Knowledge Representation, Sharing and Retrieval on the Web

Philippe Martin

Distributed System Technology Centre, Australia
philippe.martin@gu.edu.au

Abstract. By "knowledge retrieval", we refer to the automatic retrieval of statements permitting a tool to make logical inferences and answer queries precisely and correctly, as opposed to retrieving documents or statements "related to" the queries. Given the ambiguity of natural language and our current inability to make computers "understand" it, the knowledge has to be manually encoded and structured using a formal graphic/textual language and ontologies (structured catalogs of categories and associated constraints of use).

The Web currently contains a lot of data, more and more structured data (databases, structured documents) and simple metadata but very little knowledge as defined above, i.e. very few knowledge representations. Moreover, this knowledge has been encoded using various languages and unconnected or loosely connected ontologies, and following different representation conventions. Hence, currently, not only knowledge sources are rare but each require the development of a special wrapper for their knowledge to be interpreted and hence retrieved, combined or exploited.

This article reviews various projects concerning knowledge representation, sharing and retrieval on the Web, then details requirements for a "Semantic Web" and illustrates them with notations, conventions and cooperation rules from our own tool, WebKB-2. Knowledge retrieval mechanisms and interfaces used in WebKB-2 are also given as illustrations.

Table of Content

1 Introduction

Data indexed by formal terms (categories) or organized by inclusion links is called "structured data". Beliefs, definitions, facts or rules represented using a formal language and an ontology (i.e. a catalog of concepts and relations, with constraints of use and organized by semantic links) is most often called "knowledge" (or knowledge representations/statements). The Word Wide Web currently contains a lot of data, more and more structured data (on-line databases, structured documents) and simple metadata but very little knowledge[1].

Since the semantics of natural language sentences or structured data cannot be automatically extracted, people have to explicit semantics via knowledge representations. Then, tools may make logical inferences to answer queries precisely and correctly (as opposed to retrieving data/documents "related to" the queries) and without requiring the users (or application developers) to know the exact schema (structure and/or indexation categories) used in each source of data. However, knowledge sharing, merging and retrieval are only possible if the categories used in the knowledge representations are connected by semantic links, directly (by belonging to a same ontology) or indirectly (by belonging to interconnected ontologies).

A common expectation for the "Semantic Web"[2] is that many small, specialized (and possibly competing) RDF schemas (ontologies in RDF[3]) will be developed, and that in order to make knowledge representations, people or businesses will select some schemas, re-use them, and create new RDF schemas to define terms they have not found [5]. Then, according to Tim Berners-Lee[4], future Web search engines may be able to find various statements related to certain queries and logically combine a few of them to answer the query; "while nothing will make the combinatorial explosion go away, many real life problems can be solved using just a few (say two) steps of inference out on the Web".

This vision seems *unrealistic* since it is unlikely that different people will create statements that can be *logically* matched or combined when they use unconnected or loosely connected ontologies, when only a few ontological primitives are standardized (in the RDF world, these are the categories of RDFS and DAML+OIL[5]), and when no lexical/ontological/structural/semantic conventions are adopted. Like today, future Web search engines would mostly rely on term *lexical* matching, and applications would have to write a special wrapper for each knowledge source they want to utilize (furthermore, even wrappers

[1] Although describing the same facts as we do, some researchers unfortunately use the words "knowledge" instead of "data" and "ontology" instead of "knowledge", as in [3]: "The WWW can be viewed as the largest knowledge base that has ever existed. However, its support in query answering and automated inference is very limited". The terms we use are chosen to be quite unambiguous for a member of the knowledge acquisition/representation community, and their meanings are consistent with the ones given in the "free on-line dictionary of computing" at http://foldoc.doc.ic.ac.uk/foldoc/

[2] http://www.w3.org/2001/sw/

[3] http://www.w3.org/RDF/

[4] http://www.w3.org/DesignIssues/Semantic.html

[5] http://www.daml.org/2001/03/daml+oil-index

cannot compensate for badly structured or impoverished knowledge). Matching[6] or combining statements, and hence finding knowledge relevant to a query (or even only "related" to a particular object) is a problem even in a single large knowledge base (KB) such as CYC[7] where knowledge providers are trained knowledge engineers following conventions, using a unique large ontology and an expressive knowledge representation language. Yet, even in this ideal case, some choices in the ontological and structural conventions have led to knowledge which is not explicit enough to be exploited in many applications[8].

The above cited vision also seems *undesirable* because, as we will show, more centralized approaches involving large-scale knowledge servers can (i) permit a large number of users (people or agents) to cooperatively build large knowledge bases (KBs) with explicit, expressive, normalized, highly inter-connected statements and categories, and hence permit knowledge retrieval by simple hypertext navigation or provide reasoning services at selected levels of effectiveness and completeness, and (ii) exploit the KB to ease, guide and cross-check the insertion of new knowledge by each user and her re-use/annotation/correction of other users' knowledge. These features are permitted by the incremental insertion of knowledge into centralized repositories *when* it is developed, instead of *afterwards* by Web search engines (knowledge in isolation is not knowledge but merely data; thus, loosely connected schemas/RDF documents cannot be *logically* combined and their re-use requires the development of an ad-hoc wrapper for each one). To keep the advantages of the decentralized approach of the Web, categories and statements in a knowledge server must be referable via URLs (and then exported in a standard language such as KIF), and be allowed to refer to other objects on the Web via URLs. These are easy-to-achieve constraints.

For efficiency reasons, all Web-users cannot use *the same* knowledge server but they can use several general knowledge servers (e.g. managed by portal companies) and more specialized knowledge servers dedicated to specific domains. By (partly) mirroring one another's content, general and specialized servers would share a similar general ontology like WordNet[9] or CYC's ontology, and competing specialized knowledge servers would also share some similar content[10]. Thus, it would *not matter where* a Web user publishes information first, *no unique server* would have to be relied upon, and hence this "more centralized" approach would maintain the advantages of the current decentralized approach,

---

[6] For example, the statement "John is owner of a duplex in Southport" can easily be identified as a *specialization* of the query statement "a person is owner of an apartment in a city part of the Gold Coast", provided that "John" has been declared as a "person", "duplex" as a *specialization* of "apartment" and "Southport" as being a "city" which is "part" of the "Gold Coast".

[7] http://www.cyc.com/tech.html#cycl

[8] For example, actions/processes are represented as n-ary relations instead of concept nodes with explicit thematic relations to the related objects. As shown in Section 3.4, this decreases the possibility of matching or combining statements about processes.

[9] http://www.cogsci.princeton.edu/~wn/

[10] The similarity of the KBs also permits the processes of mirroring and answering queries involving several KBs.

without its problems. (A similar architecture for distributed KBs and a small-scale implementation is discussed in CYC[11]).

In this article, we first review some projects about knowledge representation, sharing and retrieval on the Web. Subsequently, we show that *within a KB as well as across the Web*, knowledge sharing and exchange implies that knowledge providers use a unique set of ontological primitives, follow lexical/structural/semantic recommendations, and use (directly or via interfaces) high-level expressive knowledge representation languages that ease the adoption of the recommendations and lead to *comparable*[12] knowledge representations. Thus, we argue that the Semantic Web implies the standardization of such elements, and show the elements adopted and implemented in our public knowledge server and annotation tool, WebKB-2[13]. We summarize the protocols used in WebKB-2 to permit the asynchronous cooperative building of the KB by the users, and finally present search interfaces and mechanisms.

2 Elements and Landmarks of Knowledge Representation and Sharing on the Web

RDF was not the first step towards knowledge sharing on the Web, and in many aspects can be seen as a backward step. The list of elements and landmarks below is organised thematically but also chronologically. It is by no means exhaustive but the selected landmark tools or ontologies continue to be the most known or usable nowadays. Readers that are familiar with the field of knowledge sharing and the Semantic Web can restrict their attention to Subsection 2.3 only.

2.1 Exchange Formats and Programming Interfaces

KIF (Knowledge Interchange Format)[14] is a low-level but expressive language (first order logic plus contexts and sets) originally designed in 1992 to permit translations between more specialized knowledge representation languages (KRLs). It has become the de-facto standard for expressing the semantics of KRLs in a computable way. It is complemented by the Ontolingua library[15] which formalizes (in KIF) various elements necessary for expressing the semantics of KRLs, e.g. sets, relations, functions, numbers and frames. By comparison,

[11] http://www.cyc.com/applications.html#dai

[12] Category/statement comparability is an important notion in this article since knowledge retrieval or inferencing is based on statement comparison and combination. A statement (resp. a category) is comparable to another statement (resp. category) if it generalizes it or specializes it. *Logic generalizations* are logic deductions. Some generalizations are simply structural simplifications and not logic deductions. This is detailed at the beginning of Section 3.4.

[13] Usable at www.webkb.org

[14] http://logic.stanford.edu/kif/dpans.html

[15] http://WWW-KSL-SVC.stanford.edu:5915/

RDF is also a low-level language but far less expressive and unsuited for logical inferences[16] or as an interlingua (this will be discussed in Section 3).

KQML (Knowledge Query and Manipulation Language)[17] is a KIF-based message format and message-handling protocol designed in 1994 to support run-time knowledge sharing among agents. It has been re-used or extended by various other agent communication languages.

GFP (Generic Frame Protocol)[18] (1995) is a set of functions that supports a generic Application Programming Interface (API) for frame representation systems (FRSs). Various FRSs have implemented a GFP server, e.g. Loom[19] and SRI[20]. The GKB-Editor (Generic Knowledge Base Editor)[21] is a GFP client that permits the graphical browsing and editing of FRSs that have GFP servers. GFP was extended and replaced by OKBC (Open Knowledge Base Connectivity)[22] in 1998.

2.2 Ontologies and Knowledge Bases

Ontologies are catalogs of categories with their associated complete or partial formal definitions which can also be seen as "constraints of use". Complete definitions are definitions of necessary and sufficient conditions (to be instance of the category). Partial definitions may be prototypes (listing "probable" relationships), definitions of sufficient conditions, definitions of necessary conditions (e.g. "subtype of" and "instance of" links from a category to another), etc. Categories may be relation types (called "properties" in RDF; they include functional relation types or "functions"), concept types (called "classes" in RDF) and individuals (class instances that are not classes themselves).

Some ontologies are about mathematical entities (e.g. sets, relations, functions, numbers, sequences and bags), or about relationships from/to physical dimensions (e.g. space, time and matter), or about a particular domain (e.g. elevators and chemical elements). They are often called "theories", are generally small, and may include, generalize, specialize or compete with other theories. Since 1993, the Ontolingua server[23] has hosted a library of such ontologies and permitted Web users to add new theories or combine theories to create knowledge bases.

Some ontologies classify all the concepts of a natural language or a particular domain, via links such as "subtype of", "instance of" and "part of". They are often called lexical ontologies and may be large. For example, WordNet[24][11] is a "lexical database for English" that was Web-accessible as early as 1990, and now connects about 337,200 words to about 109,400 concept types, and organizes

---

[16] For example, see www-rdf-logic mailing list archive at http://lists.w3.org/Archives/Public/www-rdf-logic/
[17] http://www.cs.umbc.edu/kqml/
[18] http://www.ai.sri.com/~gfp/
[19] http://www.isi.edu/isd/LOOM/LOOM-HOME.html
[20] http://www.ai.sri.com/~sipe/
[21] http://www.ai.sri.com/~gkb/
[22] http://www.ai.sri.com/~okbc
[23] http://www-ksl-svc.stanford.edu:5915/
[24] http://www.cogsci.princeton.edu/~wn

these types via various kinds of links, e.g. `specialization`, `exclusion`, `similar`, `member`, `part` and `substance`.

Some ontologies classify relation types (e.g. spatial/temporal/thematic relation types) and/or very general concept types (e.g. the notions of situation, state, process, spatial entity, physical_entity) mainly via "subtype of" links. They are often called top-level ontologies. Examples are John Sowa's ontologies[25] (1984 and 2000) and the Generalized Upper Model[26] (1994).

Top-level ontologies may be used for structuring the lop layers of lexical ontologies. For example, Sensus[6] was created in 1994 by semi-automatically merging WordNet, LDOCE (the Longmann Dictionary of Contemporary English) and two top-level ontologies: the Generalized Upper Model and Ontos. Similarly, in 1995, we have used Sowa's first top-level ontology to structure WordNet top layers and hence permit semantic checking on the use of WordNet categories [7]. In 1998, HPKB upper[27] was created by combining Sensus top-level ontology with CYC top-level ontology[28].

Categories of lexical ontologies may be used as generalizations for the categories in theories (which are generally much more precisely defined) and hence permit the retrieval and comparison of these categories and theories. This was also a goal for our work in 1995.

A knowledge base (KB) is composed of one ontology (or several interconnected ontologies) plus additional statements using these ontologies. The classification of certain statements as belonging or not to the ontology is only an implementation dependent issue. Such a distinction does not need to be made in WebKB-2. When this distinction is made in other tools, statements that involve individuals and no universal quantifier, are likely to be considered as not belonging to the ontologies. We do not make any distinction when we use the word "knowledge".

2.3 Ontology Servers

Ontology servers can also be called KB servers but the emphasis on the ontology highlights the fact that they permit Web users to modify the ontology part of the KB, which other KB servers do not allow (this technical limitation/simplification is also why database servers do not allow the interactive modification of the database schema).

The Ontolingua server[29] was probably the first ontology server (1993) and remains active. It has an HTML interface and also permits the use of KIF files. The reading and editing of each "theory" may be restricted to a group of users but, apart from locking/session mechanisms, no particular support for synchronous or asynchronous cooperation between users is provided.

Ontosaurus[30] (1996) is also an ontology server with an HTML interface that permits each user to build or edit theories. It exploits the Loom[31] FRS.

[25] http://users.bestweb.net/~sowa/ontology/
[26] http://www.darmstadt.gmd.de/publish/komet/gen-um/node1.html
[27] See HPKB-UPPER-LEVEL-LATEST in Ontolingua
[28] http://www.cyc.com/cyc-2-1/cover.html
[29] http://www-ksl-svc.stanford.edu:5915/
[30] http://www.isi.edu/isd/ontosaurus.html
[31] http://www.isi.edu/isd/LOOM/LOOM-HOME.html

The Co4 system[32] (1996) permits some asynchronous cooperation between users via protocols modeled on submission procedures for academic journals, i.e. on peer-reviewing. The result is a hierarchy of KBs, the uppermost containing the most consensual knowledge while the lowermost KBs are the KBs of contributing users. This approach leverages some problems of interconnecting and comparing independently developed ontologies but doubtfully scales to large numbers of users.

Tadzebao and WebOnto[33] (1998) support some synchronous cooperation between co-temporal users (they can exchange multimedia messages and be warned of each other's actions).

As opposed to most other ontology servers, WebKB-1[34] [8] (1998) does not store knowledge onto the server disk but can load and interpret Web-accessible files that combine text, images and knowledge in various formats (mainly Conceptual Graphs [12] and Formalized English[35]). The knowledge parts are isolated via delimiters, e.g. the XHTML tags `<KR language="CG">` and `<KR>`. (Approaches where knowledge is encoded *within* HTML tags or *via* XML tags are discussed in the next section). WebKB-1 has an indexation language permitting users to index any part of any Web-accessible file by a knowledge statement. It also has a language of commands that permits lexical based queries, knowledge-based queries. In answer to queries, instead of knowledge statements, the document elements indexed by the knowledge statements may be displayed. Commands can be used within the documents where they can be associated to hyperlinks or combined to create scripts that can be used to solve problems[36]. Thus, WebKB-1 is also a knowledge-based private annotation tool and a light-weight directed Web robot.

WebKB-2[37] [10] (2001) inherits most of the features of WebKB-1 (although its indexation and command languages are more limited) and also permits users to store knowledge into a unique KB on the server disk. As opposed to most other ontology servers, the knowledge from the various users is not stored into various loosely connected ontologies but tightly integrated into a same ontology/KB that has WordNet as backbone (thus, categories and statements are easier to retrieve, compare and re-use). Lexical problems are avoided by prefixing each category identifier with the identifier of its source (user, organization, document, ontology, ...). Lexical facilities are provided by the distinction between category identifiers (which are unique) and category names (which may be shared). The cooperative building of the KB is supported via the enforcement of editing rules (presented in Section 4). This type of asynchronous cooperation is likely to be more scalable in the numbers of users and knowledge quantity than Co4's and leads to a better integration of the knowledge. WebKB-2 also departs from other ontology servers by the size of the ontology it can manage. At present, WebKB-2 integrates various top-level ontologies plus the part of WordNet 1.7 concerning nouns (i.e. 108,000 nouns connected to 74,500 categories organized by various

32 http://ksi.cpsc.ucalgary.ca/KAW/KAW96/euzenat/euzenat96b.htm
33 http://ksi.cpsc.ucalgary.ca:80/KAW/KAW98/domingue/
34 www.webkb.org
35 http://www.webkb.org/doc/languages/
36 http://www.webkb.org/kb/sisyphus1.html
37 www.webkb.org

kinds of links). One of the few other KB systems that can manage large and dynamically modifiable ontologies is Parka-DB system[38] (1994).

Many RDF parsers exist but we do not know of any ontology server capable of really exploiting RDF, although Corese[39] does convert some RDF input into simple Conceptual Graphs[40] (CGs) and then is able to retrieve them by looking for specializations of a query graph. RDF export is simpler to implement than RDF exploitation, but RDF exports are either restricted to very simple knowledge or are ad-hoc, needs the use of extensions, and therefore requires a special interpretation to be re-used. Ontobroker (discussed in the next subsection) does some exports in RDF. WebKB-2 can export links between categories in RDF. In [9] (2000), we proposed conventions and extensions to RDF/XML for the represention of various knowledge representation cases related to the use of contexts, negation, universal quantification, collections, intervals, declarations and definitions. Late 2001, we extended this work, taken into account the recent developments of the DAML+OIL top-level ontology, and showed how each of these cases could be represented in Formalized English, KIF and Conceptual Graphs and, to a certain extent, RDF/XML[41]. We have begun to implement the import and export of RDF/XML in WebKB-2 with respect to *these* conventions and extensions.

2.4 Knowledge Within Web Documents

SHOE[42] (1996) was the first well-known language and server permitting the insertion of knowledge within HTML documents. Its claims, approach and notation were (and still are) surprisingly similar to the views and reasons of the W3C for the "Semantic Web"[43] (1998) and its recommended notation, RDF/XML. XML was first published as a W3C recommendation in 1998. The RDF model and its XML notation, RDF/XML, were designed to permit the representation and sharing of knowledge (which is difficult to do directly with XML)[44]. They were published as a W3C recommendation in 1999. In 2000, the syntax of SHOE was slightly modified to be XML-compliant. Unlike RDF/XML documents, SHOE documents can be any HTML document into which some SHOE markups have been added. However, since it is XML-based, that knowledge is difficult to read and write by people. Furthermore, the knowledge within a document is restricted to be about the object represented by the document (e.g. if a file is about a certain person, its URL becomes an identifier for that person and the knowledge lists relationships from that person to other objects). Hence, it seems there must be as many documents as individuals. Furthermore, categories cannot be declared/defined and used in the same document. Many knowledge-oriented

---

38 http://www.cs.umd.edu/projects/plus/Parka/parka-db.html
39 http://www-sop.inria.fr/acacia/soft/corese.html
40 http://www.cs.uah.edu/~delugach/CG/
41 See http://www.webkb.org/doc/translations.html
42 http://www.cs.umd.edu/projects/plus/SHOE/
43 http://www.w3.org/DesignIssues/Semantic.html
44 http://www.w3.org/DesignIssues/RDF-XML.html

XML-based notations (with associated parsers and sometimes inference engines) now exist, e.g. Rule-ML[45], OML[46], DAML[47] and RDF/XML.

Noticing that XML-like notations force the author of a text document to *duplicate* the information into a difficult to write formalized version, the authors of Ontobroker [3][48] (1998 to 2000) have opted for a tight integration to HTML: instead of introducing new tags, they ask the document authors to insert an attribute called "onto" *into* anchor tags and use the value to formalize the destination of the anchor. For example,
 means that Richard (identified by his home page URL) is a researcher. If this metadata is in Richard's home page, it may be abbreviated: . Under the same conditions,
IIIA
means that Richard's affiliation is http://www.iiia.csic.es/. Each document had to be registered to the Ontobroker server (which accessed the registered documents from time to time). The ontology could not be define in the document but had to be defined in the Ontobroker server. More precisely, modifications to the ontology had to be submitted to the person authorized to modify the ontology and discussed by the people using it. To our knowledge, the ontology within Ontobroker's web-accessible server was very small (a few dozens categories mainly about research domains and researcher/student levels). Hence, the users could only *index* their research domains and professional status with the available categories. They could not actually *represent* the content of their research, or anything else, since they could not declare new categories. Because of these restrictions and the poor expressivity of the KRL illustrated above, Ontobroker (claimed by its authors to be the "first Semantic Web server") was mainly *used as* a small database server. This is not to say that Ontobroker could not have been used as a normal ontology server since it could also parse Frame-Logics, a first-order logic frame-oriented language, but it seems that this language could only be used as a query language by Web users. To sum up, *the way* Ontobroker was proposed to Web users was probably the most unscalable and unusable way imaginable.

As introduced above, WebKB-1 (1998) and WebKB-2 (2001) exploit a third way to store knowledge in HTML documents: high-level expressive and easily readable knowledge representations that are separated from the rest of the document by special delimiters. (Examples are given in Section 3.2). If necessary, the representations may also be hidden by enclosing them between HTML comment tags. The user may mix category declarations/definitions and other statements, as long as a category is declared before being used. With WebKB-2, the user may also re-use the categories of the shared KB (about 77,000 at the beginning of 2002) via their identifiers or, when there is no ambiguity, via one of their names. Then, *when satisfied* with the content of the document, the user may commit it to the KB (even with high-level languages, knowledge modelling is not unlike

[45] http://www.dfki.uni-kl.de/ruleml/
[46] http://www.ontologos.org/OML/OML-Examples.html
[47] http://www.daml.org/
[48] http://ontobroker.semanticweb.org/

programming: it requires various syntactic and semantic checking, corrections and sometimes re-organizations).

HTML hyperlinks and some other HTML tags, especially the definition tag, can be viewed as a high-level, easily readable but poorly expressive way of encoding knowledge. For example, in WebKB-1, the following statements in Formalized English (FE), Frame-CG (FCG) and HTML were equivalent:

```
FE:   The car that has for owner John has for weight 1750 kg.
FCG:  [the car, owner: John, weight: 1750 kg]
HTML: <dl><dt>The car <dd>owner: John
                          <dd><dl><dt>weight<dd>1750 kg</dl></dl>
```

We have not re-used this idea in WebKB-2 because of its limited interest given the possibility of using Formalized English. However, using HTML elements as a way to avoid writing (too much) RDF/XML is an idea currently explored[49] by Dan Connolly (W3C).

3 Requirements for a Viable Semantic Web

As highlighted in the introduction, we think the vision of the Semantic Web as a collection of documents that re-use barely connected RDF schemas is not only unrealistic but undesirable. In this section, we list some elements that are necessary for the realization of the Semantic Web.

3.1 Need for a Standard Library of Ontological Primitives

RDF is not a particularly expressive language even with the semantic augmentations provided by the "standard" schemas RDFS and DAML+OIL. For instance, we have not found any (non ad-hoc) way to represent simple sentences like "5 persons dance together"[50] or "51% of people are women" in RDF. Many logic-related problems with RDF can be found in the www-rdf-logic mailing list archive[51].

The lack of expressiveness of RDF and the absence of standard ontological primitives force knowledge providers to represent information in a biased or impoverished way or invent their own (mutually incompatible) extensions. Both cases make knowledge exploitation, sharing and re-use difficult.

Many formal specification languages such as Z[52] come with a mathematical toolkit, i.e. functions and relations related to the building blocks for knowledge

[49] http://www.w3.org/2000/07/hs78/

[50] There is no "set" class nor "size" property/relation in RDF, RDFS or DAML+OIL. There is a "cardinality" property in DAML+OIL but it is about the number of relations that instances of a certain class can have. Representing "together" is also a problem since there is neither a way to represent that an universally quantified variable is within the scope of an existentially quantified variable, nor a special keyword to specify a "collective" interpretation for a collection (RDF only proposes the "distributive" and "cumulative" interpretations).

[51] http://lists.w3.org/Archives/Public/www-rdf-logic/

[52] http://spivey.oriel.ox.ac.uk/~mike/zrm/

representation: sets, relations, functions, numbers, sequences and bags. KIF, the best accepted knowledge exchange format, also comes with a similar toolkit and is complemented by the Ontolingua library.

A similar mathematical toolkit needs to be *standardized* in schemas such as RDFS to permit knowledge representation, sharing and exploitation. For example, RDF engines cannot provide an implementation handling sets, general negation or universal quantification if a vocabulary is not fixed. (We recognize the DAML+OIL schema is a first step in that direction).

3.2 Need for Expressive Notations

A usual concern about expressive notations is that they are too complex to handle efficiently. Actually, it is more correct to say that *when* statements use complex features *and when* these complex features are exploited by an inference engine for logical inferences, this inferencing may not be efficient. However, when statements are biased because the notation is too restrictive or the user is not precise, they cannot be exploited for (correct) logical inferencing by any application.

Most KRLs are customized for a particular inference engine and are not expressive enough for precise representations of natural language sentences, and hence, information in general. However, most information or knowledge on the Web are not dedicated to a particular application. A restricted model and notation such as RDF+RDFS+DAML+OIL and RDF/XML cannot be used as an interlingua because it *arbitrarily* limits the expression and exploitation of knowledge representations.

On the other hand, there is no harm in using expressive notations since, an inference engine is *not* obliged to take into account all the features of the language (i.e. all the categories in the "standard" schemas/ontologies) and perform all the logical deductions. What inferencing is done is an application-dependant choice, and not simply for efficiency reasons: the kinds of rules to apply (e.g. to handle modalities) can also sometimes *only* be chosen according to the application. Hence, the issues of completeness and decidability are *not* related to notations but to inference engines.

In the HTML/XML worlds, applications ignoring parts of the structured data is a tradition. In the specification of some knowledge exchange languages and APIs, such as KIF and OKBC, various *levels of conformance* for compliant inference engines are listed. Alternatively, each inference engine may advertize the categories to which they accord a special interpretation (and thence which features they exploit and how).

Inference engines do not even have to exploit category definitions: they may implement some efficient ad-hoc exploitation of them (the formal definitions still permit the semantics of the categories to be specified and permit the programmer to know and delimit the kinds of deduction the implementation performs). Techniques to search specializations of a query graph [12], or more generally, path retrieval techniques (based on structural matching and exploiting specialization links between categories), can be efficient[53] *and* provide satisfying results for

[53] If the query graph and each of the statements has a tree structure, the search complexity is polynomial (see [1]).

knowledge retrieval. For example, by treating a relation/property "not" as if it had no special meaning, WebKB-2 can efficiently retrieve the representations of "there is no duplex for rent in Southport" and "Southport is part of the Gold Coast" in answer to the (formalization of the) query "Is there an apartment for rent on the Gold Coast?". In this example, as opposed to the one given in Footnote 6, the results are not "logic specializations" of the query but nonetheless relevant answers. More details will be given in Section 3.4 and Section 5.3.

3.3 Need for High-level (and Expressive) Notations

A problem for automatic knowledge retrieval and inferencing is that a same piece of information can be expressed in many different incomparable ways. This problem is particularly acute when a *low-level* general syntax such as KIF (LISP) or RDF (XML) is employed, or when standard schemas offer partially redundant ontological primitives[54].

Some ways to represent information are more explicit, re-usable, comparable and easier-to-handle than other ones. Hence, to improve knowledge use and re-use possibilities: (i) *knowledge representation conventions (or "recommendations") should be standardized*; (ii) *high-level languages (or graphical interfaces) should guide the user and lead her to use the adopted conventions*. We have proposed a minimal set of lexical/structural/ontological recommendations in [9]. We give a summary of these in the next section. These recommendations are also usefully observed within a KB server. WebKB-2 users are asked to follow them and the high-level notations that we have designed – Frame-CG and Formalized English – encourage their adoption.

Frame-CG (FCG)[55] is a notation that we have derived from CGLF (the Conceptual Graph[56] linear form) [12] to improve on its readability and expressivity (which were already the main reasons for the success of Conceptual Graphs). The three main improvements were: (i) the introduction of many kinds of quantifiers in the form of English articles or expressions (e.g. "many", "between 2 and 5", "at least 6.5%"); (ii) a shorter and more natural way to express relations between objects; and (iii) the convention that the scope and precedence of quantifiers in a graph (seen as a logic formula) are related to the graph structure and node order (as in predicate logic)[57].

Formalized English (FE) is identical to FCG apart from some syntactic sugar used for grouping and connecting objects. The model behind these notations (i.e. the model implemented in WebKB-2)[58] may be seen as a generalization of the

[54] For example, to represent an "xor" between two statements, one could think of using an RDF "alt" container, a DAML+OIL "disjointWith" relation (by creating an anonymous class for each of the statements) or a classic "xor" relation (e.g. KIF "xor" relation).

[55] Grammar in http://www.webkb.org/doc/F_languages.html#FCG

[56] http://www.cs.uah.edu/~delugach/CG/

[57] In CGLF, only contexts are important to determine the scope of quantifiers; otherwise, universal quantifiers are assumed to have wider scope than existential quantifiers except when the keyword "@certain" is associated to them; this convention leaves room to ambiguities.

[58] http://www.webkb.org/doc/dataModel.html

Conceptual Graph model, RDF and terminological logics. Like these models, it is a logic-based semantic network model and permits to store logical statements. The KIF model is not yet completely included but soon will be (we currently have some problems with sets and second-order statements).

To illustrate FCG and FE and compare them to the other cited languages, below is the representation of an English sentence in CGLF, FCG, FE, KIF, predicate logic (PL) and RDF/XML (the XML format for the RDF data model). Namespaces are omitted. "Ned" is assumed to be a declared identifier for an instance of the type "Person". The 's' at the end of "cars" and "sells" in the FCG and FE representations are automatically removed by WebKB-2 (since a universal-like quantifier is used with these categories).

```
E [59] : Ned sold (the same) 3 cars twice on the 21/1/2001.
CGLF: [Person: Ned]<-(agent)<-[Sell: {*}@2]-
                                { ->(object)->[Car: {*}@3 @certain];
                                  ->(time)->[Date: #21/1/2001];  }
FCG:  [3 cars, object of: (2 sells, agent: Ned, time: 21/1/2001)]
FE:   3 cars are object of 2 sells with agent Ned and time 21/1/2001.
KIF[60]:(forAllN 3 '?c car (forAllN 2 '?s sell
         (and (agent '?s Ned) (object '?s '?c) (time '?s '21/1/2001))))
```

PL: $\exists cars\ set(cars) \wedge size(cars, 3) \wedge \forall c \in cars$
$\exists sells\ set(sells) \wedge size(sells, 2) \wedge \forall s \in sells$
$agent(s, Ned) \wedge object(s, c) \wedge time(s, 21/1/2001)$

```
RDF[61]: <kif:Set ID="cars"><size>3</size></kif:Set>
        <rdf:Description aboutEach="#cars">
           <rdf:type resource="Car"/>
           <object><rdf:Description>
                       <kif:Set ID="sells"><size>2</size></kif:Set>
                       <rdf:Description aboutEach="#sell">
                          <agent resource="Ned"/> <time>21/1/2001</time>
                       </rdf:Description>
                    </rdf:Description></object>
        </rdf:Description>
```

More translation examples can be found on the WebKB-2 site[62].

The need for higher-level (and more expressive) notations than RDF/XML is well recognized[63]. As "an academic excercise", Tim Berners-Lee has begun the design of Notation3[64], another notation for RDF which has some points in common with CGLF, FCG, FE and frame languages. (However, Notation3

[59] This sentence does not specify whether the cars have been sold individually, 2 by 2, or 3 by 3. This ambiguity is kept in the representations.

[60] Here is our KIF definition for the "forAllN" quantifier:
(defrelation forAllN (?num ?var ?type ?predicate) :=
 (exists ((?s set)) (and (size ?s ?num)
  (truth ^(forall (,?var) (=> (member ,?var ,?s) (and (,?type ,?var) ,?predicate))))))

[61] This RDF representation is only a tentative.

[62] E.g. at http://www.webkb.org/doc/translations.html and http://www.webkb.org/kb2/translation.html

[63] http://www.w3.org/DesignIssues/Logic.html

[64] http://www.w3.org/DesignIssues/Notation3.html

does not (yet) have any special syntax for extended quantifiers, collections, functions and definitions). Although Berners-Lee writes that he has not designed Notation3 "as an alternative to RDF's XML syntax which has the fundamental advantage that it is in XML", one may wonder what this advantage is supposed to be since he also acknowledges that most notations may be "Web-ized"[65] by using URIs for category identifiers. Even if knowledge can be represented in XML, it is unlikely that XML objects are directly used by advanced inference engines, and that knowledge providers read or write XML-based languages. Hence, translations to and from the XML world are necessary. From a purely syntactical viewpoint, the use of a Lisp-like notation (such as KIF) as a general low-level interlingua makes more sense because Lisp is concise and has adequate quotation (contextualization) features.

From any viewpoint we can think of, the use (and ideally, the standardization) of a high-level expressive notation would make even more sense since then knowledge is easier to write, read, compare, exchange and exploit[66]. Being readable and not XML-based, knowledge representations can also be mixed and hyperlinked with text and images within HTML/XML documents (WebKB-1 and WebKB-2 exploit such documents).

3.4 Need for Lexical/Structural/Ontological Conventions

Consider the statements "a person is doing something" and "Ned is selling a car" and their FCG representations `[a person, agent of: an activity]` and `[Ned, agent of: (a sell, object: a car)]`. The second graph is a *specialization* of the first, i.e. it has more information in its structure (one more relation) and in its components ("Ned" is an instance of the type "person" and "sell" is a subtype of "activity"). Therefore, since only existential quantifiers are involved in those graphs, the second logically entails the first[67]. In other words, if the first is used as a query graph, the second is a logical answer.

Similarly, the second graph can also be seen as a specialization of the FCG given in the previous example but, since it involves universal quantifiers, there is no logical entailment relation between the two graphs. Hence, we simply say the graphs are *comparable* (in the same way that two categories are comparable if they are linked by a subtype link or an instance link).

Now, suppose that a user declares a relation type "sell" to represent the information "A person sells a car" via 2 nodes linked by a relation; in FCG: `[a person, sell: a car]`. This graph leaves the "agent" and "object" relations implicit and is not comparable to any of the previous graphs. The user could associate a definition to the relation type "sell" to permit the expansion of the previous graph to: `[a person, agent of: (a sell, object: a car)]` but such an expansion can be a complex process and few inference engines perform

[65] http://www.w3.org/DesignIssues/RDFnot.html

[66] Let us stress again that a high-level expressive language such as FCG or FE is not intended to limit what the knowledge provider can express but how she express it, and furthermore its expressiveness does *not* impose constraints on what inference engines must do.

[67] For more details and a mathematical proof, see [1].

it. The relation type “sell” cannot be re-used when other relationships (such as “time” or “purpose”) have to be represented, and would be incomparable with other relation types “sell2” and “sell3” used to represent these relationships. Furthermore, relations cannot be quantified. In summary, the use of relations other than basic binary relations should be avoided because this use leads to representations that are less explicit and comparable. Even if a Web-based knowledge-oriented information retrieval engine does some *lexical matching* on category names to complement structural/semantic matching, concept types “sell” are more likely to be used in unrelated KBs (if basic binary relations are used) than relation types such as “sell2” or “sellSomethingAtSomeTime” (these kinds of identifiers are quite typical when relational/functional syntaxes such as Lisp are used).

As opposed to concept types, there is not a great number of basic binary relation types needed to represent natural language. For example, WebKB-2 has about 74,500 concept types derived from the WordNet lexical database about nouns, but it has a stable ontology of only 140 relation types and 50 of these types appeared sufficient to us for representing most usual natural languages sentences. Basic binary relation types are an efficient way to guide and normalize the knowledge representation task. Thanks to the signatures associated with these relation types, an inference engine can easily perform some elementary semantic checking and propose corrections when signatures are violated.

Because of its Lisp-like syntax, KIF does not encourage the use of basic binary relations only. Like most frame-based or graph-based languages, RDF only accepts binary relations but its cumbersome XML syntax discourages knowledge providers to be precise. For the same reasons, KIF and RDF discourage the use of adequate quantification, and do not prevent the use of verbs, adverbs, and adjectives as category identifiers/names even though such categories cannot be quantified (e.g. “any qualify” and “3 qualified” are meaningless), can rarely be compared to other categories, and leave information implicit. Thus, to permit knowledge sharing, lexical/structural/ontological conventions are required, and their observance needs to be encouraged by high-level notations.

RDF/RDFS and the “Meta Content Framework Using XML”[68] have some “naming conventions” for category identifiers: words used should be singular, with a lowercase first letter for relation types and an uppercase first letter for other kinds of categories, and the intercap style should be adopted when the identifier is composed of several words. Using names in the singular is a sound convention because categories can then be quantified in various ways (whereas for example a category “cars” cannot be easily quantified (what “a cars” or “any cars” mean?) and is not comparable to “car”). However, with the intercap style and the first letter in uppercase, the correct cases in the names may be lost and, at least in English, there is no way to recover that information. Readable and correctly spelled category identifiers are needed when using the identifiers in menus or presenting information with languages such as Formalized English (FE). (In RDF, correct spellings can be specified via the `label` relation but this is a cumbersome and rarely used feature).

Hence, a summary of a minimal set of conventions that we advocate is:

[68] http://www.w3.org/TR/NOTE-MCF-XML/#secA.

– *lexical conventions.* Whenever possible, use a correctly written English singular noun or nominal expression for each category identifier. Separation between words is to be done with underscores (dashes and quotes may be used when part of the usual spelling of words, e.g. "Niemann-Pick_disease" and "Fallot's_tetralogy").
– *structural/ontological conventions.* Only use basic binary relations and respect reading conventions[69]. Whenever possible, use or specialize categories from standard ontologies and use the least expressive ontological primitives: try to avoid general negation, disjunctions, second-order statements, collections, etc. In the RDF context, this amounts to using RDF and DAML+OIL ontological primitives whenever possible. Within WebKB-2, this amounts to selecting categories of the shared ontology and then following the menus or using the "For Ontology" (FO) notation for links between categories and FCG or FE for other kinds of statements. Via these notations and the ontology (relation types, general schemas/templates, etc.), the WebKB-2 user is guided to represent most things in a normalized way: states, processes, descriptions, indexations, characteristics, measures, numbers, collections, temporal/spatial/logical entities/relations, etc.
– *semantic conventions.* Be as precise as possible: give adequate quantifiers, contextualize statements in time, space, authorship, etc. Re-use and complement existing knowledge. To enforce this in WebKB-2, (i) a category can only be declared by connecting it with another and it must have at least one generalization or specialization, and (ii) a statement cannot be entered if it contradicts or is directly comparable to an existing statement, unless the author asserts the relationships between the two statements.

3.5 Need for Flexible Ways to Refer to a Category

There are more efficient and elegant approaches than others to avoid lexical problems in a KB when there are multiple knowledge providers and multiple names for each category.

In RDF, a category is uniquely identified by a URI, e.g. `http://www.foo.com` and `http://www.bar.com/doc.html#car`. Within a multi-user KB server, *it makes more sense to use user identifiers than document URIs as knowledge source identifiers.* Thus, in WebKB-2, a category identifier can be a URI (or an e-mail address) but also the concatenation of the knowledge provider's identifier and a key name, e.g. `wn#dog`, `wn#time`, `pm#IR_system` ("wn" refers to WordNet 1.7 and "pm" is the login name of the user represented by the category `philippe.martin@gu.edu.au`). In this third case, the category may still be referenced from outside the KB by prefixing the identifier with the URL of

[69] Most semantic networks models (including RDF) have adopted the convention that a relation "R" from a node "A" to a node "B" should be read "the R of A is B" or "A has for R B". In models where relations can be of any arity (e.g. KIF) no such convention is generally advocated, resulting to various usages and interpretation problems (e.g. in the ontologies of KIF, the relations "subset" and "member" counter-intuitively have the source set as a second argument instead of as the first).

the KB, e.g. `http://www.webkb.org/kb/wn#time`. This method is used when knowledge is exported in RDF/XML.

In addition to an identifier, a category may have various names (which may also be names of other categories). In FE, FCG and FO, a category identifier may show several names, e.g. `wn#dog__domestic_dog__Canis_familiaris` (at least two underscores must be used for separating the names). Given 95% of current categories in WebKB-2 come from WordNet, the "wn" prefix may be left implicit, e.g. `#time` means `wn#time`. More precisely, "wn" is the default creator. An ordered list of default creators can be specified, e.g. "`default creators: pm wn;`".

Below is the way the FO notation can be used in WebKB-2 to store that the concept type `pm#thing` has been created on the 29/11/1999, given two names by its creator "pm", that the user "oc" has added a French name and an "instanceOf" link to the RDF "class" category, that "pm" has added a disjointWith link to the uppermost relation type (the link creator is left implicit since it is the same as creator of the source category) and given three subtypes, two of which forming a "close partition" ("disjoint union" in DAML terminology).

```
pm#thing__top_concept_type (^thing that is not a relation^) 29/11/1999
  _  chose (oc fr),
  ^ rdfs#class (oc),
  ! pm#relation,
  >  {(pm#situation pm#entity)} pm#thing_playing_some_role;
```

Here is a partial translation in RDF/XML. The creators of the links could not be represented in a standard/simple way.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
         xmlns:daml="http://www.daml.org/2000/10/daml-ont#"
         xmlns:pm="http://www.webkb.org/kb/theKB_terms.rdf/pm#">
<rdfs:Class rdf:about="http://www.webkb.org/kb/theKB_terms.rdf/pm#Thing">
  <rdfs:label xml:lang="en">thing</rdfs:label>
  <rdfs:label xml:lang="en">top_concept_type</rdfs:label>
  <rdfs:label xml:lang="fr">chose</rdfs:label>
  <dc:Creator>philippe.martin@gu.edu.au</dc:Creator>
  <rdfs:comment>thing that is not a relation</rdfs:comment>
  <rdf:type
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <daml:disjointWith
      rdf:resource="http://www.webkb.org/kb/theKB_terms.rdf/pm#relation"/>
</rdfs:Class> </rdf:RDF>
```

Here is how the FO notation was used by "pm" to declare a category for the "instanceOf" relation, specify the equivalent RDF category and an inverse relation.

```
pm#kind__type__class (pm#thing,rdfs#class)
  = rdf#type,
  < dc#type,
  - pm#instance;
```

Here is a partial translation in RDF/XML using the previous namespaces.

```
<rdf:Property rdf:about="http://www.webkb.org/kb/theKB_terms.rdf/pm#kind">
  <rdfs:label xml:lang="en">kind</rdfs:label>
  <rdfs:label xml:lang="en">type</rdfs:label>
  <rdfs:label xml:lang="en">class</rdfs:label>
  <dc:Creator>philippe.martin@gu.edu.au</dc:Creator>
  <rdfs:range
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <daml:samePropertyAs
          rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Type"/>
  <rdfs:subPropertyOf
                rdf:resource="http://purl.org/metadata/dublin_core#type"/>
  <daml:inverseOf
      rdf:resource="http://www.webkb.org/kb/theKB_terms.rdf/pm#instance"/>
</rdf:Property>
```

More details on our top-level ontology and how it integrates other top-level ontologies can be found on the WebKB-2 site (www.webkb.org).

WebKB-2 maintains links between each category and its creator and names, and conversely. *This permits the use of names instead of identifiers within statements as long as there is no ambiguity.* Relation signatures are exploited to eliminate candidate categories[70]. If there is more than one candidate for a category, the parsing stops and the list of candidates is printed to help the user refine her statement. For a query graph, there is no harm in making this choice automatically and let the user refine the query if an incorrect category has been selected. For improved readability, we often use names instead of category identifiers in the example graphs of this article.

A problem that prevents this facility to be adopted *within RDF documents on the Web* is that the RDF schemas they import may change (new names may be added to categories) and hence ambiguities may appear.

Within a KB that integrates a natural language ontology, this facility is particularly useful to accelerate the writing of knowledge.

[70] For example, "flight" is a name currently shared by 9 categories: 4 representing processes, 3 representing collections, 1 representing a psychological feature, and 1 representing a physical entity ("flight of stairs"). If a concept node is about a "flight" and is the destination of a relation with type pm#on_location, given the signature associated to pm#on_location, only one sense of "flight" is relevant, the one representing the physical entity "flight of stairs".

3.6 Need for a Shared Natural Language Ontology

Links from a natural language ontology such as WordNet[71] form the backbone of a large shared KB, and are a way to connect ontologies on the the Web.

Such links permit WebKB-2 to relate, compare and retrieve knowledge representations. They also provide the user with various categories (meanings) for a word, and various distinctions for a notion, many of which she may not have considered. This leads the user to enter more precise and comparable representations. The semantic constraints associated with the top level categories of the ontology are inherited by all the categories of the natural language ontology, and this permits some automatic checking on all users' statements and extensions to the ontology.

We initialized the current KB of WebKB-2 with the content of the lexical database WordNet 1.7: 108,000 nouns and 74,500 categories referred by nouns (in accordance with our lexical conventions, we ignored information regarding verbs, adverbs and adjectives).

Various kinds of links connect these categories: `specialization`, `exclusion`, `similar`, `member`, `part`, `substance`, and their inverse links. The interpretation of links other than `specialization`, `exclusion` and `similar` are not always clear nor consistent within Wordnet. For example, a `part` link from the category `airplane` to the category `wing` could mean that "any airplane has for part at least 1 wing" or "all airplanes have for part the same wing", "any wing is part of a plane", etc. We assumed the first interpretation was correct for direct links (e.g. `part`, `substance`, etc.) and therefore opposite for their inverse links (`part of`, `substance of`, etc.). This interpretation is exploited in our graph comparison/retrieval algorithms.

To permit the use of WordNet in a KB server, we have (i) generated a unique identifier for each category, using the most commonly used word for that category, whenever it was possible; (ii) distinguished the Wordnet `specialization` links into `subtype` links and `instance` links by isolating about 6000 individuals, and (iii) re-structured and complemented WordNet top-level ontology with about 150 concept types, and 200 relation types. Thanks to this re-organization and our ontology checking mechanisms, we detected about 300 semantic errors (e.g. categories specializing exclusive categories, redundancies, subsumption links used instead of part-of links or member-of links, etc.) and manually corrected them. We also made about 500 lexical corrections. (See www.webkb.org/doc/wn/ for details).

WordNet is also used in other knowledge-based systems, e.g. AI-trader[72], a knowledge base broker, and Ontoseek[4], a knowledge retrieval system. Both permit their users to enter "simple" CGs (i.e. existentially quantified and without contexts) for representing knowledge, and permit "queries for specializations of a query graph" for retrieving knowledge.

[71] http://www.cogsci.princeton.edu/~wn/

[72] http://www.vsb.informatik.uni-frankfurt.de/projects/aitrader/intro.html

3.7 Need for More Centralization

According to Tim Berners-Lee[73], "many KR systems had a problem merging or interrelating two separate knowledge bases, as the model was that any concept had one and only one place in a tree of knowledge ... The RDF world, by contrast is designed for this in mind, ...". Although RDF schemas may indeed import other RDF schemas and RDF documents may import various RDF schemas, in order to compare two statements from different RDF documents, an RDF engine has to classify the categories used in these statements into a *unique* specialization hierarchy[74]. This is most often impossible (unless the two documents mostly re-use the same schemas) because of the disconnected specialization hierarchies (and hence insufficient information to compare the categories and statements). What are currently called "ontology-merging techniques", are only semi-automatic algorithms heuristically matching categories based on their names, links to other categories[75], and sometimes other properties such as their frequency of occurrence in documents[13].

From the knowledge provider's view, re-using distributed RDF schemas is also a difficult and sub-optimal task. First, she must find schemas on the Web with categories similar to the ones she wants to use, then select some schemas that are not mutually inconsistent and write another schema to define the categories she has not found. Tools exploiting distributed schemas cannot provide guidance nor much cross-checking since they do not have a large ontology to exploit. In WebKB-2, thanks to the initialization of the KB with WordNet, the user enters a word and is presented with the categories that represent its various meanings, generalizations and specializations. She can select one category or find a more appropriate category by navigating along semantic links. When a new category is required, the user can add it by connecting the new category to an existing category via a link of a selected type. Since the new category is added to a large and tightly interconnected ontology, it can be accessed and exploited in many ways. With distributed schemas, to achieve a similar level of connectedness, *each* schema creator would have to check that there is a relation between *each* of her categories and *all* relevant categories in *all other* existing schemas on the Web.

There is an intermediate way between the highly decentralized approach advocated by the W3C and the approach we have adopted. That is to develop RDF schemas/documents by re-using (importing) ontologies of large KB servers such as WebKB-2. Then, tools could provide *some* guidance and cross-checking, and do a relatively good job at integrating these schemas/documents even when developed separately since they would at least be based on the same large natural language ontology. WebKB-2 permits its categories or parts of its ontology to be referred and accessed via URLs and can import knowledge from Web documents into its shared KB, permanently or for testing puposes. However, the constraints are the same as when knowledge is entered manually, and an import is rejected if a problem is encountered. With the intermediate way, future Web search engines will have to be more permissive.

---

[73] http://www.w3.org/DesignIssues/RDFnot.html

[74] Not simply a tree since a category may have several parents.

[75] See Chimaera at http://www.ksl.Stanford.EDU/software/chimaera/

4 Mechanisms for Cooperatively Editing a Shared KB

In addition to the previous requirements, within a KB server, protocols are needed to permit the cooperative building of a KB and maximise the re-use, inter-connection and hence later retrieval of the knowledge representations. We have mentionned the approach used in the Co4 system but noted that it unlikely scales to large KBs. It also does not encourage a tight interconnection between the knowledge of the various users. Thus, we now describe our approach.

The WebKB-2 user is asked to be as precise as possible when making statements in order to avoid conflicts in the KB and permit to answer queries more adequately. For instance, a user (say "user1") should not simply represent that "birds fly" (in FCG: `[user1#birdsFly [any bird, agent of: a flight]]`) since this is not always true. If this happens, other users are encouraged to "correct" this representation. In WebKB-2, any user can do this by creating a new graph that connects the "faulty" graph to a more precise version using a relation of type `pm#corrective_specialization` (then, depending on display options, the first version may be filtered out by WebKB-2 when responding to queries). Similarly, if a user thinks a statement from another user can be generalized, she can use a relation of type `pm#corrective_generalization`. For example, if "user1" stated that "birds fly" and "user2" wants to correct and specialize that by "a study made by Dr Foo found that in 1999, 93% of healthy birds could fly", she can write:

```
[user1#birdsFly, corrective_specialization:
   [user2#93pcOfHealthyBirdsCanFlyAccordingToFoo
       [ [93% of (bird, experiencer of: a good health),
            can be agent of : a flight
         ], time: 1999], source: (a study, author: Foo@bird.org)]
   ]]
```

(Note: if a graph is not explicitly named, WebKB-2 generates a name for it).

We believe that a scalable approach for cooperation between users of a knowledge base server implies two goals: (i) each user should be able to represent what she considers true, correct or complement other users' knowledge in a non-destructive manner. She should be able to use the categories and names she wants – providing that lexical recommendations are respected and existing categories re-used or specialized – and should not have to discuss and find an agreement with other users each time a conflict arises; (ii) knowledge from different users should remain consistent and tightly interconnected to permit comparison, search, cross-checking and optimal unification across the KB.

These two goals are commonly thought to be incompatible but we have already partly shown how they can both be achieved, providing users connect their categories and graphs to other existing ones. What remains to be presented is the set of *removal/modification/addition protocols required for semantic conflicts to be managed asynchronously and without person-to-person agreement*. The following four points describe our approach.

1) A user may **remove a category, link or graph** only if she has created it and unless this removal induces an inconsistency in the user's knowledge. If the category, link or graph being removed is used by other users or is necessary for

their knowledge to remain consistent, it is actually not removed from the KB but its ownership is changed to one of the users relying on its existence. Inconsistency detection in WebKB-2 currently only exploits relation signatures, exclusion links and specialization links. However, we plan to exploit inconsistencies detected by users and signaled by users with a relation of type `pm#contradiction` between two graphs.

2) The creator of a category may **modify a link** connected to this category – so that the link uses an alternate category – unless this modification itself induces an inconsistency. The creator of a relation type may modify its signature unless such change induces an inconsistency (in which case, she must first modify the ontology or related graphs so that the inconsistency disappears). A user may **not modify a graph** that she has not created, but she can connect it to another graph via a relation of type `pm#corrective_specialization`, `pm#overriding_specialization`, `pm#corrective_generalization` or, if none of the previous ones apply, `pm#correction`. This last relation type should also only be used if the ontology cannot be modified to correct the first graph. Since graphs can be used for representing links, these three relation types may also be used by a user to "correct" links between categories. Depending on display/filtering options, corrected graphs or links may be displayed/used for inference or not.

3) A user may **add a graph or a link**, even if she is not the creator of the linked categories, unless this addition introduces an inconsistency or redundancy. For consistency and re-use purposes, WebKB-2 does not accept a graph that already has a specialization or a generalization in the KB; this feature is detailed in the next subsection. When this happens, the user must either refine her graph before trying to re-add it, modify the ontology or use one of the four "corrective" relations cited above.

4) In any of these previous cases, when the knowledge of a user is modified by another user, the change should automatically be e-mailed to the first user or presented the next time she logs onto the KB server.

An alternative approach would be to *always allow* the creator of a category to add, modify or remove categories or links she has created *even when* that change induces an inconsistency in other users' knowledge. Under this scheme, the inconsistency would have to be repaired automatically. Since the update means a change of interpretation of a category (at least from the viewpoint of other users), one way to repair the inconsistency is to "duplicate" the categories and links that should not be modified in order to avoid the inconsistency (i.e. the modified category and some of its subtypes from the same user). The "duplicates" are then attributed to other users whose knowledge depend on them. Algorithms for this duplication have been detailed previously[76]. Although this approach would allow each user to *ignore* how her categories are used by other users, it is less optimal than manual corrections, reduces cooperation between users and the tight interlinking of their knowledge. This approach would also be complex to implement and could not be extended to handle graph modifications in a similar manner.

[76] http://www.webkb.org/doc/PhD.html

4.1 Control on Graph Additions

The WebKB-2 user may not add a graph `g1` if it contradicts, generalizes or specializes an existing graph `g0`, without connecting `g1` to `g0` via a relation of type `pm#corrective_generalization`, `pm#corrective_specialization`, `pm#correction` or `pm#overriding_specialization`. *There is one exception*: when `g1` instantiates `g0`.

For example, consider Fig. 1 where some statements are represented in Formalized English (FE) and exclusion/specialization/instantiation relationships between them are given. A user is not allowed to enter "no bird can be agent of a flight" or "2 birds can be agent of a flight" if the statement "at least 1 bird can be agent of a flight" is already present in the KB. Assuming its identifier is `pm#AtLeast1birdCanBeAgentOfFlight`, the user should enter:

```
pm#AtLeast1birdCanBeAgentOfFlight has for corrective_specialization
  'no bird can be agent of a flight'
```
or:
```
pm#AtLeast1birdCanBeAgentOfFlight has for correction
  '2 birds can be agent of a flight'.
```

However, a user may enter "Tweety can be agent of a flight" even if the statements "2 birds can be agent of a flight" or "any bird can be agent of a flight" already exist in the KB because this is what we call an "instantiation": the new graph simply gives an example or occurence of a more general statement (there is no potential conflict between the authors' respective intentions).

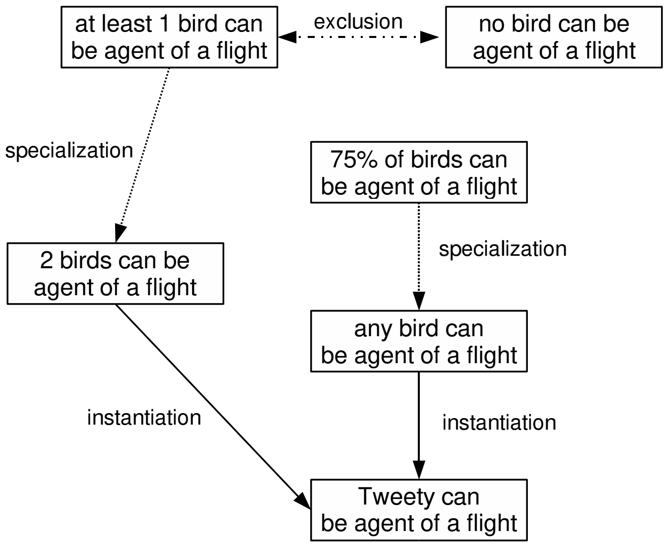

Fig. 1. Explicit connections between graphs are required when exclusion/specialization (but not instantiation) relationships are discovered by WebKB-2.

5 Search Interfaces and Mechanisms

Knowledge servers of the Semantic Web, i.e. large-scale multi-users knowledge server, need to be usable both by knowledge engineers (or software agents exploiting knowledge) and average Web users. Although the first group requires various options to search, filter and browse the ontology and statements, an average Web user only needs to find the right category for the object she has in mind; she should not have to update the ontology apart from sometimes introducing a new category simply by giving it a type or a supertype. Both novices and experts need guidance when entering statements in order to ease the knowledge representation task and permit the production of *explicit and comparable* statements.

The interface of Ontosaurus and the Ontolingua editor do not ease the comprehension of (portions of) the KB since relations from an object are not easily explorable on more than one level of depth, and there is no filtering options available. Graphical editors for graphs or links between categories, as for example in Ontobroker, are certainly more appealing to novice users than indented lists and graph linearizations but take a lot of space on the screen (which limit the quantity of information than can be displayed and impose many scrolling or browsing), generally require the users to download special librairies, are slow to load and execute, permit to view only one graph, and rarely have the facilities that comes for free with textual versions: the possibility to mix graphs with (or hyperlink them to) images, textual elements, or other graphs in documents, the possibility to re-use by copy-paste, to search via lexical search and more generally to be readily parsable by other applications. Ontorama[77] is an hyperbolic viewer that permits the browsing of subtype links in the WebKB-2 ontology but, as other similar viewers, is of little practical interest for knowledge engineers working on such a large KB.

In this section, we show some of the interfaces of WebKB-2, for average Web users and knowledge engineers. We also show why and how mechanisms for "classic search for specializations of a query graph" [12] need to be extended to permit a full "search for path specialization". Such search mechanisms are both powerful and easy to use for knowledge retrieval, and can be tractable[78]. They were also used in Algernon[79], an inference system based on a tractable reasoning system called Access-Limited Logic [2].

All the queries or assertions that can be made via the WebKB-2 interface can also be made by any application over the Web via a GET or POST HTTP request and with the same language of commands.

77 http://www.webkb.org/ontorama/

78 If the query graph and each of the statements has a tree structure (including sets and contexts within each statement), the search complexity is polynomial (see [1]). In WebKB-2, coreference variables may be used within the statements, thus introducing cycles. Furthermore, the matching algorithms use a simple depth-first exploration with controls to avoid loops. Hence, they do not have a polynomial complexity. However, given users' queries and statements are generally without cycle (and small), this is not detrimental in practice.

79 http://www.cs.utexas.edu/users/qr/algernon.html

5.1 Searching Categories and Links

Fig. 2 shows the interface for knowledge engineers to search categories or links. It proposes various selection options (names, kinds of connected links, kinds of creator or non-creator) and format options (recursive exploration, language, hyperlinking). The counterpart of this interface for average users is a simple text field (to enter a word, regular expression or directly a category identifier); it is proposed in the WebKB home page. Fig. 3 and Fig. 4 show the result of the query in Fig. 2, i.e. a search for categories with the name "person".

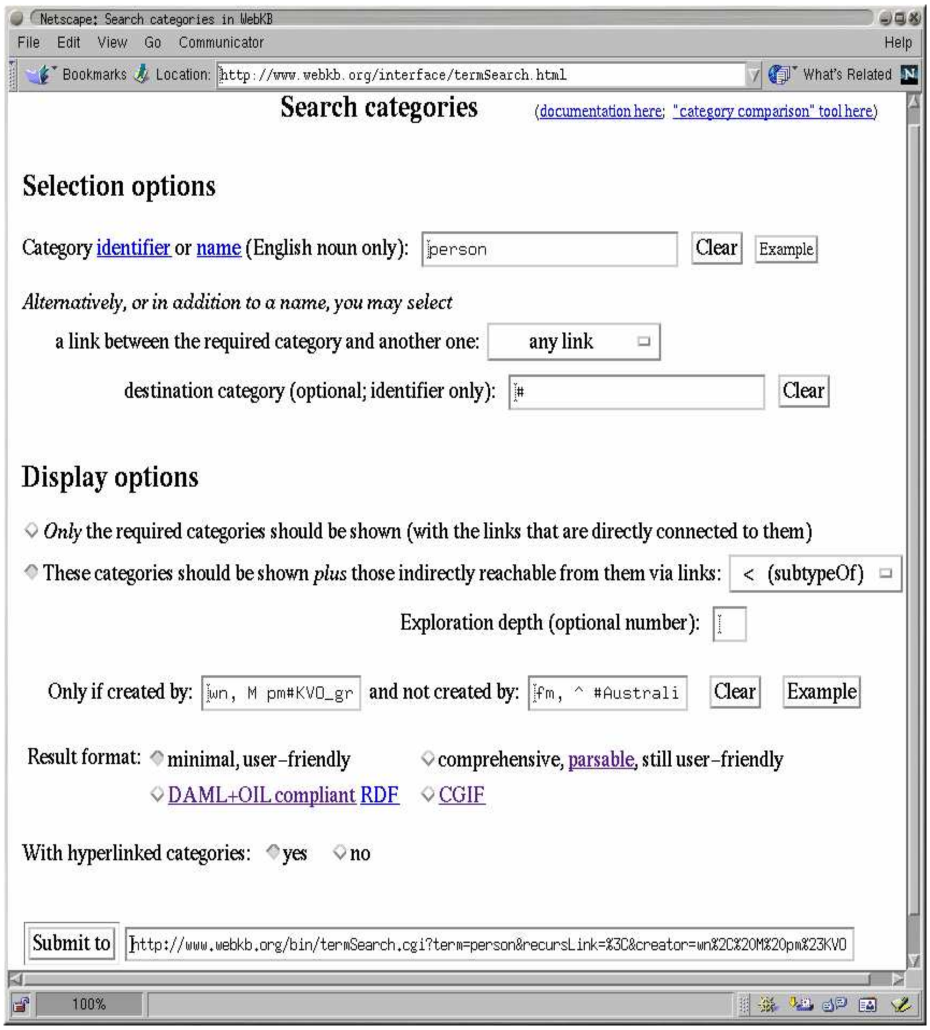

Fig. 2. Query for links and graphs related to #person and created by WordNet (wn) or a member of KVO (M pm#KVO_group) but not by F. Modave (fm) nor an Australian (^ #Australian); subtypeOf links must be recursively explored.

Fig. 3. Result of the previous query (Fig. 2).

Fig. 4. Result of the previous query (Fig. 2) for "novice users".

5.2 Accessing or Adding Graphs Via Generated Interfaces

Fig. 3 and Fig. 4 show that graphs directly or indirectly using a category are accessible from this category (or a confirmation that no graph uses this category). Each category identifier (even when shown within a graph) is displayed hyperlinked to permit access to its related links and graphs. Most link identifiers are also hyperlinked to ease the exploration of the KB. Hyperlinks to search/add forms are also given (e.g. see "click here for a search form" in Fig 4).

These forms are generated based on the schemas (general statements) associated to the category or its supertypes. Fig. 5 shows the form generated to guide the addition of a statement about a new or already registered user. The three schemas exploited for this purpose are shown in Fig. 3. The directives `$(no inheritance)$` and `$(explore)$` stored in the concept node annotations control the generation of the form. The first directive prevents the use of schemas associated to supertypes of the category. The second leads to the generation of an hyperlink to another form for detailing a related object. In other words, this second directive permits the re-use of schemas related to related objects to enable form cascading. Fig. 6 illustrates such a cascade. `$(explore)$` is also used to control the depth of menus generated using subtype partitions (e.g. the categories for colors and for days of the week are organised into hierarchies of subtype partitions; such partitions permit WebKB-2 to generate organized menus and filter categories likely to be less relevant).

These forms guide and ease knowledge capture. Since they normalize knowledge capture, they also lead to more comparable statements. At present, schemas in WebKB-2 are mostly associated to top-level concept types (e.g. `pm#situation`, `pm#description` and `pm#physical_entity`). These schemas are inherited by all types in the ontology that have no overriding schemas. They include the most useful relations from a certain object, permitting the user to ignore less precise relation types imported from other ontologies or relation types with structural purpose only (e.g. `pm#relation_from_spatial_entity`). As Fig. 5 shows, each form also has a field to permit the use of relation types not listed in the form.

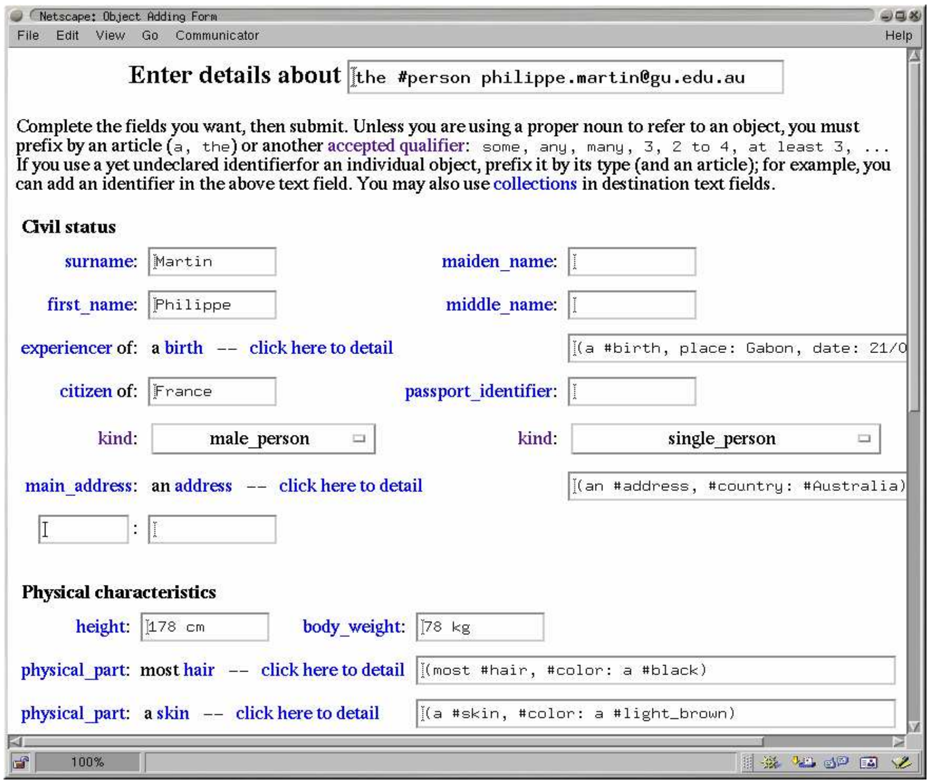

Fig. 5. A generated form to enter a statement about a new/existing person. The schemas shown in Fig. 3 are used to generated it. The knowledge provider must enter its identifier and password at the end of the form.

Fig. 6. Form called from the form in Fig. 5 to enter information about an address.

To guide and facilitate the representation of knowledge by average users, many specialized schemas are also required, e.g. for "house", "car", "selling", "renting", etc. Users may also create and associate schemas to any category: a schema is simply a statement that uses a general quantifier ("any", "most", "20%", ...) in the first concept node.

When a form is submitted, WebKB-2 generates a graph with the information (see Fig. 7). If this graph does not violate the syntax/semantic/cooperation rules, and if all the category names it contains can be unambiguously resolved to category identifiers, it is entered into the KB. The creation date and the graph identifier are automatically generated and added to the graph.

Search forms are similar to knowledge capture forms above except that the generated command is not a graph assertion but a query graph.

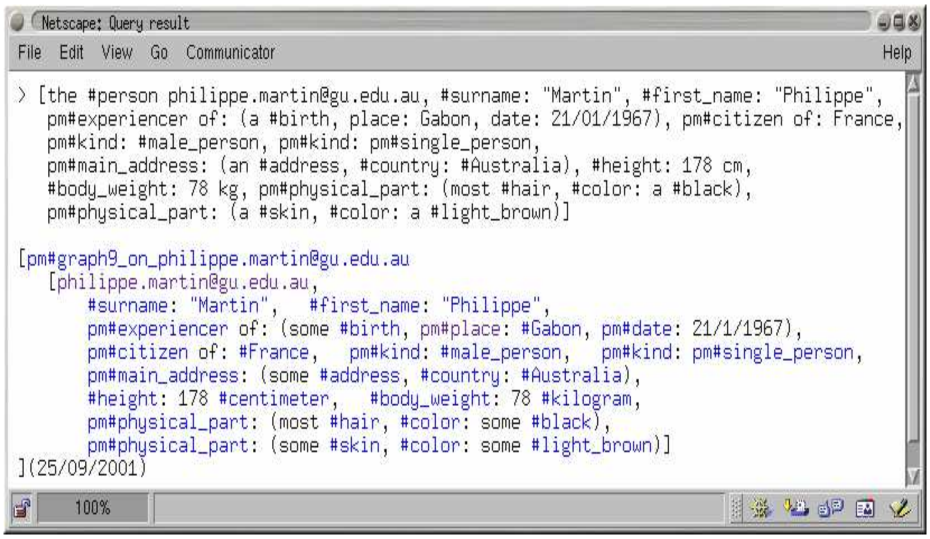

Fig. 7. Command (FCG) generated when the form of Fig. 5 is submitted.

5.3 Mechanisms for Searching Graphs

Classic searches for specializations of a query graph permit searches "by the content". However, they need to be extended for more flexibility in the formulation of the query graph and to increase the number of relevant answers. WebKB-2 uses four extensions.

1) Let us assume the KB includes the graphs `[John, owner of: a car]` and `[John, owner of: an appartment]`. A classic search for graphs specializing the query graph `[a man, owner of: a car, owner of: a lodging]` would not retrieve the previous graphs since only the union of these specializes the query graph. When WebKB-2 tests if a graph `g` can be a specialization of the query graph, it also looks for more information in graphs related to `g` by a same individual (same identifier of coreference variable), or that use a type in `g` with a universal quantifier (with an existential quantifier, there may not be any connection), or that define necessary conditions for a type in `g`. If `g` plus some related graphs permit to answer the query graph, they are displayed separately: joining them would often not produce a meaningful graph (e.g. their embedding graphs could not be joined). As another example, two other graphs that could be presented in answer to the previous query are:

```
[ [[Tom \\IBM_employee, owner of: an apartment], time: 2000], author: Tom]
[ [any IBM_employee, owner of: a car], author: IBM]
```

2) Searches should take into account knowledge represented via links instead of graphs.For instance, let us assume the categories representing the geographical areas "Gold Coast" and "Southport" are connected via a `part` link and the knowledge base includes the following graph.

```
[philippe.martin@gu.edu.au, agent of: (the renting,
         object: (an apartment, part: 1 bedroom, location: Southport),
         instrument: 140 Australian_dollars, period: a week,
         beneficiary: Spirit_Of_Finance)]
```

WebKB-2 exploits the ontology to present this graph in answer to the query graph `[an apartment, location: (a district, part of: Gold_Coast)]`.

3) Let us assume the graph `[John, owner of: a lodging]` is in the knowledge base and a query graph is `[a man, owner of: an apartment]`. The first graph is not a specialization of the query graph since `wn#housing__lodging` is a supertype of `wn#apartment__flat` not the reverse. However, a user may want such a graph to be provided. This is why WebKB-2 provides two graph search commands: "spec" to search specializations of the graph given in parameter, and "?" to search graphs *comparable* to the one given in parameter. With the second command, supertypes of categories in the query graph are also used. The first graph would not answer the query "`? [a man, owner of: a bike]`" since `wn#housing` is neither a subtype nor a supertype of `wn#bicycle__bike`.

4) Structural flexibility should be permitted in query graph specification. We believe the simplest way (both for the user and from an implementation perspective) is to allow the specification of path sequences with common regular expression operators ("*" for "0, 1 or many times", "+" for at "at least 1 time", "?" for "0 or 1 time"). Let us assume the following graph is in the KB:

```
[philippe.martin@gu.edu.au, agent of:(a research, within_group: KVO_group)]
```

Users looking for a person conducting research at "Griffith Uni., Gold Coast campus" are unlikely to find this graph via classic searches for specialization only. However, since `pm#School_of_IT_at_Griffith_Uni_Gold_Coast_Campus` is connected via a `part` link to `pm#KVO_group` and via a `location` link to `QLD#GCcGU__Gold_Coast_campus_of_Griffith_Uni`, and since `pm#relation` is the uppermost relation type, it should be possible to find this graph with:

```
    spec [a person, agent of: (a research, relation+: GCcGU)]
or: spec [a research, (relation: a thing)+ location: GCcGU]
or: spec [a research, relation 3+ (part of: a group)3+ location:GCcGU]
```

("3+" means that at most 3 relations of the specified type should be traversed).

Fig. 8 shows one of WebKB-2's interfaces for searching graphs. Names, instead of category identifiers, have been used and "pm" has been specified as the creator of the graphs to retrieve. Fig. 9 shows the result. It first indicates that two categories share the name "Gold_Coast" and that the first has been selected. Then, a graph answering the query is displayed, with its categories hyperlinked.

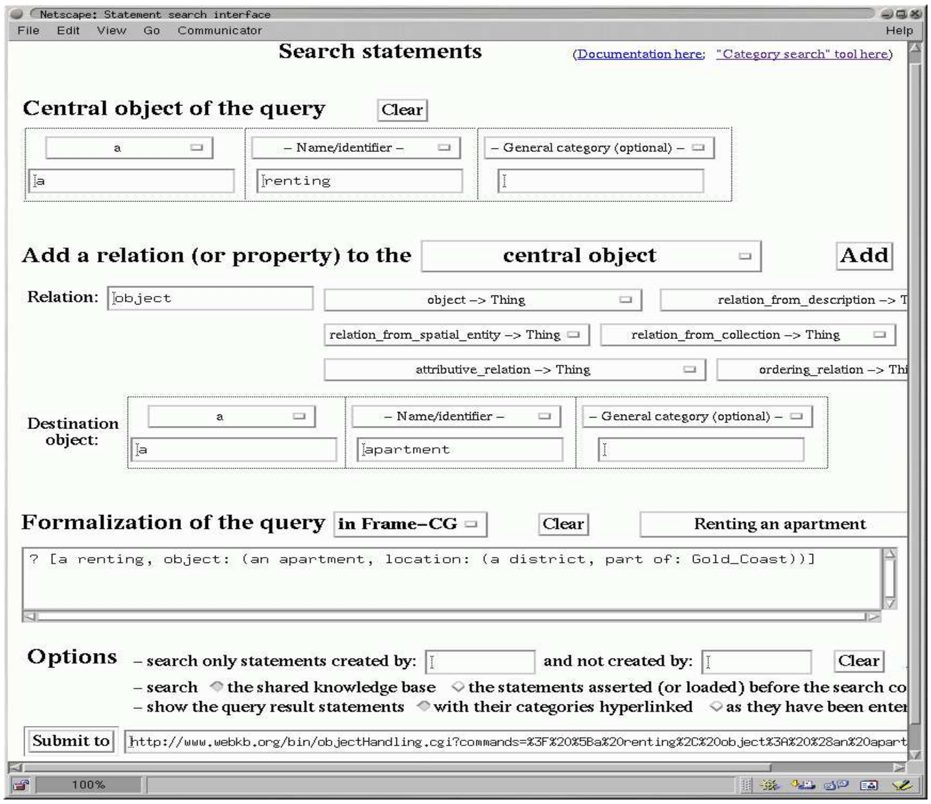

Fig. 8. Query for the specializations of a graph

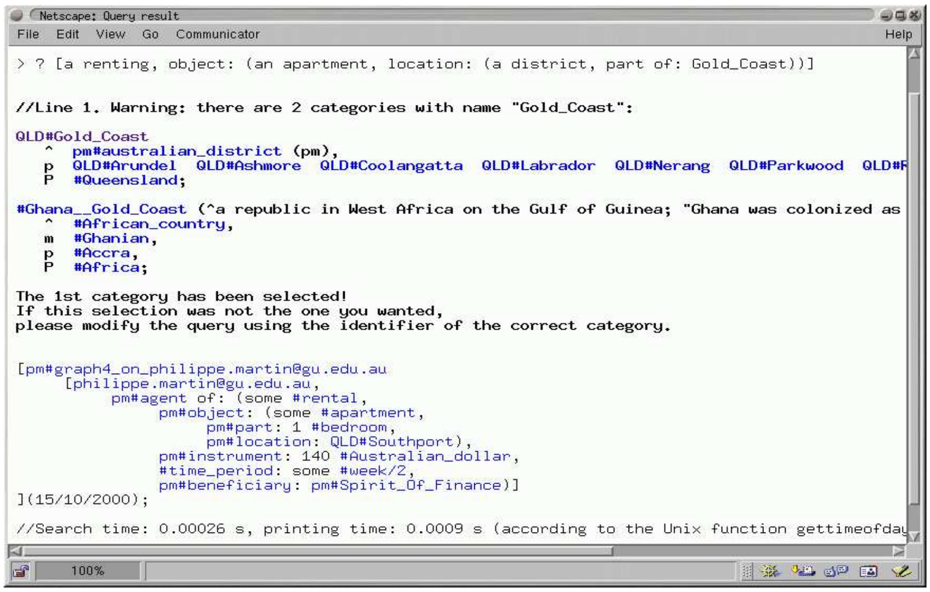

Fig. 9. Result of the previous query

6 Conclusion

This article has presented elements helping the realization of the goals of the Semantic Web, i.e. "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"[80].

This paper first listed important Semantic Web related projects, and elements required for knowledge sharing within a KB as well as across the Web: a library of ontological primitives, an ontology of natural language, lexical/structural/ontological conventions, and high-level expressive notations supporting them. Then, it was shown that knowledge-based servers could further ease the knowledge representation task, improve cooperation between knowledge providers, knowledge retrieval and re-use.

Three paradigms have been emphasized: (i) a more centralized approched can be adopted to solve the problems of the "highly distributed" approach advocated by the W3C without loosing any of its advantages, (ii) as far as knowledge is concerned, "the more the better", be it the number of conventions, the size of standard ontologies, the size of the KB, the expressivity of the languages and the precision of the representations (in all cases except the first, some information can easily be ignored or automatically filtered out when not needed), (iii) global approaches (i.e. module/file based) are more coarse-grained than local approaches (inter-connections between elements) and hence less precise/explicit and flexible when complexity increases.

Three goals also describe the presented aproaches: ease of representation, scalability and possibilities of use and re-use. These goals converge: knowledge capture is a well recognized bottleneck, and knowledge use/re-use is both a goal and a method.

Entering information in WebKB-2 or a similar KB server is more difficult than entering sentences in a document, but information from documents cannot be automatically retrieved and interconnected to respond to precise queries. We believe that entering information in WebKB-2 is easier than in most other systems thanks to our ontologies, notations and features (generated menus, the possibility to use everyday words instead of category identifiers, etc.). Some kinds of information remain difficult to represent precisely but we think that WebKB-2, or some evolution of it, can be used by Yellow-Pages-like-services or community servers to allow people to advertise products and services or, more generally, publish knowledge. In summary, it may be seen as a prototype for Semantic Web servers.

Acknowledgments

This work is supported by a research grant from the Distributed Systems Technology Centre.

[80] http://www.w3.org/2001/sw/

References

1. M. Chein and M.L. Mugnier, "Positive Nested Conceptual Graphs", *Proc. 5th Int'l Conf. on Conceptual Structures* (ICCS 97), Springer Verlag, LNAI 1257, pp. 95–109.
2. J. M. Crawford and B. J. Kuipers, "Algernon – a tractable system for knowledge representation", *SIGART Bulletin* 2(3), June 1991, pp. 35–44.
3. D. Fensel, S. Decker, M. Erdmann and R. Studer, "Ontobroker: Or How to Enable Intelligent Access to the WWW", *Proc. 11th Knowledge Acquisition Workshop* (KAW98), Banff, Canada, April 1998, pp. 8–23. ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/OB.KAW.ps
4. N. Guarino, C., Masolo and G. Vetere, "Ontoseek: Content-based Access to the Web", *IEEE Intelligent Systems*, Vol. 14, No. 3, 1999, pp. 70–80.
5. J. Hendler, "Agent and the Semantic Web", *IEEE Intelligent Systems*, Vol. 16, No. 2, 2001, pp. 30–37.
6. K. Knight and S. Luk, "Building a Large-Scale Knowledge Base for Machine Translation", *Proc. of the 12th national conference on artificial intelligence* (AAAI'94), Seattle, USA, July 1994. http://www.isi.edu/natural-language/resources/sensus.html
7. ph. Martin, "Using the WordNet Concept Catalog and a Relation Hierarchy for Knowledge Acquisition", *Proc. of Peirce'95*, 4th International Workshop on Peirce, Santa Cruz, California, August 18, 1995. http://www.inria.fr/acacia/Publications/1995/peirce95phm.ps.Z
8. P. Martin and P. Eklund, "Embedding Knowledge in Web Documents", *Proc. of the 8th Int'l World Wide Web Conference* (WWW8), Toronto, Canada (1999). http://www.webkb.org/doc/papers/www8/www8.ps
9. P. Martin and P. Eklund, "Conventions for Knowledge Representation via RDF", *Proc. of WebNet 2000*, ACCE press, pp. 378–383. http://www.webkb.org/doc/papers/webnet00/
10. P. Martin and P. Eklund, "Large-scale cooperatively-built heterogeneous KBs", *Proc. 9th Int'l Conf. on Conceptual Structures* (ICCS 01), Springer Verlag, LNAI 2120, 2001, pp. 231–244. http://www.webkb.org/doc/papers/iccs01/iccs01.pdf
11. G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller K., "Five Papers on WordNet", *CSL Report 43*, Cognitive Science Laboratory, Princetown University, July 1990. http://www.cogsci.princeton.edu/~wn/papers/
12. J.F. Sowa, "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley, Reading, MA, 1984.
13. G. Stumme and A. Maedche, "FCA-Merge: A Bottom-Up Approach for Merging Ontologies", *Proc. 5th Int'l Joint Conference on Artificial Intelligence* (IJCAI 01), Morgen Kaufmann, Seattle, USA, August 2001, pp. 1–6.