presents a discussion of "semantic annotation" which can be seen as a way of generating information that is essential for enabling conceptual searching.

Annotations on text documents can be distinguished into two groups according to their scope:

1. *Document-level annotations*, which refer to the whole document. Such examples are the DC elements (Title, Subject, Creator, etc.);
2. *Character-level annotations*, which refer to a fragment of a document, determined by start and end characters. An example might be a comment attached to a particular part of a document. Character-level annotations are usually meant when the term "annotation" is used for text documents without further clarification.

It is worth mentioning that *hyperlinks* can be considered as a specific sort of character-level annotation, when the metadata is, essentially, a reference to another document or part of document.

Further, annotations can also be distinguished with respect to the way in which they

are attached to the text. The basic choices here are:

3. *Embedded markup*, when the annotations are incorporated within the document. In this case the metadata is bundled together with the data. Examples are markup languages such as HTML, where the annotations are specified though pairs of start and end tags, e.g. `abc<tag>de</tag>gf`. When document-level annotations have to be represented this way, they are sometimes attached in a special section at the start or end of the document – one example is the `<head>` section in the HTML files. An example of character-level embedded annotations is a footnote.
4. *Standoff references*, when the annotations are maintained separately from the document to which they refer. In the case of character-level annotations, the reference should also specify the specific part of the document. One approach for this is based on position, e.g. offset and length; another possibility is the usage of some sort of anchoring and linking mechanism. An example of specification based on standoff annotations is TIPSTER, [Grishman 1997].

The different types of annotation are shown diagrammatically in Figure 4.1. In his thesis on architectures for language engineering Cunningham [1999], provides an overview of various annotation models and discusses their advantages and disadvantages in the context of text processing systems and applications. Similar analysis, but in the context of open hypermedia systems (OHS), can be found in [van Ossenbruggen et al. 2002]. Here we will only briefly mention few of the main characteristics of the embedded and the standoff models:

- Embedded markup is not applicable in cases when the author of the metadata has no write-permission to the document;
- Standoff annotations may become inconsistent in the event of change to the document to which they refer;
- Access to embedded annotations requires processing (e.g. parsing) of the

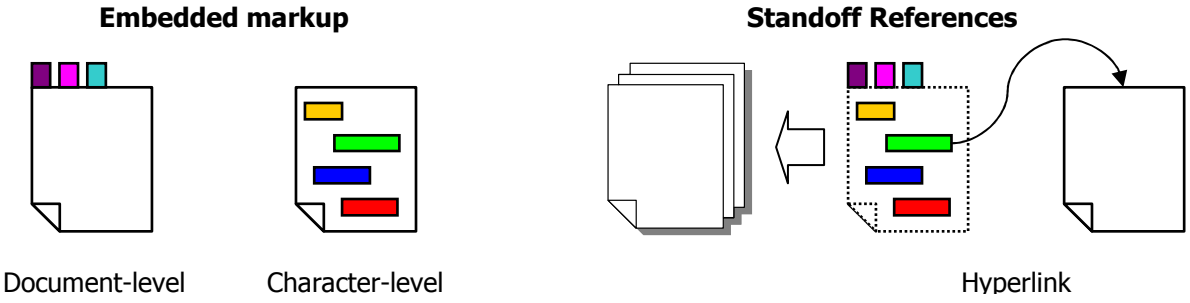

Figure 4.1: Types of Annotation

documents. Thus, such annotations are not appropriate for applications where random (non-sequential) access to the annotations is important. Conversely, standoff annotations can be maintained and queried efficiently in structured form (e.g. in a database);

- Embedded annotations are simpler to encode, read and manage when the volume of the markup is relatively small. However, they are inappropriate when the volume of the markup becomes comparable to or bigger than that of the text itself;

- Tagging mechanisms based on embedded annotations have difficulties in handling overlapping (as opposed to nested) annotations;
- Embedded annotations should always be distributed together with the document, which can cause IPR issues, unnecessary redundancy or conflict when multiple sets of annotations are available for one and the same document.

4.1. Semantic Annotation

If we abstract the current Web away from the transport, content type, and content formatting aspects, it could be regarded as a set of documents with some limited metadata, attached to them (document-level annotations about title, keywords, etc.), and with hyperlinks between the documents (see the left-hand side part of Figure 4.2).

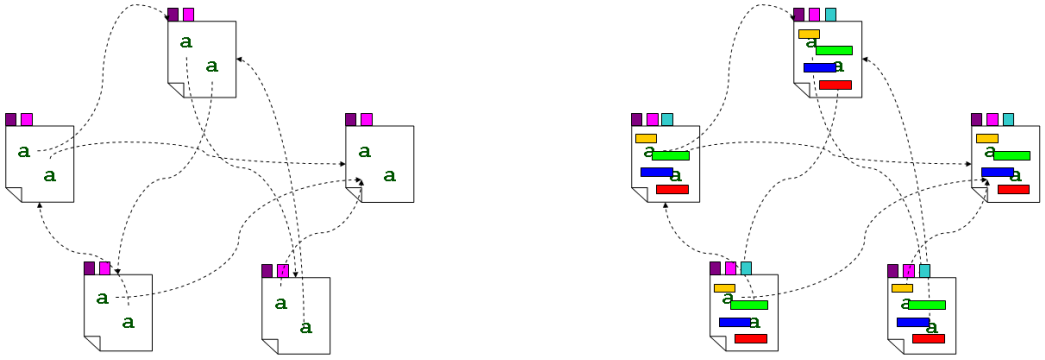

Figure 4.2: The Current WWW (left) and the Semantic Web (right)

What does the Semantic Web add to this picture? – Essentially, *semantic* metadata of different kinds, both on the document- and the character-level. Figure 4.2 compares links on the current web to those on the semantic web. Typically, the Semantic Web has a greater number of more meaningful annotations, as compared to the current WWW. Many of those annotations represent links to external knowledge, which constitute a new sort of connectivity that is not presented on Figure 4.2, but is extensively discussed in this section (Figure 4.3 and Figure 4.4).

In order to uncover the added value of the Semantic Web, it is crucial to elaborate a little further regarding the nature of semantic metadata. Suppose, we add a tag `<2134>` to some portion of a document as follows "… `Abc <2134>xyz</2134>` …". Is this metadata useful? Can we call it semantic? – Without further assumptions, the answers are negative. In order to have metadata useful in a Semantic Web context, it should mean something, i.e. the symbols (or expressions or references) that constitute it should allow further interpretation. Interpretation in this context means allowing the assigning of some additional information to the symbols. It is important to realize that interpretation is only possible with respect to something; to some domain, model, context, (possible) world. This is the domain that (the interpretations of) the symbols are "about." Obviously, annotations in RDF(S), OWL, or some other language refer to a model of the world. Annotations can be expressed in RDF(S), but they are not *about* RDF(S), as depicted in Figure 4.4.

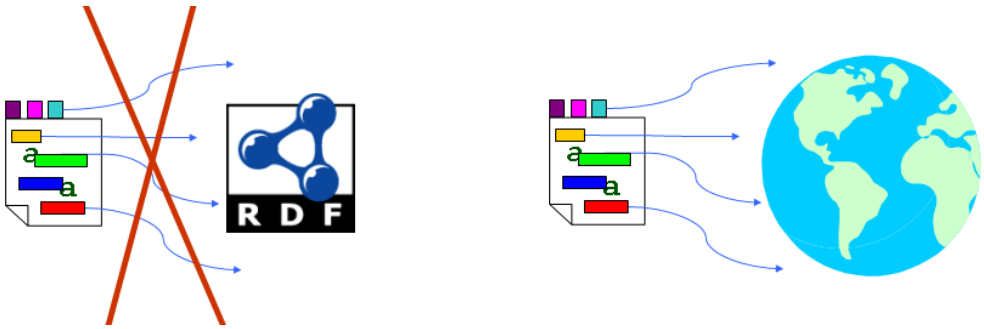

Figure 4.3: Metadata about the World, not about RDF

Further, the metadata can hardly refer to (or be interpreted directly with respect to) the world. Such references cannot be formal and unambiguous. What the semantic metadata can be expected to refer to directly is a KB, a formal model of (some aspect of) the world, as depicted in Figure 4.4. Such a KB specifies some world knowledge which serves as a semantic link from the metadata to the world. Note, that in the Semantic Web context such a KB can be as scattered and heterogeneous as the metadata is. Guha and McCool, [2003], consider this KB itself to be the Semantic Web: "the Semantic Web is not a web of documents, but a web of relations between resources, denoting real world objects". In our view the Semantic Web is the combination of the KB and the semantic annotations referring to it (not just the KB).

For automatic processability, the interpretations of metadata should be performed automatically by machines in strict and predictable fashion. This requires a formal definition of how the metadata should be interpreted and, because of this, a formal definition of the context. Assuming that one and the same context can be modelled in different ways, allowing different (and potentially ambiguous) interpretations, what has to be specified is the conceptualization – as defined in [Gruber 1993]: "a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose." This is where ontologies are used to act as logical theories for the "formal specification of a conceptualization" (again in [Gruber 1993], see also section 3.2.)

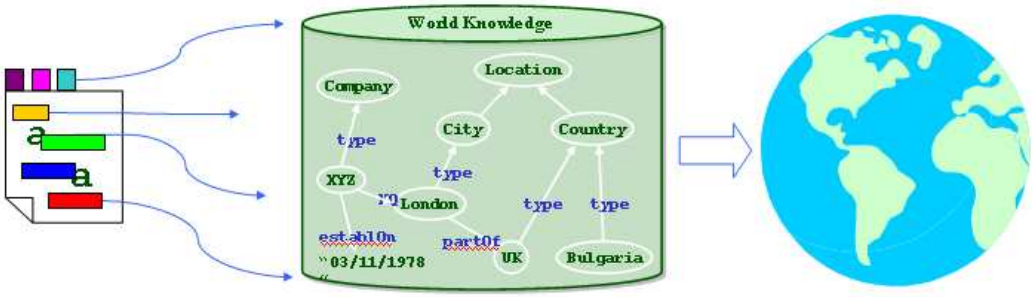

Figure 4.4: Metadata Referring to World Knowledge

Although the above discussion is informally presented, we consider it rather important for the realization of the Semantic Web. It is the intuition of the authors that the KR and modelling issues related to the development or generation of useful semantic annotations require more specific attention. RDF(S) and OWL are designed to serve well for data modelling in as diverse and heterogeneous environments as possible. Thus, they provide very little modelling guidance and constraints. For instance, an RDF(S) annotation of an HTML page can include at the

same time a definition of the class `Person` (which is a piece of ontological knowledge), the description of a specific person Mr. X (which should normally be world knowledge, part of a KB) and the fact that this person is an author of the webpage (which is the only piece of actual metadata describing the web page). We believe that the development of real world Semantic Web applications require some concrete knowledge modelling commitments to be made and the corresponding design and representation principles to be set out.

Semantic annotation is a specific metadata generation and usage schema aiming to enable new information access methods and to enhance existing ones. The annotation scheme offered here is based on the understanding that the information discovered in the documents by an IE system constitute an important part of their semantics. Moreover, by using text redundancy and external or background knowledge, this information can be connected to formal descriptions, i.e., ontologies, and thus provide semantics and connectivity to the web.

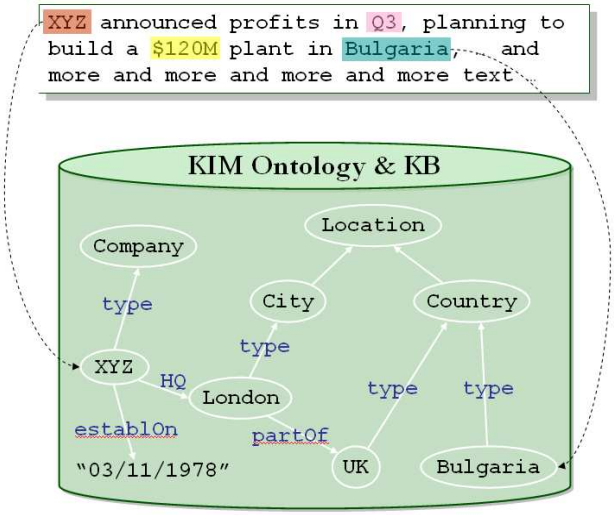

Figure 4.5: Semantic Annotation

The task of realising the vision of the Semantic Web will be much helped, if the following basic tasks can be properly defined and solved:

1. Formally annotate and hyperlink (references to) entities and relations in textual (parts of) documents;
2. Index and retrieve documents with respect to entities/relations referred to.

The first task could be seen as a combination of a basic press-clipping exercise, a typical IE task, and automatic hyper-linking. The resulting annotations represent a method for document enrichment and presentation, the results of which can be further used to enable other access methods.

The second task is just a modification of the classical IR task – documents are retrieved on the basis of relevance to entities or relations instead of words. However the basic assumption is quite similar – a document is characterised by the bag of tokens constituting its content, disregarding its structure. While the basic IR approach considers word stems as tokens, there has been considerable effort in the last decade towards using word-senses or lexical concepts (see [Mahesh 99] and [Voorhees 98]) for indexing and retrieval. Similarly, entities and relations can be seen as a special sort of a token to be indexed and retrieved.

In a nutshell, Semantic Annotation is about assigning to entities and relations in the text links to their semantic descriptions in an ontology (as shown in Figure 4.5). This sort of semantic metadata provides both class and instance information about the entities/relations.

Most importantly, automatic semantic annotation enables many new applications: highlighting, semantic search, categorisation, generation of more advanced metadata, smooth traversal between unstructured text and formal knowledge. Semantic annotation is applicable to any kind of content – web pages, regular (non-web) documents, text fields in databases, video, audio, etc.

4.2. Ontology-Based Information Extraction

Ontology-Based IE (OBIE) is the technology used for semantic annotation. One of the important differences between traditional IE and OBIE is the use of a formal ontology as one of the system's resources. OBIE may also involve reasoning.

Another substantial difference of the semantic IE process from the traditional one is the fact that it not only finds the (most specific) type of the extracted entity, but it also identifies it, by linking it to its semantic description in the instance base. This allows entities to be traced across documents and their descriptions to be enriched through the IE process. When compared to the "traditional", the first stage corresponds to the Named Entities Extraction (NE) task and the second stage corresponds to the Coreference (CO) task. Given the lower performance achievable on the CO task, semantic IE is in general a much harder task.

OBIE poses two main challenges:

- the identification of instances from the ontology in the text
- the automatic population of ontologies with new instances in the text

4.2.1. Identification of instances from the ontology

If an ontology is already populated with instances, the task of an OBIE system may be simply to identify instances from the ontology in the text. Similar methodologies can be used for this as for traditional IE systems, using an ontology rather than a flat gazetteer. For rule-based systems, this is relatively straightforward. For learning-based systems, however, this is more problematic because training data is required. Collecting such training data is, however, likely to be a large bottleneck. Unlike traditional IE systems for which training data exists in domains like news texts in plentiful form, thanks to efforts from MUC, ACE [ACE 04] and other collaborative and/or competitive programs, there is a dearth of material currently available for semantic web applications. New training data needs to be created manually or semi-automatically, which is a time-consuming and onerous task, although systems to aid such metadata creation are currently being developed.

4.2.2. Automatic ontology population

In this task, an OBIE application identifies instances in the text belonging to concepts in a given ontology, and adds these instances to the ontology in the correct location. It is important to note that instances may appear in more than one location in the ontology, because of the multidimensional nature of many ontologies and/or ambiguity which cannot or should not be resolved at this level (see e.g. [Felber 84, Bowker 95] for a discussion).

4.2.3. Applying "Traditional" IE in Semantic Web Applications

In this section we give a brief overview of some current state-of-the-art systems which apply traditional IE techniques for semantic web applications such as annotating web pages with metadata. Unlike ontology-based IE applications, these do not incorporate ontologies into the system, but either use ontologies as a bridge between the IE system and the final annotation (as with AERODAML) or rely on the user to provide the relevant information through manual annotation (as with

the Amilcare-based tools).

4.2.3.1. AeroDAML

AeroDAML [Kogut and Holmes 01] is an annotation tool created by Lockheed Martin which applies IE techniques to automatically generate DAML annotations from web pages. The aim is to provide naive users with a simple tool to create basic annotations without having to learn about ontologies, in order to reduce time and effort and to encourage people to semantically annotate their documents. Aero-DAML links most proper nouns and common types of relations with classes and properties in a DAML ontology.

There are two versions of the tool: a web-enabled version which uses a default generic ontology, and a client-server version which supports customised ontologies. In both cases, the user enters a URI (for the former) and a filename (for the latter) and the system returns the DAML annotation for the webpage or document. It provides a drag-and-drop tool to create static (manual) ontology mappings, and also includes some mappings to predefined ontologies.

AeroDAML consists of the AeroText IE system, together with components for DAML generation. A default ontology which directly correlates to the linguistic knowledge base used by the extraction process is used to translate the extraction results into a corresponding RDF model that uses the DAML+OIL syntax. This RDF model is then serialised to produce the final DAML annotation. The AeroDAML ontology is comprised of two layers: a base layer comprising the common

knowledge base of AeroText, and an upper layer based on WordNet. AeroDAML can generate annotations consisting of instances of classes such as common nouns and proper nouns, and properties, of types such as coreference, Organisation to Location, Person to Organization.

4.2.3.2. Amilcare

Amilcare [Ciravegna and Wilks 03] is an IE system which has been integrated in several different annotation tools for the Semantic Web. It uses machine learning (ML) to learn to adapt to new domains and applications using only a set of annotated texts (training data). It has been adapted for use in the Semantic Web by simply monitoring the kinds of annotations produced by the user in training, and learning how to reproduce them. The traditional version of Amilcare adds XML annotations to documents (inline markup); the Semantic Web version leaves the original text unchanged and produces the extracted information as triples of the form <annotation, startPosition, endPosition> (standoff markup). This means that it is left to the annotation tool and not the IE system to decide on the format of the ultimate annotations produced.

In the Semantic Web version, no knowledge of IE is necessary; the user must simply define a set of annotations, which may be organised as an ontology where annotations are associated with concepts and relations. The user then manually annotates the text using some interface connected to Amilcare, as described in the following systems. Amilcare works by preprocessing the texts using GATE's IE system ANNIE [Cunningham et al. 02], and then uses a supervised machine learning algorithm to induce rules from the training data.

4.2.3.3. MnM

MnM [Motta et al. 02] is a semantic annotation tool which provides support for annotating web pages with semantic metadata. This support is semi-automatic, in that the user must provide some initial training information by manually annotating documents before the IE system (Amilcare) can take over. It integrates a web browser, an ontology editor, and tools for IE, and has been described as "*an early example of next-generation ontology editors*" [Motta et al. 02], because it is web-based and provides facilities for large-scale semantic annotation of web pages.

It aims to provide a simple system to perform knowledge extraction tasks at a semi-automatic level.

There are 5 main steps to the procedure:

- the user browses the web
- the user manually annotates his chosen web pages
- the system learns annotation rules
- the system tests the rules learnt
- the system takes over automatic annotation, and populate ontologies with the instances found. The ontology population process is semi-automatic and may require intervention form the user.

4.2.3.4. S-CREAM

S-CREAM (Semi-automatic CREAtion of Metadata) [Handschuh et al. 02] is a tool which provides a mechanism for automatically annotating texts, given a set of training data which must be manually created by the user. It uses a combination of two tools: Onto-O-Mat, a manual annotation tool which implements the CREAM framework for creating relational metadata [Handschuh et al. 01], and Amilcare.

As with MnM, S-CREAM is trainable for different domains, provided that the user creates the necessary training data. It essentially works by aligning conceptual markup (which defines relational metadata) provided by Ont-O-Mat with semantic markup provided by Amilcare. This problem is not trivial because the two representations may be very different. Relational metadata may provide information about relationships between instances of classes, for example that a certain hotel is located in a certain city. S-CREAM thus supports metadata creation with the help of a traditional IE system, and also provides other functionalities such as web crawler,

document management system, and a meta-ontology.

4.2.3.5. Discussion

One of the problems with these annotation tools is that they do not provide the user with a way to customise the integrated language technology directly. While many users would not need or want such customisation facilities, users who already have ontologies with rich instance data will benefit if they can make this data available to the IE components. However, this is not possible when “traditional” IE methods like Amilcare are used, because they are not aware of the existence of the user’s ontology.

The more serious problem however, as discussed in the S-CREAM system [Handschuh et al. 02], is that there is often a gap between the annotations and their types produced by IE and the classes and properties in the user’s ontology. The proposed solution is to write some kind of rules, such as logical rules, to achieve this. For example, an IE system would typically annotate London and UK as locations, but extra rules are needed to specify that there is a containment relationship between the two (for other examples see [Handschuh et al. 02]). However, rule writing of the proposed kind is too difficult for most users and a new solution is needed to bridge this gap.

Ontology-Based IE systems for semantic annotation, to be discussed next, address both problems:

- The ontology is used as a resource during the IE process and therefore it can benefit from existing data such as names of customers from a billing database.
- Instance disambiguation is performed as part of the semantic annotation process, thus removing the need for user-written rules.

4.2.4. Ontology-Based IE

The following are systems which are based on IE technology that is ontology-aware.

4.2.4.1. Magpie

Magpie [Domingue et al. 04] is a suite of tools which supports the interpretation of webpages and ”collaborative sense-making”. It annotates webpages with metadata in a fully automatic fashion and needs no manual intervention by matching the text against instances in the ontology. It automatically populates an ontology from relevant web sources, and can be used with different ontologies. The principle behind it is that it uses an ontology to provide a very specific and personalised viewpoint of the webpages the user wishes to browse. This is important because different users often

have different degrees of knowledge and/or familiarity with the information presented, and have different browsing needs and objectives.

Magpie’s main limitation is that it does not perform automatic population of the ontology with new instances, i.e., it is restricted only to matching mentions of already existing instances.

4.2.4.2. PANKOW

The PANKOW system (Pattern-based Annotation through Knowledge on the Web) [Cimiano et al. 04] exploits surface patterns and the redundancy on the Web to categorise automatically instances from text with respect to a given ontology. The

patterns are phrases like: the <INSTANCE> <CONCEPT> (e.g., the Ritz hotel) and <INSTANCE> is a <CONCEPT> (e.g., Novotel is a hotel). The system constructs patterns by identifying all proper names in the text (using a part-of-speech tagger) and combining each one of them with each of the 58 concepts from their tourism ontology into a hypothesis. Each hypothesis is then checked against the Web via Google queries and the number of hits is used as a measure of the likelihood of this pattern being correct.

The system's best performance on this task in fully automatic mode is 24.9% while the human performance is 62.09%. However, when the system is used in semi-automatic mode, i.e., it suggests the top five most likely concepts and the user chooses among them, then the performance goes up to 49.56%.

The advantages of this approach are that it does not require any text processing (apart from POS tagging) or any training data. All the information comes from the web. However, this is also a major disadvantage because the method does not compare the context in which the proper name occurs in the document to the contexts in which it occurs on the Web, thus making it hard to classify instances with the same name that belong to different classes in different contexts (e.g., Niger can be a river, state, country, etc.). On the other hand, while IE systems are more costly to set up, they can take context into account when classifying proper names.

4.2.4.3. SemTag

The SemTag system [Dill et al. 03] performs large-scale semantic annotation with respect to the TAP ontology[10]. It first performs a lookup phase annotating all possible mentions of instances from the TAP ontology. In the second, disambiguation phase, SemTag uses a vector-space model to assign the correct ontological class or determine that this mention does not correspond to a class in TAP. The disambiguation is carried out by comparing the context of the current mention with the contexts of instances in TAP with compatible aliases, using a window of 10 words either side of the mention.

The TAP ontology, which contains about 65,000 instances, is very similar in size and structure to the KIM Ontology and instance base discussed in section 5.5. (e.g. each instance has a number of lexical aliases). One important characteristic of both ontologies is that they are very light-weight and encode only essential properties of concepts and instances. In other words, the goal is to cover frequent, commonly-known and searched for instances (e.g., capital cities, names of presidents), rather than to encode an extensive set of axioms enabling deep, Cyc-style reasoning. As reported in [Mahesh et al. 96], the heavy-weight logical approach undertaken in Cyc is not appropriate for many NLP tasks.

The SemTag system is based on a high-performance parallel architecture -Seeker, where each node annotates about 200 documents per second. The demand for such parallelism comes from the big volumes of data that need to be dealt with in many applications and make automatic semantic annotation the only feasible option. A parallel architecture of a similar kind is currently under development for KIM and, in general, it is an important ingredient of large-scale automatic annotation approaches.

[10] http://tap.stanford.edu/tap/papers.html

4.2.4.4. KIM Platform

The Knowledge and Information Management system (KIM) is a product of OntoText Lab [Kiryakov et al 05]. KIM is an extensible platform for semantics-based knowledge management which offers IE-based facilities for metadata creation, storage, and conceptual search. The system has a server-based core that performs ontology-based IE and stores results in a central knowledge base. This server platform can then be used by diverse applications as a service for annotating and querying document spaces.

The ontology-based Information Extraction in KIM produces annotations linked both to the ontological class and to the exact individual in the instance base. For new (previously unknown) entities, new identifiers are allocated and assigned; then minimal descriptions are added to the semantic repository. The annotations are kept separately from the content, and an API for their management is provided.

The instance base of KIM has been pre-populated with 200,000 entities of general importance that occur frequently in documents. The majority are different kinds of locations: continents, countries, cities, etc. Each location has geographic coordinates and several aliases (usually including English, French, Spanish, and sometimes the local transcription of the location name) as well as co-positioning relations (e.g. subRegionOf.).

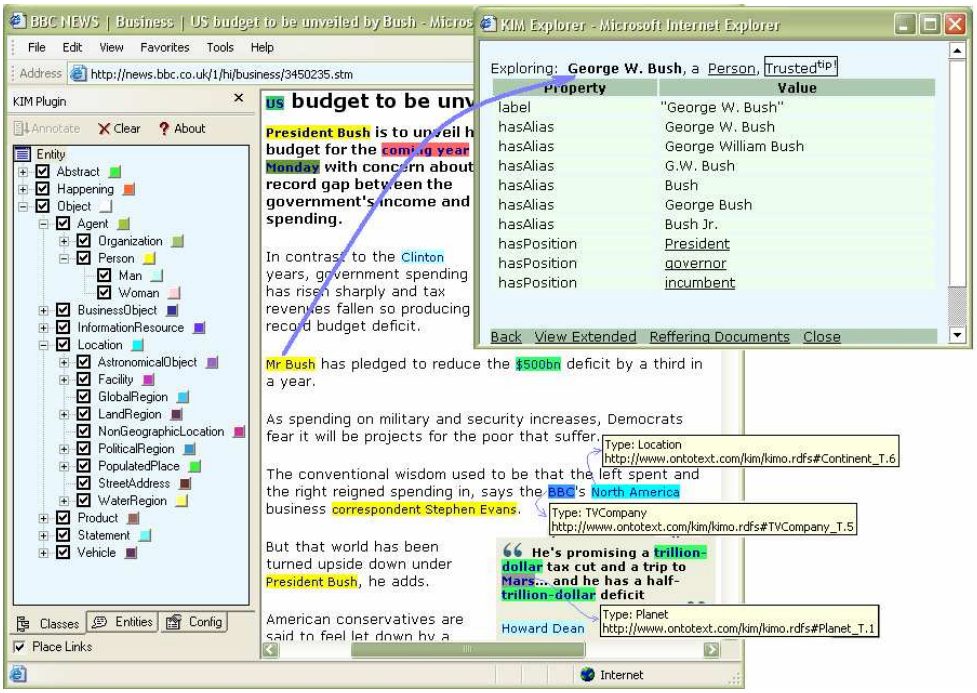

Figure 4.6: KIM plug-in showing the KIM Ontology KB Explorer

The difference between TAP and KIM instance base is in the level of ambiguity – TAP has few entities sharing the same alias, while KIM has a lot more, due to its richer collection of locations. Another important difference between KIM and SemTag is their goal. SemTag aims only at accurate classification of the mentions that were found by matching the lexicalizations in the ontology. KIM, on the other hand, is also aiming at finding **all** mentions, i.e., coverage, as well as accuracy. The latter is a harder task because there tends to be a trade-off between accuracy and coverage. In addition, SemTag does not attempt to discover and classify new instances, which are not already in the TAP ontology. In other words, KIM performs two tasks –

ontology population with new instances and semantic annotation, while SemTag performs only semantic annotation.

KIM Front-Ends

KIM has a number of different front-end user interfaces and ones customized for specific applications are easily added. These front-ends provide full access to KIM functionality, including semantic indexing, semantic repositories, metadata annotation services, and document and metadata management. Some example front-ends appear below.

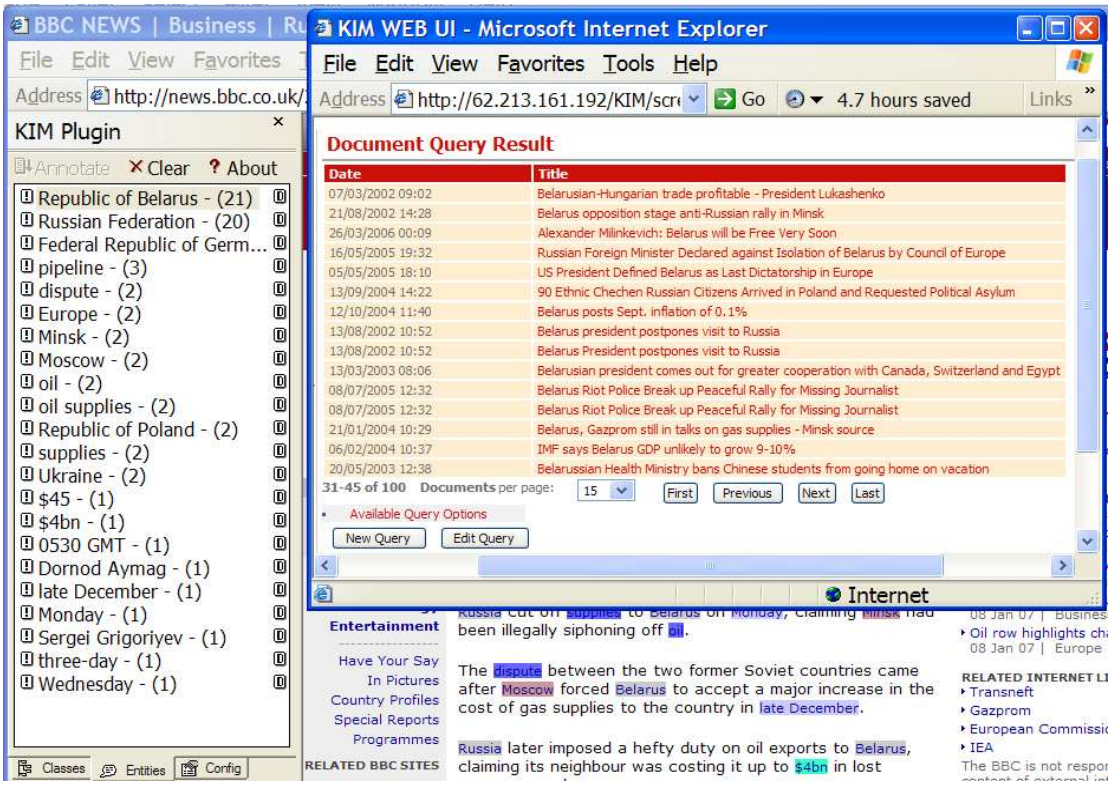

Figure 4.7: Resulting entities and related documents

The KIM plug-in for Internet Explorer[11] provides lightweight delivery of semantic annotations to the end user. On its first tab, the plug-in displays the ontology and each class has a colour used for highlighting the metadata of this type. Classes of interest are selected by the user via check boxes. The user requests the semantic annotation of the currently viewed page by pressing the Annotate button. The KIM server returns the automatically created metadata with its class and instance identifiers. The results are highlighted in the browser window, and are hyperlinked to the KIM Explorer, which displays further information from the ontology about a given instance (see top right window).

The text boxes on the bottom right of Figure 4.6 that contain the *type* and *unique identifier* are seen as tool-tips when the cursor is positioned over a semantically annotated entity.

Selecting the “Entities” tab of the plug-in generates a list of entities recognised in the

[11] KIM Plug-in is available from http:// www.ontotext.com/kim

current document, sorted by frequency of appearance, as shown in Figure 4.7. This tab also has an icon to execute a semantic query. The result is then shown as a list of documents. The goal is to enable users, while browsing and annotating, to find seamlessly other related documents by selecting one or more entities from the current document.

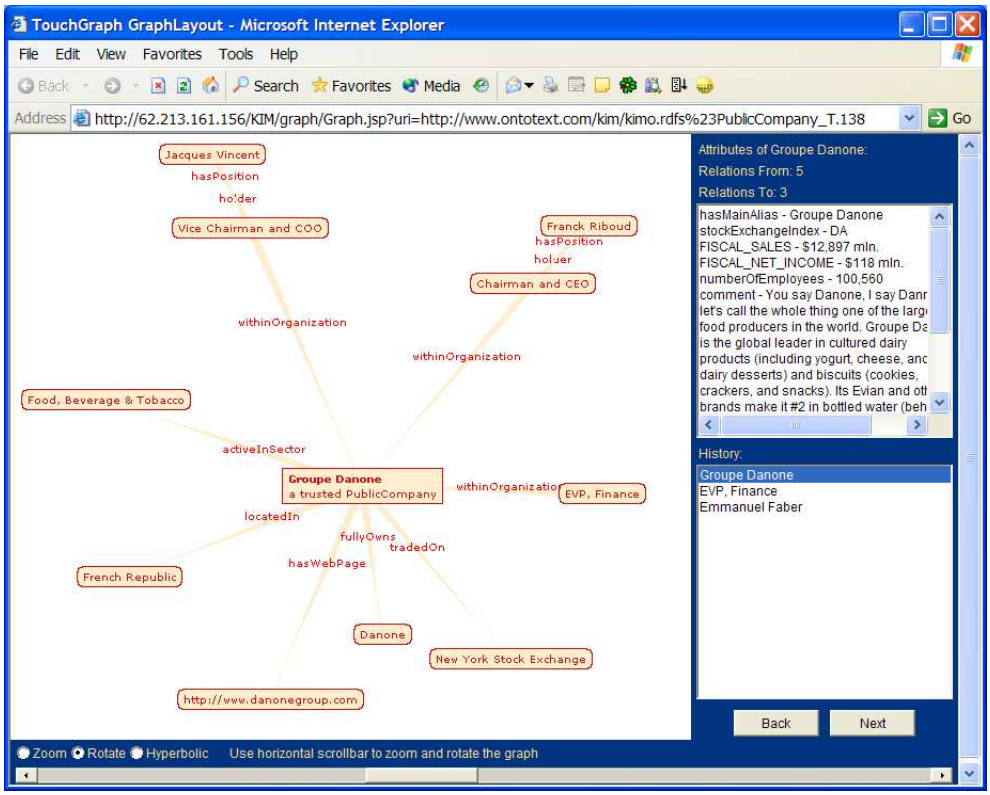

Figure 4.8: A Graph View of the description in Danone Group

Finally, a hyperbolic tree-based graph representation of the semantic repository is also available as an alternative of the KIM Explorer. As presented in Figure 4.8, the graphical explorer visualization is focused on one entity, (Danone Group is demonstrated on the screenshot). Its links to and from other entities are presented via labeled arcs. The attributes of the entity are provided in a separate pane on the right-hand side of the view. Further, the entities which are not in focus can also be "expanded", in order to visualize their link as well (this is the case with the "Chairman and CEO" node on the screenshot).

4.3. Semantic Annotation of Named Entities

Semantic annotation of named entities (SANE) is a specific metadata generation process, aiming to enable new information access methods and to extend some of the existing ones. The annotation schema, discussed here, is based on the intuition that named entity references constitute an important part of the semantics of the documents in which they occur. Moreover, via the use of KBs of external background knowledge, those entities can be related to formal descriptions of themselves and related entities and thus provide more semantics and connectivity to the web.

In a nutshell, SANE is character-level annotation of mentions of entities in the text with references to their semantic descriptions (as presented in Figure 4.5). This sort

of metadata provides both class-level and instance-level information about the entities. Such semantic annotations enable many new types of applications: highlighting, indexing and retrieval, categorization, generation of more advanced metadata, smooth traversal between unstructured text and available relevant structured knowledge. Semantic annotation is applicable for any sort of text – web pages, non-web documents, text fields in databases, etc. Further knowledge acquisition can be performed on the basis of the extraction of more complex dependencies – analysis of relationships between entities, event and situation descriptions, etc.

To use SANE in information retrieval, two basic tasks need to be addressed:

- Identify and mark references to named entities in textual (parts of) documents and link these references to descriptions of the entities in a KB[12];
- Index and retrieve documents with respect to the entities they refer to.

The first task resembles at the same time a basic press-clipping exercise, a typical IE[13] task, and hyper linking. The resulting annotations then provide the semantic data for document enrichment and presentation, which can be further used to enhance information retrieval (the second task) as discussed in Section 6.

4.4. Named Entities

In the Natural Language Processing (NLP) field, and particularly the Information Extraction (IE) tradition, **named entities** (NE) are considered: *people, organizations, locations*, and others, referred to by name, [Chinchor and Robinson 1998]. By a wider interpretation, these also include scalar values (*numbers, dates, amounts of money)*, *addresses*, etc.

NEs should be handled in a different, special way because of their different nature and semantics compared to general words (terms, phrases, etc.) While the former denote particulars (individuals or instances), the latter typically denote universals (concepts, classes, relations, attributes). Even a basic level of formal semantic definition of general word senses involves modelling of lexical semantics and common sense[14]. On the other hand, useful descriptions of named entities can be modelled on the basis of much simpler and more specific “factual” world knowledge.

4.5. Semantic Annotation Model and Representation

In this section we discuss the structure and the representation of SANE, including the necessary knowledge and metadata. The basic prerequisites for the representation of semantic annotations are:

---

[12] There can also be references to various ontologies. On one hand, there could be a direct reference from the annotation to the class of the entity, as it is defined in an ontology. On the other, some instances can be part of ontologies (if they are considered part of shared conceptualization; see section Ontologies).

[13] Information extraction (IE) is a relatively young discipline within Natural Language Processing (NLP), which conducts partial analysis of text in order to extract specific information, [Cunningham et al, 1999].

[14] WordNet is the most popular large scale lexical database, providing partial descriptions of the word senses in the English language. It can be considered also as a lexical ontology or a KB.

- an ontology, defining the entity classes and allowing unambiguous references to those;
- entity identifiers, which allow these to be distinguished and linked to their semantic descriptions;
- a knowledge base (KB) with entity descriptions.

Entity descriptions actually make up the non-ontological part of formal knowledge in the semantic repository. The entity descriptions represent a KB, a body of instance knowledge or data. Such KB can either be available as pre-populated background knowledge and/or be extended through information extraction from the documents.

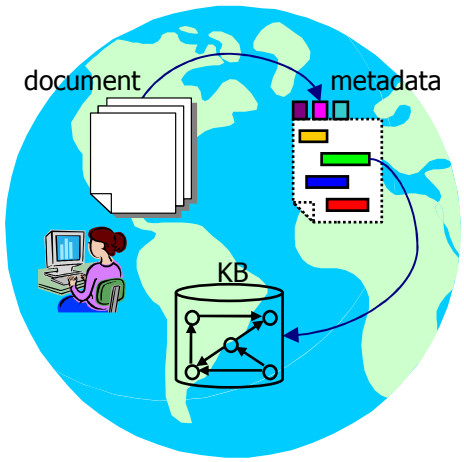

Figure 4.9: Distributed Heterogeneous Knowledge

As with other sorts of annotations, a major question about the representation is: “To embed or not to embed?” There are a number of arguments, giving evidence that semantic annotations are best decoupled from the content they refer to. One key reason for this is the ambition to allow for dynamic, user-specific, semantic annotations – conversely, embedded annotations become a part of the content and may not change according to the interest of the user or to the context of usage. Further, complex embedded annotations would have a negative impact on the volume of the content and could complicate its maintenance – e.g. imagine that a page with three layers of overlapping semantic annotations needs to be updated without compromising their consistency.

Given that semantic annotations should preferably be kept separate from the content to which they refer, the next question is whether or not (or to what extent) the annotations should be integrated with the ontology and the KB. It is the case that such an integration seems profitable – it would be easier to keep the annotation in sync with the class and entity descriptions. However, there are at least three important considerations to be made in this regard:

- Both the number and the complexity of the annotations differ from those of the entity descriptions – the annotations are simpler, but more numerous than the entity descriptions. Even considering middle-sized corpora of documents, the number of annotations typically reaches tens of millions. Suppose that 10M annotations are stored in an RDF(S) store together with 1M entity descriptions. Suppose also that on average annotations and entity descriptions are represented with 10 statements each. The difference, regarding the inference approaches and the hardware that is capable of efficient reasoning and access to a 10M-statement semantic repository and to a 110M-statement repository, is considerable.
- Separation of concerns: if the world knowledge (ontology and instance data) and the document-related metadata are kept independent, this would mean that for one and the same document, different extraction, processing, or authoring methods will be able to deliver alternative metadata, referring to one and the same knowledge store.

- Importantly, it should be possible that the ownership and the responsibility for the metadata and the knowledge are distributed. In this way, different parties can develop and separately maintain the content, the metadata, and the knowledge.

On the basis of the above arguments, we propose a general model which allows for decoupled representation and management of the documents, the metadata (annotations), and the formal knowledge (ontologies and instance data), as illustrated in Figure 4.9.

5. Semantic Indexing and Retrieval

As already mentioned, one of the key tasks which can be performed on top of semantic annotations is indexing and retrieval of documents with respect to the corresponding semantic features. This is a modification of the classical IR task – documents are retrieved on the basis of relevance to concepts instead of words. However the basic model is quite similar – a document is characterized by the bag of tokens[15] which constitute its content, disregarding its structure. While the basic IR approach considers the word lemmata (base forms) or stems as tokens, there have been considerable efforts for the last decade related to indexing with respect to two sorts of higher-level semantic features, namely:

- word-senses, lexical concepts, references to controlled vocabularies;
- named entities (including numbers, dates, etc.).

Both types of indexing can serve as a basis for cross-lingual IR. Lexical-concepts in one language can be related to such in another language or to some sort of interlingua – this was one of the main objectives for the development of the EuroWordNet[16] lexical ontology. On the other hand, once properly recognized, the named entities references are language independent (both the mentions of “London” and “Llundain” will be tagged with one and the same entity identifier).

5.1. Indexing With Respect to Lexical Concepts

Many of the words in natural languages are polysemous, i.e. they can have more than one meaning. For instance, the word “bank” can denote both a financial institution and a river bank. In WordNet (and other lexical ontologies) this linguistic phenomena is handled through association of the word with different lexical concepts[17] representing their different meanings. One of the main problems in the course of semantic annotation with respect to lexical ontologies is word-sense disambiguation (WSD) – the selection of the correct lexical concept, which represents the meaning of the word in the specific context. Once WSD is performed and documents are annotated with respect to lexical concepts, indexing and retrieval with respect to them solves the problem with query term polysemy and index term synonymy mentioned in section 1.1. WSD and the usage of lexical resources for IR

[15] Sometimes “token” is used with a specialized meaning in the NLP field – in essence, tokens are the elements of the text, as they are separated by white-spaces and punctuation (the delimiters, are also considered tokens). [Kiryakov and Simov, 1999] introduces the term “atomic text entities” (ATE) as a general notion of token in IR context, to avoid ambiguity with the NLP usage of “token”.

[16] http://www.illc.uva.nl/EuroWordNet/

[17] The lexical concepts in WordNet are called synsets (from synonym set).

are discussed in Chapter 4. Here we will comment only on the principle advantages that such indexing can provide, given a lexical ontology with super-concept/sub-concept relationships, as well as possible indexing and retrieval techniques, as discussed in [Kiryakov and Simov, 1999].

Suppose the following IR setup with hierarchically-structured feature space:

- The documents are indexed with respect to (occurrence in the documents of references to) lexical concepts.
- The lexical concepts are properly related to the corresponding more general concepts (hypernyms) and less general concepts (hyponyms).
- The query is specified through lexical concepts (either because the user directly selected them, or because a "bag of keywords" query has been semantically annotated).

Let us define *hyponym-matching* as a retrieval operator which matches more general concepts in the query with more specific ones in the document. Using such operator a query containing the concept for "bird" will match documents referring "duck" or "eagle"; this follows the intuition that if a specific species of birds are mentioned than, than this is also a reference to bird. On the contrary, a document which refers to "bird" will not be hyponym-matched to a query including "hen"; intuitively, there is a no guarantee that a document about birds have something to say about "hens".

First, let us note that hyponym-matching offers clear benefits for the users as compared to the mainstream IR techniques (e.g. vector-space model using word lemmata as features). In a typical search engine, documents mentioning "duck", but not mentioning explicitly "bird", will not be returned as a result for a query for "bird" – the lack of hyponym-matching leads to poor recall. To fix this "manually" the user should include in the query all possible species of birds and their synonyms, which would be an unreliable and inefficient exercise. Further, the words from the expanded query will match words in the text, which may be used in a different meaning there – for example, the query expansion will contain "duck" which in a specific text could have been used for cloth[18]. The effect is a reduction in precision.

An interesting question is how hyponym-matching can be integrated into an existing system, e.g. a vector-space based probabilistic model. Let us see first how indexing and retrieval can be performed with respect to lexical concepts (disregarding hyponymy). Suppose that before being indexed the documents are pre-processed as follows: (i) they are semantically annotated with lexical concepts and (ii) each occurrence of a word (or a multi-word token) is replaced by the identifier of the corresponding concept. The queries can be pre-processed in the same manner. In a simplified world, this is an easy way to make an existing IR engine implement semantic search – the vector-based similarity between queries and documents should be as good model for relevance, as when documents are indexed with respect to word lemmata.

One straightforward solution for extending this model with hyponym-matching is query expansion. Each of the concepts in the query can be replaced with the set of itself plus all of its hyponyms (sub-concepts). This approach is simple and can prove

[18] According to WordNet 2.1 (http://wordnet.princeton.edu/perl/webwn) one of the meanings of "duck" is: a heavy cotton fabric of plain weave; used for clothing and tents.

sufficient in many contexts, but one should be aware of its disadvantages:

- A concept with a bigger set of hyponyms will gain bigger weight in the relevance calculation as opposed to such with no or just a few hyponyms. This problem can partly be solved if the IR engine supports disjunction (i.e. OR operator) – however, this is not a natural feature for the engines based on probabilistic models.
- The set of all sub-concepts of all the concepts in the query, could grow to thousands of elements, which can cause problems with the performance of the IR engine.

An alternative approach (let us name it *hypernyms-indexing*) is to modify the indexing strategy, so, that the documents are indexed with respect to the hypernyms of the concepts they refer to. In such case, a document mentioning the concept "eagle" will also appear in the reverse index for its super-concept (e.g. "bird") and the super-concept of the super-concept (e.g. "animal") and so forth following the subsumption chain to the top concept. In cases when there is no control of the engine's indexing strategy, this effect can be achieved if each word gets annotated not only with the specific lexical concept, corresponding to its meanings, but also with the super concepts. Then in the document pre-processing phase, the identifiers of the super-concepts will also appear in the document index.

Query expansion is no longer necessary, when hypernyms-indexing is involved, because the relevance of the document to the more general concepts has been reflected in the indices. A query for "bird" will retrieve a document which only mentions "eagle", without any need for query modification. The problems of this approach can be summarized as follows:

- The overall size of the indices will grow, due to the fact that each token in the document appears in multiple indices. The growth can be estimated as a factor close to the average depth of the of the hypernymy hierarchy;
- The indices for the most general concepts will get huge and cause efficiency problems.

These problems can be addressed to some extent if limited query- or index-expansion is performed. For instance, one can put a constraint on the number of levels of the hypernyms to be considered or to the total number of hyponyms to be used for expansion. In all cases, it should be clear that the adaptation of a probabilistic IR model (tuned for a flat feature set) to deliver good performance for a hierarchically structured feature set is far from trivial. Experiments show that the adaptation of a "general-purpose" lexical ontology for this task is problematic, [Voorhees, 1998]. An interesting implementation is reported in [Mahesh et al, 1999]: a large scale lexical ontology, build for the specific for IR task, is used for the IR engine built into one of the major RDBMS systems. The evaluation of the system on some of the tasks of the TREC competition, prove clear performance benefits for this approach.

5.2. Indexing With Respect To Named Entities

Historically, the issue of special handling of named entities seems to have been somewhat neglected by the information retrieval (IR) community, apart from some shallow handling for the purpose of Questions/Answering tasks. However, a recent large-scale human interaction study on a personal content IR system of Microsoft,

[Dumais et al, 2003], demonstrates that, at least in some cases, named entities are central to user needs:

> "The most common query types in our logs were People/Places/Things, Computers/Internet, and Health/Science. In the People/Places/Things category, names were especially prevalent. Their importance is highlighted by the fact that 25% of the queries involved people's names, which suggests that people are a powerful memory cue for personal content. In contrast, general informational queries are less prevalent."

As the volume of web content grows rapidly, the demand for more advanced retrieval methods increases accordingly. Based on semantic annotation of named entities (SANE), efficient indexing and retrieval techniques can be developed, involving an explicit handling of the named entity references.

In a nutshell, SANE could be used to index both "NY" and "N.Y." as occurrence of the specific entity "New York", as though a unique identifier for that entity occurred in the text in place of the different syntactic variations of the strings used to denote it. Since present systems do not involve entity recognition, they will index on "NY" (for the former), and "N" and "Y" (for the latter), which demonstrates well some of the problems with the keyword-based search engines.

Given metadata-based indexing of content, advanced semantic querying becomes feasible. A query against a repository of semantically annotated documents can be specified in terms of restrictions on the entity's type, name, attribute values, and relations to other entities. For instance, a query can request all documents that refer to Person-s that hold some Position-s within an Organization, and which also restricts the names of the entities or some of their attributes (e.g. a person's gender). Further, semantic annotations can be used to match specific references in the text to more general queries. For instance, a query such as "Redwood Shores company" could match documents mentioning specific companies such as ORACLE and Symbian, which are located in this town.

Hybrid query modes, like the one mentioned above, can provide unmatched analytical levels through a combination of:

- Database-like structured queries, extended with the reasoning capabilities of the semantic repositories.
- IR-like probabilistic models.

Although the above sketched enhancements look promising, further research and experimentation are required to determine to what extent and in which way(s) they can improve existing IR systems. It is hard, in a general context, to predict how semantic indexing will combine with the symbolic and the statistical methods currently in use. Large scale experimental data and evaluation efforts (similar to TREC) are required for this purpose.

5.3. Exploiting Massive Background Knowledge – TAP

As discussed above, conventional search engines have no model of how the concepts denoted by query terms may be linked semantically. When searching for a paper published by a particular author, for example, it may be helpful to retrieve additional information that relates to that author, such as other publications, curriculum vitae, contact details, etc. A number of search engines are now emerging

that use techniques to apply ontology-based domain-specific knowledge to the indexing, similarity evaluation, results augmentation and query enrichment processes.

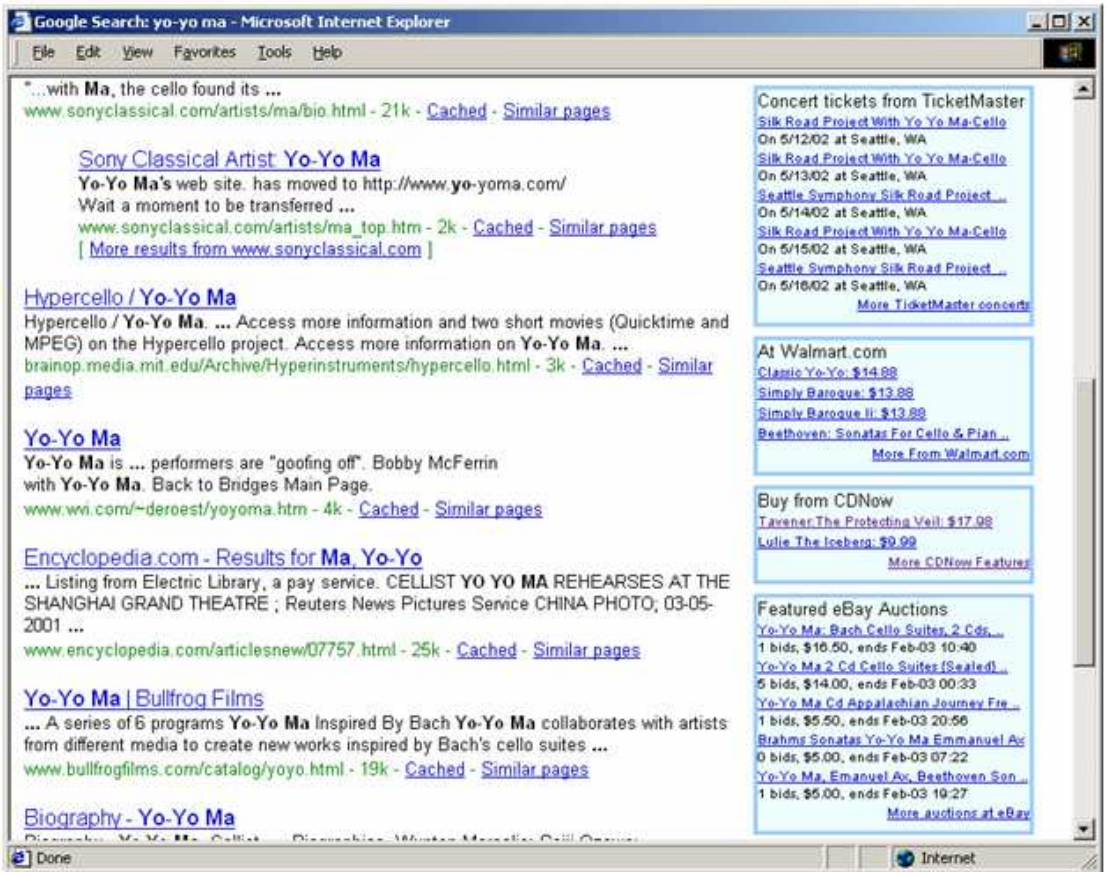

Figure 5.1: Semantic Search with TAP

TAP, [Guha and McCool 2003], is Semantic Web architecture, which allows RDF(S)-compliant consolidation and querying of structured information. [Guha et al., 2003] describe a couple of Semantic Web-based search engines: ABS – activity-based search and W3C Semantic Search. In both cases TAP is employed to improve traditional search results (obtained from Google, http://www.google.com) when seeking information in relation to people, places, events, news items, etc. TAP is used for two tasks:

1. Result augmentation: the list of documents returned by the IR system is complemented by text and links generated from the available background knowledge;
2. Query term disambiguation: the user is given the opportunity to choose the concrete entity she is searching for, than the system attempts to filter the results of the IR system to those referring only this entity. An approach using several statistics for this purpose is sketched in [Guha et al., 2003] without details on the implementation.

The semantic search application, which runs as a client of TAP sends a user-supplied query to a conventional search engine. Results returned from the conventional search engine are augmented with relevant information aggregated from distributed data sources that form a knowledge base (the information is

extracted from relevant content on targeted web sites and stored as machine-readable RDF annotations). The information contained in the knowledge base is independent of and additional to the results returned from the conventional search engine. A search for a musician's name, for example, would augment the list of matching results from the conventional search engine with information such as current tour dates, discography, biography, etc. Figure 5.1 shows a typical search result from ABS. Special attention is paid to the selection of a dataset to show and its presentation.

Relatively simple heuristics are used to find the concepts which are relevant to the query. No considerable processing of the query terms is performed – according to [Guha et al., 2003], relevant are considered concepts, which have a label (name) that contains one of the query terms.

The couple of semantic search engines mentioned above do not perform any pre-processing or indexing of the documents – they are based on an existing search engine. [Dill et al., 2004] presents a system called SemTag, which performs automatic semantic annotation of texts with respect to large scale knowledge bases available though TAP, solving a task similar to the one presented in the next section.

5.4. Character-level Annotations and Massive World Knowledge – KIM Platform

The KIM platform, [Popov et al., 2003], provides infrastructure and services for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured content.

As a base line, KIM performs character-level semantic annotation of named entities, as described in Section 4.3. The automation of this task is possible though information extraction technology, based on GATE text engineering architecture (http://www.gate.ac.uk). KIM analyzes texts and recognizes references to entities (such as persons, organizations, locations, dates). Then it tries to match the reference with a known entity that has a unique URI and description. In cases when there is no match, a new URI and description are generated automatically – this is the situation when ontology population takes place. Finally, the reference in the document is annotated with the URI of the entity. KIM is equipped with an Internet Explorer plug-in, which uses these annotations for highlighting and hyperlinking as presented in Figure 4.6. The mentions are coloured in accordance with the class (type) of the entity; hyperlinks provide access to popup forms presenting their descriptions in the KB. As the later include also references to other related entities, the user can further traverse the KB. This way the plug-in allows smooth transition from the text to the KB and exploration of the available structured knowledge.

In order to enable the easy bootstrapping of applications, KIM is based on the PROTON ontology, [Terziev et al., 2005], which consists of about 250 classes and 100 properties. Furthermore, a knowledge base (KIM's World KB, WKB), pre-populated with about 200, 000 entity descriptions, is bundled with KIM. Its role is to provide as a background knowledge (resembling a human's common culture) a quasi-exhaustive coverage of the entities of general importance – those, which are considered well-known and thus not explicitly introduced in the documents, which makes it hard to get their descriptions automatically extracted. KIM uses OWLIM

high-performance semantic repository (http://www.ontotext.com/owlim) to manage the WKB together with the extracted instance data.

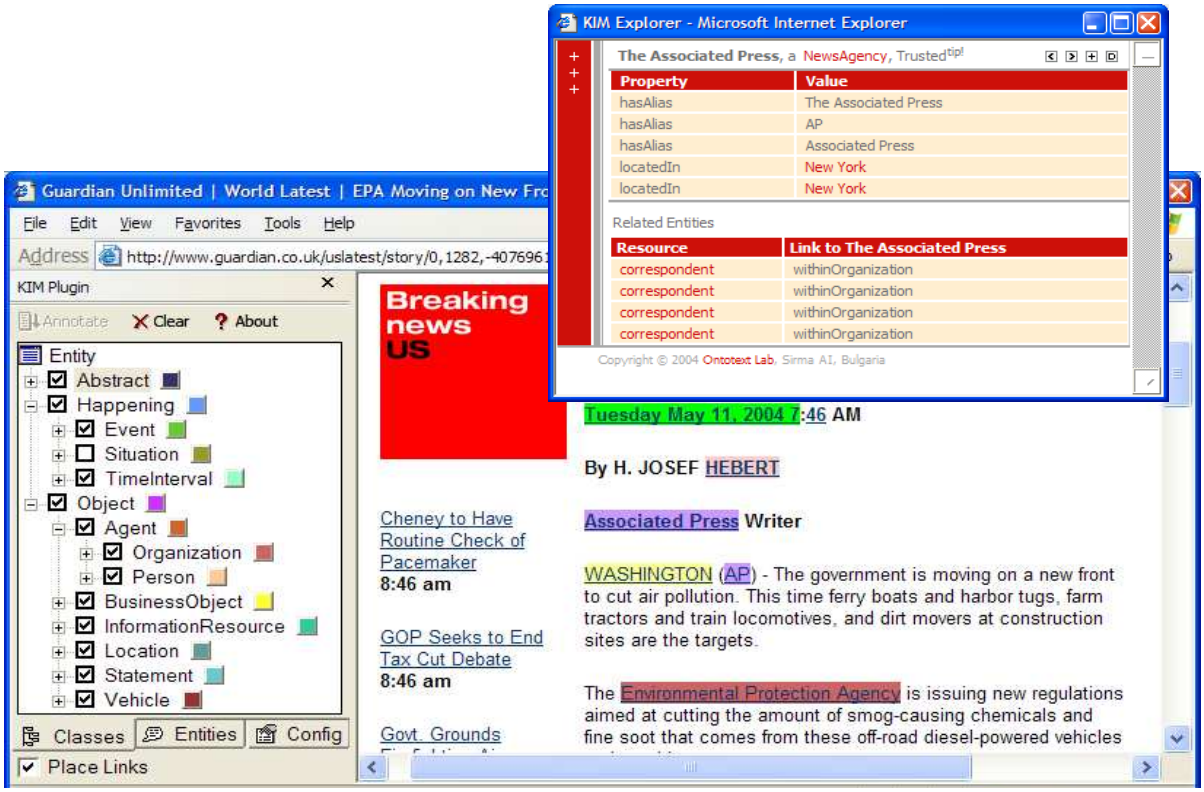

Figure 5.2: Semantic Browsing and Navigation in KIM

The semantic annotations are used under the indexing and retrieval schema presented in section 5.2. Unique entity ID is inserted in the text at the places where the entity is referred. The application of entity co-reference resolution means that the system would regard the strings “Tony Blair” “Mr Blair” “the Prime Minister” as referring to the same entity in the KB. Than the texts are passed for indexing to standard full-text search engine; in its basic configuration KIM uses for this purpose Lucene engine (http://lucene.apache.org/java/docs/).

This allows KIM to offer the semantic queries which combine structured queries, reasoning, and full-text search. The most generic search interface (named “Entity Pattern Search), allows the specification of queries about any type of entity, relations between such entities and required attribute values (e.g. “find all documents referring to a `Person` that `hasPosition` ‘CEO’ within a `Company`, `locatedIn` a `Country` with name ‘UK’ ”). To answer the query, KIM applies the semantic restrictions over the entities in the KB. The resulting set of entities is matched against the semantic index and the referring documents are retrieved with relevance ranking according to these entities.

KIM provides also a simplified search interface for several predefined patterns. In Figure 5.3 a semantic query is specified, concerning a person whose name begins with “J”, and who is a spokesman for IBM.

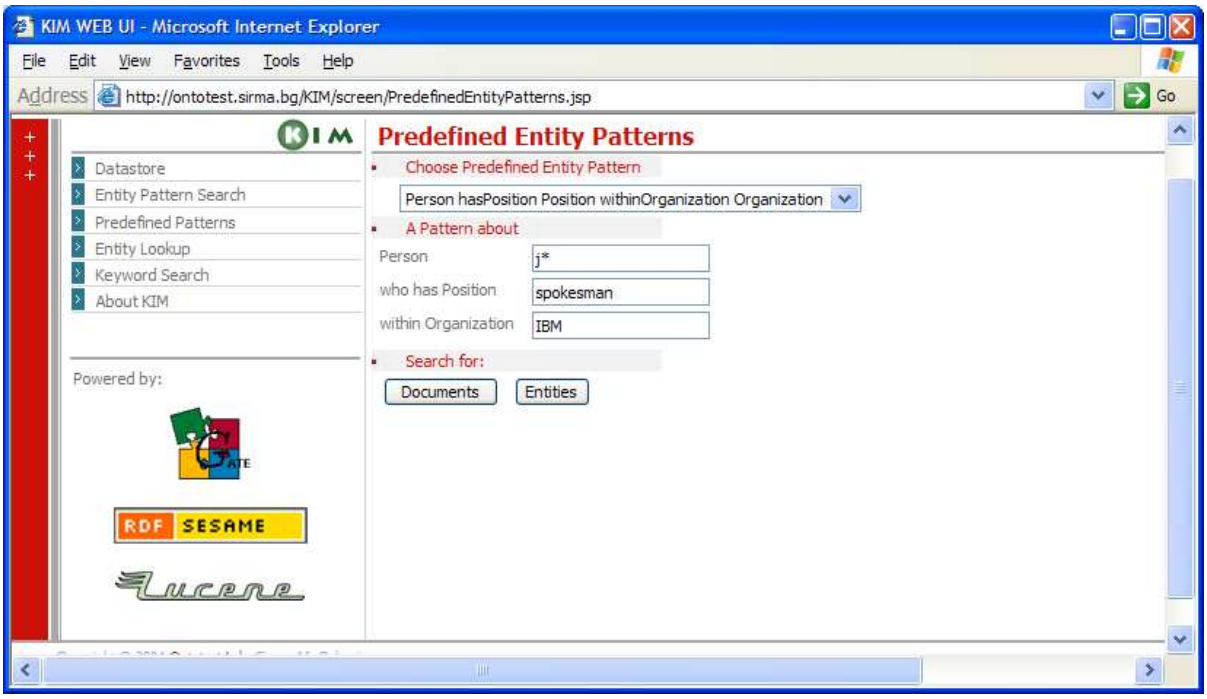

Figure 5.3: Semantic Querying in KIM

Figure 5.4 shows that four entities have been found in the documents indexed. It is then possible to browse a list of documents containing the specified entities and KIM renders the documents, with entities from the query highlighted (in this example IBM and the identified spokesperson).

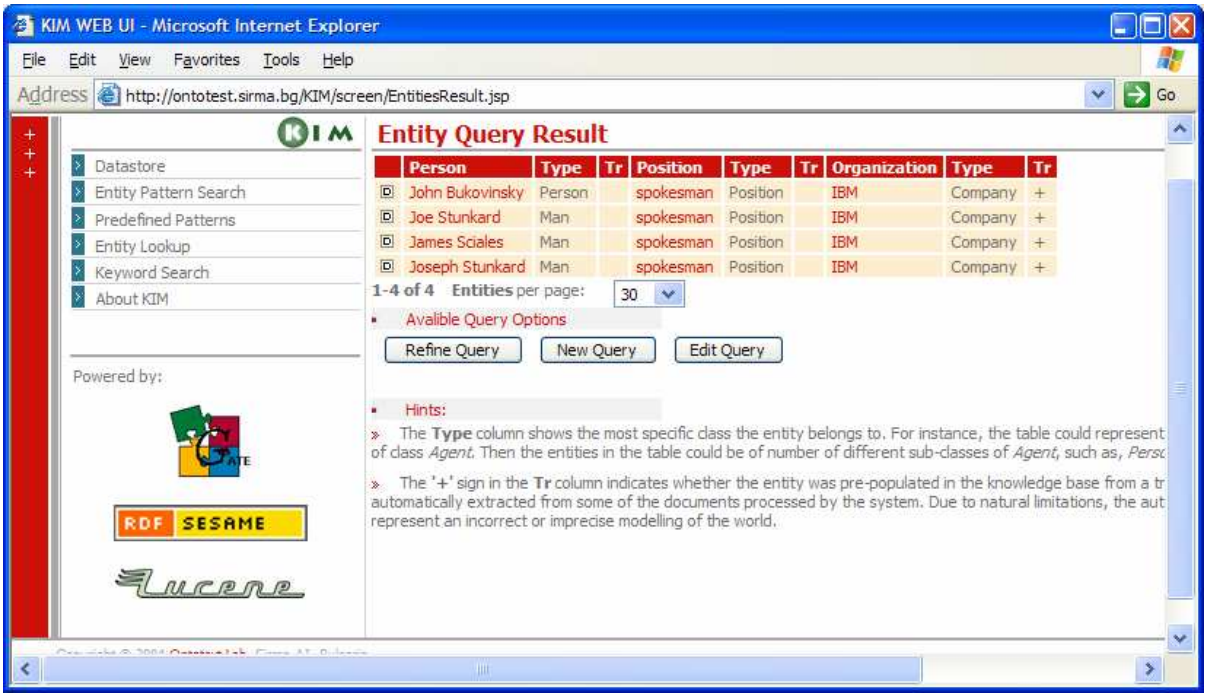

Figure 5.4: Semantic Query Results

In other work, [Bernstein et al. 2005] describe a controlled language approach whereby a subset of English is entered by the user as a query and is then mapped into a semantic query via a discourse representation structure. Vallet et al. (2005) propose an ontology-based information retrieval model using a semantic indexing

scheme based on annotation weighting techniques.

5.5. Semantic Browsing and Navigation

Web browsing complements searching as an important aspect of information-seeking behaviour. Browsing can be enhanced by the exploitation of semantic annotations and below we describe systems which offer a semantic approach to information browsing (in some cases combined with searching).

Magpie [Domingue et al., 2004] is an internet browser plug-in which assists users in the analysis of webpages. Magpie adds an ontology based semantic layer onto web pages on-the-fly as they are browsed. The system automatically highlights key items of interest (in a way similar to KIM, see Figure 5.2), and for each highlighted term it provides a set of 'services' (e.g. contact details, current projects, related people) when you right-click on the item. This relies, of course, on the availability of a domain ontology appropriate to the page being browsed – similarly to TAP, Magpie annotates on the basis of match of a label from the KB.

CS AKTiveSpace (Glaser et al., 2004) is a Semantic Web application which provides a way to browse information about the UK Computer Science Research domain, by exploiting information from a variety of source including funding agencies and individual researchers. The application exploits a wide range of semantically heterogeneous and distributed content. AKTiveSpace retrieves information related to almost two thousand active Computer Science researchers and over twenty four thousand research projects, with information being contained within thousands of published papers, located in different university web sites. This content is gathered on a continuous basis using a variety of methods including harvesting publicly available data from institutional web sites, bulk translation from existing databases, as well as other data sources. The content is mediated through an ontology, and stored as RDF triples; the indexed information comprises around ten million RDF triples in total.

CS AKTive Space supports the exploration of patterns and implications inherent in the content using a variety of visualisations and multi dimensional representations to give unified access to information gathered from a range of heterogeneous sources.

[Alonso 2006] presents an impressive proof-of-concept prototype of a KM solution, using various features of ORACLE 10gR2 to implement semantic metadata-based search and browse. The system combines RDF support, full-text indexing, and clustering functionality. The system's user interface implements several navigation modes, based on visualisation components available as libraries from third-parties.

Siderean's Seamark Navigator[19] allows for faceted search based on document-level metadata represented in RDF: subject, author, publisher, date. Internally, the system combines an RDF repository (which may or may not incorporate reasoning) and a full-text search engine. The search user interface allows for interactive focussing: for each facet the user is presented with the most popular values. For each value (e.g. a specific author) the system presents the number of documents matching this value – this provides information to the user about the "selectivity" of the specific values. The user can chose a value of a facet and it is added to the filter, following which the values and the selectivity figures presented are altered to consider only the

[19] http://www.siderean.com/products_suite.aspx

documents matching the current filter.

6. The PrestoSpace Choice

The choice in the PrestoSpace project has been influenced by the very requirements of the project. In PrestoSpace the complex multi-phase process of digitising, analysing and accessing audiovisual data requires a powerful way of modelling the analysis results and making them available as potential search and navigation restrictions and access keys. The chosen solution based on the KIM Platform includes:

- Java Server (KIM) for semantic annotation, document and annotation management, semantic indexing and search;
- Ontology (based on PROTON) to model the searchable information space;
- Knowledge Base to represent real-world entities and lower-level metadata features associated with individual audiovisual pieces of data
- Modular Web-based user interfaces to serve as a presentation layer for the purpose of accessing the digital archive underneath.

As a piece of software the KIM Server is being used both on the documentation and on the publication sides of the PrestoSpace project. On the documentation side is serves mainly as an accommodator of the extracted entities and metadata, while on the publication side it is the basis of the search and accessing of the digital archive. While it does not deal with the audiovisual content itself it deals with its description through which the user could access the relevant data to her needs.

The communication between the KIM Servers present on these two sides is implemented through the atomic communication unit in PrestoSpace: an EDOB. Through this medium the metadata associated with a material is being transferred in order to partially replicate a part of the knowledge present at documentation side to the publication server and make it available to the front-end applications there. Since the knowledge base is being changed dynamically with new material descriptions entering into the KIM Server, snippets of this very knowledge base are being transferred along with EDOBs to ensure that all the relevant knowledge allowing the publication side to “understand” the metadata of the material is present.

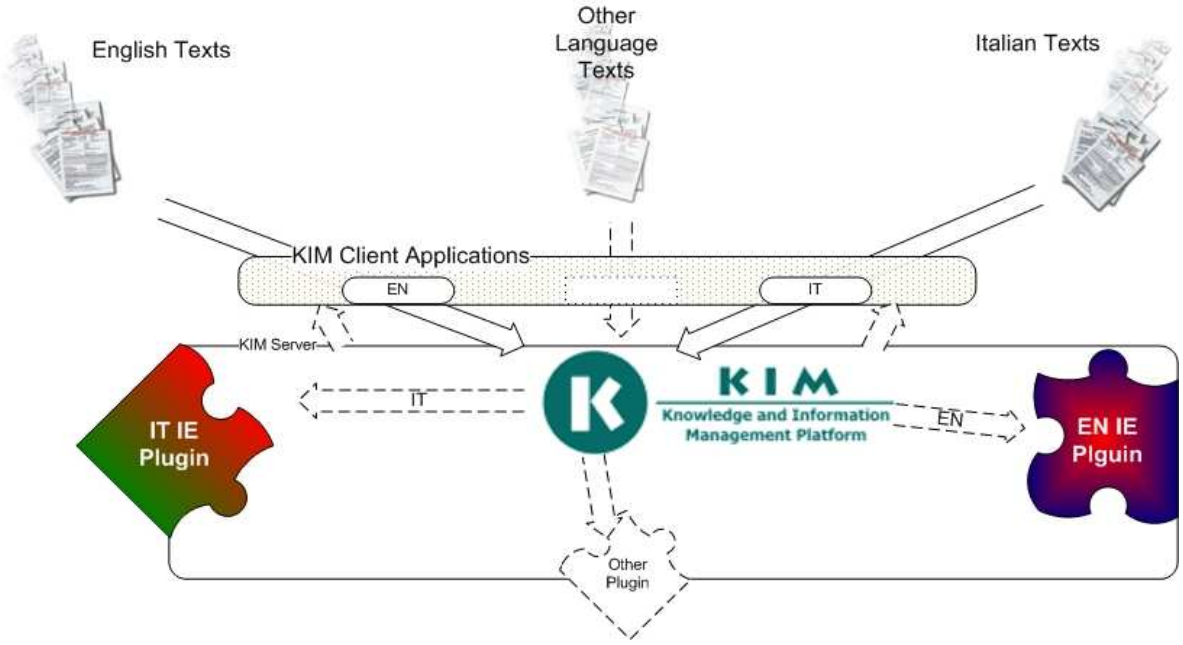

Figure 6.1: IE plugins and runtime contexts

Beside the infrastructural and domain specifics a subtle requirement in PrestoSpace is the multilinguality of the processing, representation and access methods. Ontologies and knowledge bases allow naturally multilingual aspects of the same knowledge or data which was another reason for the choice of software. However, a requirement of coexistence of more than one semantic analysis module working over text (a result of the previous analysis over the audiovisual materials) brought the requirement for the chosen infrastructure to change so to meet the needs. The solution is depicted on Fig 6.1 representing a simple scenario of having IE modules for two different languages which are optionally used depending on the type of the incoming documents.

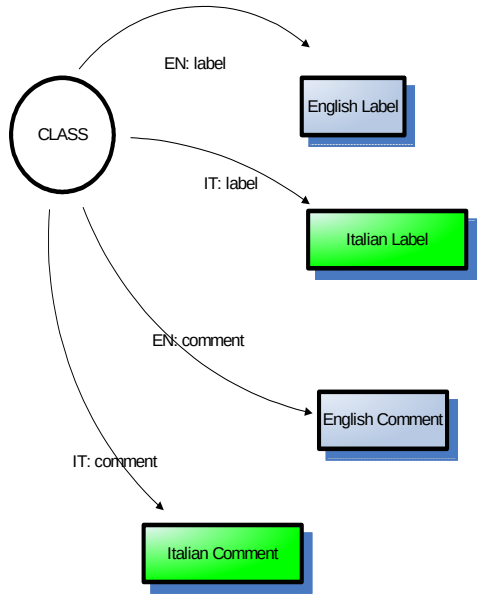

Figure 6.2: Language aspects of resources

In terms of representing the background and extracted knowledge the PROTON ontology has been used. Due to its modular nature and its basic coverage of upper-level concepts it is suitable to accommodate domain-specific extensions. In the same time this alleviates the domain-specific specialists from having to specify general concepts like *Person, Location, Organization, Date* and so on. The PrestoSpace requirements include extension of the ontology with domain-specific concepts as well as inclusion of language aspects to the existing resources. The modular structure of the ontology is depicted on Fig. 6.3.

The multilinguality in PROTON is handled by providing language specific labels and attributes of the resources coexisting in the same ontology and knowledge base. An

example of a class having two language aspects is shown on Fig. 6.2.

Beside the multilingual aspects each entity in the knowledge base is represented by a set of alternative names, its attributes and relationships to other entities (Fig.6.3).

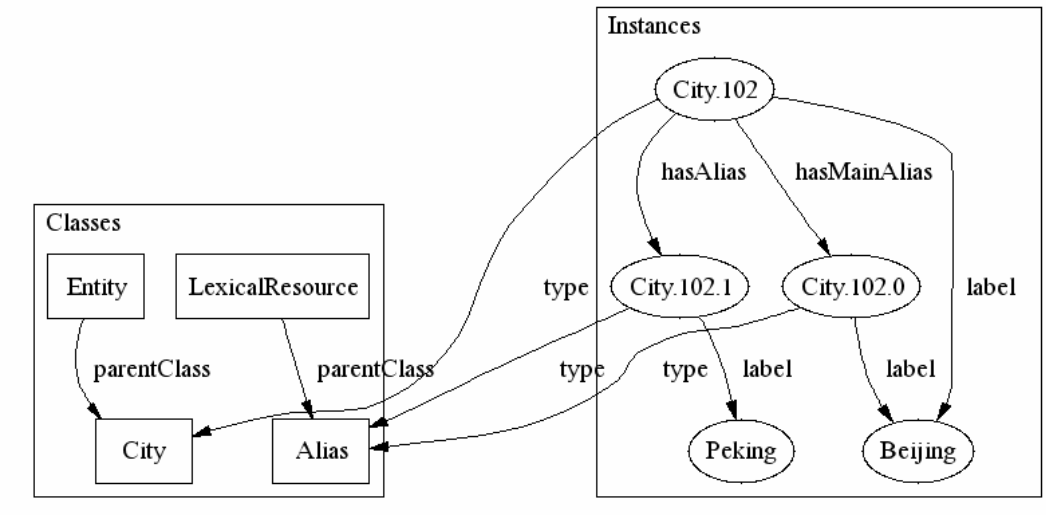

Figure 6.3: Semantic Entity Description

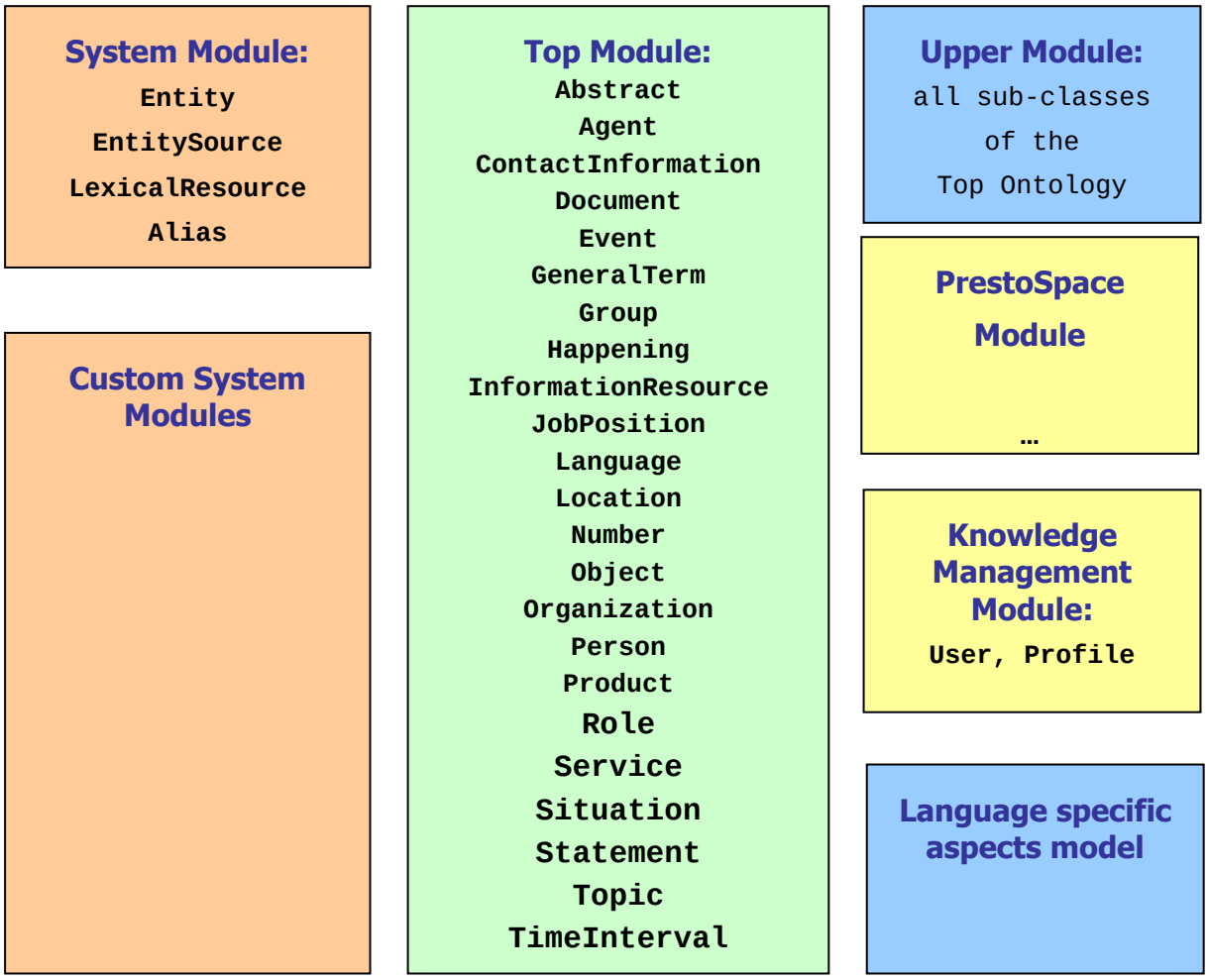

Figure 6.4: PROTON Modules

Based on the described approach the KIM Platform provides conceptual search infrastructure and allows for further extension for even more complicated search approaches relying on IE-based pre-processing and semantic indexing.

7. Conclusion

This document describes the current technological landscape concerning traditional and advanced search techniques. It also describes the pre-processing steps needed to allow conceptual or semantic indexing and eventually retrieval. Finally the KIM Platform being the choice of the PrestoSpace project has been presented. The platform encompasses all the benefits of the advanced conceptual search techniques integrated in one in a flexible yet uniform manner.

8. References

ACE, Feb 2004. Annotation Guidelines for Entity Detection and Tracking (EDT). Available at http://www.ldc.upenn.edu/Projects/ACE/

Alonso, O. 2006. *Building semantic-based applications using Oracle*. Developer's Track at WWW2006, Edinburgh, May 2006. http://www2006.org/programme/item.php?id=d16

Beckett, D. 2004. *RDF/XML Syntax Specification (Revised).* http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

Brickley, D., Guha, R.V, eds. 2000. *Resource Description Framework (RDF) Schemas,* W3C http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

Bernstein, A.; Kaufmann, E.; Goehring, A.; Kiefer, C.. 2005. *Querying Ontologies: A Controlled English Interface for End-users*, Proc. 4th Intl. Semantic Web Conf., ISWC2005, Galway, Ireland, November 2005, Springer-Verlag, 2005.

Bowker, L. 1995. *A multidimensional approach to classification in Terminology: Working with a computational framework.* PhD thesis, University of Manchester, England.

Chinchor, N., Robinson, P. 1998. *MUC-7 Named Entity Task Definition* (version 3.5). In Proc. of the MUC-7.

Cimiano, P., Handschuh, S., and Staab, S. 2004. Towards the Self-Annotating Web. In *Proceedings of WWW'04*.

Ciravegna, F. and Wilks, Y. 2003. Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, Amsterdam.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).

Cunningham, H. 2000. *Software Architecture for Language Engineering*. PhD Thesis, Univ. of Sheffield.

Cunningham, H. 1999. *Information Extraction: a User Guide* (revised version). Department of Computer Science, University of Sheffield, May, 1999.

Davies, J., Studer, R., Sure, Y. & Warren, P., *Next Generation Knowledge Management, BT Technology Journal*, Vol. 23, No 3, July 2005.

DCMI Usage Board. 2005. *DCMI Metadata Terms.* http://dublincore.org/documents/2005/06/13/dcmi-terms/

Dean, M; Schreiber, G. – editors; Bechhofer, S; van Harmelen, F; Hendler, J; Horrocks, I.; McGuinness, D. L; Patel-Schneider, P. F.; Stein, L. A. (2004). OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-ref/

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T.,

Domingue, J., Dzbor, M. and Motta, E. (2004) "Collaborative Semantic Web Browsing with Magpie" in The Semantic Web: Research and Applications Davies, J., Bussler, C.,

Fensel, D. and Studer, R. (editors). Proceedings of ESWS, 2004, pp388-401. LNCS 3053, Springer-Verlag.

Domingue, J., Dzbor, M., and Motta, E. 2004. Magpie: Supporting Browsing and Navigation on the Semantic Web. In N. Nunes and C. Rich, editors, *Proceedings ACM Conference on Intelligent User Interfaces (IUI)*, pages 191–197.

Dumais, S., Cutrell E., Cadiz J., Jancke G., Sarin R. and Robbins D. *Stuff I've Seen: A system for personal information retrieval and re-use.* In proc. of SIGIR'03, 2003, Toronto, ACM Press.

Dill, S.; Eiron, N.; Gibson, D.; Gruhl, D.; Guha, R.; Jhingran, A.; Kanungo, T.; McCurley, K. S.;Rajagopalan, S.; Tomkins, A.; Tomlin, J. A.; Zienberer, J. Y. 2003. *A Case for Automated Large Scale Semantic Annotation.* Journal of Web Semantics, 1(1), 2004.

Ehrig, M., Haase, P., Hefke, M., Stojanovic, N. (2005). *Similarity for ontologies — a comprehensive framework*. Proc. 13th European Conference on Information Systems, May 2005.

Felber, H. 1984. *Terminology Manual*. Unesco and Infoterm, Paris.

Glaser, H., Alani, H., Carr, L., Chapman, S., Ciravegna, F., Dingli, A., Gibbins, N., Harris, S., Schraefel, M.C. and Shadbolt, N. (2004) "CS AKTiveSpace: Building a Semantic Web Application" in The Semantic Web: Research and Applications Davies, J., Bussler, C., Fensel, D. and Studer, R. (editors). Proceedings of ESWS, 2004, pp388-401. LNCS 3053, Springer-Verlag.

Grishman, R. *TIPSTER Architecture Design Document Version 2.3*. Technical report, DARPA, 1997. http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/

Gruber, T. R. 1992. *A translation approach to portable ontologies.* Knowledge Acquisition, 5(2):199-220, 1993. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

Gruber, T. R. 1993. *Toward principles for the design of ontologies used for knowledge sharing*. In N. Guarino & R. Poli, (Eds.), International Workshop on Formal Ontology, Padova, Italy, 1993. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html

Guarino, N.; Giaretta, P. 1995. *Ontologies and Knowledge Bases: Towards a Terminological Clarification.* In N. Mars (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam: pp. 25-32

Guarino, N. 1998. *Formal Ontology in Information Systems.* In N. Guarino (ed.) Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, June 6-8, 1998. IOS Press, Amsterdam, pp. 3-15.

Guha, R.; McCool, R. 2003. *Tap: A semantic web platform*. Computer Networks, 42:557 – 577, 2003.

Guha, R.; McCool, R.; Miller, E. 2003. *Semantic Search*. WWW2003, May 20-24, 2003, Budapest, Hungary.

Handschuh, S., Staab, S., and Ciravegna, F. 2002. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pp. 358–372, Siguenza, Spain.

Handschuh, S., Staab, S., and Maedche, A. 2001. CREAM – creating relational metadata with a component-based, ontology-driven framework. In *Proceedings of K-CAP 2001*, Victoria, BC, Canada.

Kiryakov, A. 2006. *Ontologies for Knowledge Management*. Chapter 7 in: Davies, J; Studer, R; Warren, P. (eds.). *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Wiley, UK, 2006.

Kiryakov A., Simov K. Iv. 1999. *Ontologically Supported Semantic Matching***.** In proc. of "NODALIDA'99: Nordic Conference on Comp. Linguistics", Trondheim, Dec. 9-10, 1999.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D. 2005. "Semantic Annotation, Indexing and Retrieval", *Journal of Web Semantics*, Vol. 2, No. 1.

Kiryakov, A.; Popov, B.; Ognyanov, D.; Manov, D.; Kirilov, A.; Goranov, M. 2004. *Semantic Annotation, Indexing, and Retrieval.* Elsevier's Journal of Web Semantics, Vol. 1, ISWC2003 special issue (2), 2004. http://www.websemanticsjournal.org/

Kiryakov, A; Ognyanov, D; Manov, D. 2005. *OWLIM – a Pragmatic Semantic Repository for OWL*. In Proc. of International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, 20 Nov, New York City, USA.

Klyne, G; Carroll, J. J. 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. 3C recommendation 10 Feb, 2004. http://www.w3.org/TR/rdf-concepts/

Kogut, P. and Holmes, W. 2001. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *First International Conference on Knowledge Capture (K-CAP 2001), Workshop on Knowledge Markup and SemanticAnnotation*, Victoria, B.C.

Mahesh, K., Nirenburg, S., Cowie, J., and Farwell, D. 1996. An Assessment of Cyc for Natural Language Processing. Technical Report MCCS Report, New Mexico State University.

Motta, E., Vargas-Vera, M., Domingue, J., Lanzoni, M., Stutt, A., and Ciravegna, F. 2002. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pp. 379–391, Siguenza, Spain.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. and Goranov, M. 2003. *KIM – Semantic Annotation Platform.* In Proc. of 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Florida, USA. LNAI Vol. 2870, pp. 834-849, Springer-Verlag Berlin Heidelberg 2003

Terziev, I.; Kiryakov, A.; Manov, D. 2004. *D1.8.1. Base upper-level ontology (BULO) Guidance,* report EU-IST Integrated Project (IP) IST-2003-506826 SEKT), 2004. http://proton.semanticweb.org/D1_8_1.pdf

Vallet, D.; Fernandez, M.; Castells, P. 2005. An *Ontology-based Information Retrieval Model*, Proc. 2nd European Semantic Web Conference, ESWC2005, Heraklion, Crete, May/June 2005, Springer-Verlag, Berlin. Editors: Gómez-Pérez, A. and Euzenat, J. LNCS 3532/2005. Springer Berlin / Heidelberg

van Ossenbruggen, J., Hardman L., Rutledge L., *Hypermedia and the Semantic Web: A Research Agenda*. Journal of Digital information, volume 3 issue 1, May 2002.

8.1. Related Readings

Carr, L.; Bechhofer, S.; Goble, C.; Hall, W. 2004. *Conceptual Linking: Ontology-based Open Hypermedia.* In The WWW10 Conference, Hong Kong, May 2003, pp. 334-342.

Ciorascu, C.; Ciorascu, I.; Stoffel, K. 2003. *knOWLer Ontological Support for Information Retrieval Systems*.

Matsuo, Y.; Morim, J.; Hamasaki, M. 2006. *POLYPHONET: An Advanced Social Network Extraction System from the Web.* WWW 2006, May 22-26, 2006, Edinburgh, UK.

Mayfield, J.; Finin, T. 2003. *Information retrieval on the Semantic Web: Integrating inference and retrieval*, SIGIR Workshop on the Semantic Web, Toronto, 1 August 2003.

Moldovan D., Mihalcea R.. *Document Indexing Using Named Entities.* In "Studies in Informatics and Control", Vol. 10, No. 1, March 2001.

Pustejovsky J., Boguraev B., Verhagen, M., Buitelaar P., and Johnston M., *Semantic Indexing and Typed hyperlinking.* In Proc. of the AAAI Conference, Spring Symposium, NLP for WWW, 120-128. Stanford University, CA, 1997.