

# BETWEEN TOO INFORMAL AND TOO FORMAL

[Dr Philippe Martin](#)<sup>1</sup>, [Dr Mike Eboueya](#)<sup>2</sup>, [Dr Jun Jo](#)<sup>1</sup>, [Dr Lorna Uden](#)<sup>3</sup>

1: Griffith University (School of ICT), 2: La Rochelle University, 3: Staffordshire University  
e-mail: pm at phmartin dot info

## ABSTRACT

This article presents various elements of solutions to encourage and permit people to provide, share, retrieve, update and annotate information in more precise, structured and normalized ways within a knowledge repository (which therefore is not a classic informal repository based on documents or supported by a database but it is a knowledge base which can include or index informal elements and thus can complement classic information repositories). Those elements, which are implemented in the knowledge server WebKB-2, relate to syntaxes, querying and comparison mechanisms, cooperation protocols or approaches, and ontologies. The provided examples focus on the representation and comparison of tools, techniques and ideas related to knowledge engineering. The acceptance of our approach by non-technical people remains to be tested.

## 1 INTRODUCTION

Some information repository projects use or intend to use formal knowledge bases (KBs), e.g., the [Open GALEN](#) project which has created a KB of medical knowledge, the [QED Project](#) which aims to build a "formal KB of all important, established mathematical knowledge", and the [Halo project](#) which has for long term goal the design of a "Digital Aristotle". However, even when not aimed to support problem solving, designing formal KBs is inherently a difficult and time-consuming exercise that current generic KB systems (KBSs) still do not guide well. Hence, despite the benefits of a formal approach, it is very rarely used for creating information repositories and especially corporate memories.

Instead, information repositories are most often composed of informal documents that are independently created by people (typically by publishing a Web document or sending an email to a mailing list). This approach is simple but the well-known drawback is that it is then often difficult to retrieve or compare information because (i) the various needed pieces of information are scattered within many documents and expressed in different ways and at often inadequate levels of detail, (ii) these pieces of information cannot be automatically organised into a semantic network by any current (or even currently foreseeable) natural language understanding (NLU) technique. The use of cooperatively edited informal documents (as in wikis) helps to reduce the scattering of information but introduces new problems and does not by itself lead to better or sufficient structuring of the information. Structured documents (e.g., documents following an XML schema), databases and application-oriented interfaces enforce some structure but (i) they also often restrict what can be entered even when they provide "free text" entry fields, and (ii) the semantic of the prescribed structure is most often left implicit and is insufficient to be used by KBSs. The manual or automatic use of lightweight metadata for indexing (parts of) documents (e.g., [Dublin Core](#) metadata, informal categories such as those of the [Open Directory Project](#), general categories such as those of [WordNet](#) or specialized categories such as those from the [KA2 project](#) and some other [Semantic Web](#) projects) only permits document retrieval and does not lead to any browsable semantic network synthesizing and comparing facts or ideas. The occasional use of semantic relations (as in semantic wikis) is also insufficient for "knowledge retrieval" (i.e., precision-oriented information retrieval) or, more generally, automated reasoning. The use of poorly structured, graphical and overly permissive semi-formal notations such as those used for [Concept Maps](#) (or their ISO version, [Topic maps](#)) often lead to information that are more difficult to understand, retrieve and exploit than when regular informal sentences are used (commented examples about this are given in [\(Sowa, 2006\)](#)). [Controlled languages](#), i.e., semi-formal languages that look like natural languages but have a restricted syntax or a restricted vocabulary, are often seen as good compromises between formal and informal languages; however, (i) they do not scale (i.e., in the general case, they are not expressive and formal enough or become too complex to use when extended), and (ii) there are often more structured, precise, normalised and readable ways to express knowledge (for example, partOf or generalization hierarchies).

The solution advocated in this article is to provide people various kinds of supports that encourage and permit them to be as precise or formal as they are willing to be, and to re-use, complement, annotate or correct each other's knowledge. One approach is to provide these supports via a knowledge server such as [WebKB-2](#) (Martin, 2003a); this leads to cooperatively updated semi-formal KBs which should be more and more precise as the number of users grows. Another implementation approach would be to re-use a peer-to-peer network but this would be more complex to achieve and would offer no theoretical advantage except for privacy issues (the managers of a central repository *can* access any piece of information without the consent of its provider). The following sections present the kinds of supports we propose (they are implemented or being implemented in WebKB-2). Section 2 illustrates various readable, expressive, normalising, formal and semi-formal notations. Section 3 discusses needed features of querying and comparing methods. Section 4 summarizes protocols to support cooperation between knowledge providers and also a method to value contributions and contributors. Section 5 shows an integration of lexical and top-level ontologies, and proposes a way to modularise inputs within information sources. This is a

synthesis article which makes explicit and relates the ideas behind several of our works separately published at other conferences, and which also introduces recent refinements. WebKB-2 can complement classic information repositories and tools such as KPMG's K-World by offering additional or alternative ways to enter, index and retrieve some of their content. Although non-technical people may find a knowledge-oriented approach difficult to understand at first, it is not actually difficult and its adoption can be incremental.

## 2 VARIOUS READABLE, EXPRESSIVE AND NORMALISING NOTATIONS

### 2.1 General Notations

Research in knowledge representation has focused on reasoning and therefore not on creating *notations that are both very readable and expressive*, although Sowa (1984) had this in mind when creating the Conceptual Graph (CG) model, its linear form (CGLF) and display/graphical form (CGDF). His relative success is what still makes much of appeal of the CG formalism and approach (a related reason is that, being higher level than other notations, the CG notations lead to more "normalised" knowledge representations which therefore ease knowledge sharing and retrieval). It is however possible to go further in terms of "high-level-ness", i.e., readability (concision and/or intuitiveness) and normalising effect (restricting the number of automatically incomparable ways something can be represented). Martin (2002) did so by creating the Frame-CG (FCG), Formalized-English (FE) and For-Links (FL) notations. [All have LALR\(1\) grammars](#). FL is discussed in the next section. FE looks like some pidgin English but is structurally equivalent to FCG which is an extremely concise notation that includes constructs for extended quantifiers, meta-statements, functions and various interpretations of sets (hence, it is semantically equivalent to [KIF](#)). FE is quite verbose and hence is not adequate for really building or browsing a reasonably complex KB but it can be shown to anyone. Hence, it can for example be used for showing the various interpretations that a NLU parser makes of a sentence expressed in a natural or controlled language and then let the user select the right interpretation or precise the sentence.

The following example illustrates the representation of an English (E) sentence in FE, FCG, CGLF, PL (predicate logic), KIF, RDF (more precisely, [RDF/XML](#)) and [LTM](#) (the Linear form of Topic Maps). Since numerical quantifiers (here, "2" and "3") must be used, no current controlled language (apart from FE) can represent such a sentence correctly, and any RDF representation is ad-hoc (furthermore, no representation in [N3](#) is possible). Except for the LTM representation (which cannot be made more precise), all these representations are formal if the used terms (e.g., "car") are formal, i.e., declared and possibly defined. However, the 's' at the end of "cars" and "sells" is a lexical facility offered by WebKB-2 for FCG and FE when a universal-like quantifier (e.g., "any", "2" and "3") is used: in such a case, WebKB-2 automatically removes the 's'. The selling act is represented via a concept (instead of a relation as in the LTM representation) to allow the quantification (with "2" here) and the use of the relation "time" (that is, more than one relation). Hence, to permit knowledge comparison, all representations of selling acts should use the concept type "sell". More generally, all actions should be represented via a concept type. FE and FCG lead to such a normalisation. Most languages (e.g., KIF, RDF and LTM) do not. As opposed to these languages, FE and FCG have many features such as the numerical quantifiers and time data-types illustrated below which, since the users do not have to define them, help knowledge sharing.

```
E:      Ned sold (the same) three cars twice on the 21/1/2001.
        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 below.

FE:      3 cars are object of 2 sells with agent Ned and time 21/1/2001.
        If there had been only 1 sell, "Ned sells 3 cars with time 21/1/2001" could have been used
        in FE as a shortcut for "Ned is agent of a sell with object 3 cars and time 21/1/2001".

FCG:     [3 cars, object of: (2 sells, agent: Ned, time: 21/1/2001)]

CGLF:    [Person: Ned]<-(agent)<-[Sell: {*}@2]- { <-(object)<-[Car: {*}@3 @certain];
        <-(time)<-[Date: #21/1/2001]; }

PL:      ∃cars set(cars) ^ size(cars,3) ^ ∀c ∈ cars
        ∃sells set(sells) ^ size(sells,2) ^ ∀s ∈ sells agent(s,Ned) ^ object(s,c) ^ time(s,21/1/2001)

KIF:     (forallN 3 ?c car (forallN 2 ?s sell (and (agent ?s Ned) (object ?s ?c) (time ?s '21/1/2001))))
        With: (defrelation forallN (?num ?var ?type ?predicate) :=
        (exists ((?s set)) (and (size ?s ?num)
        (truth ^ (forall (,?var) (= (member ,?var ,?s) (and (,?type ,?var) ,?predicate))))))

RDF:     <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>

LTM:     {Ned, sell, [[three cars twice]]} ~ NedSell3. {NedSell3, time, [[21/1/2001]]}.
```

## 2.2 Restricted Notations

We use the word "links" for referring to conceptual relations between individuals (e.g., statements, particular cities or persons) or non-quantified types. When knowledge representations only involve links, that is, when quantifiers and certain complex uses of sets and meta-statements need not be used, it is possible to use FL (For-Links) which is a simpler notation than FE or FCG although it is nearly as expressive as RDF + OWL-Full. One of the major advantages of FL is that it permits most of the knowledge related to an object to be represented via links from this object (and, for the major kinds of links, to avoid repeating them) instead of having to write or read separate statements. Thus, FL permits very concise and readable representations (the provided predefined links and syntactic sugar also bring a strong normalising factor). Examples are given below. FL has recently been extended to include all the necessary constructs for "structured discussions" (as illustrated below), which makes it a good (and often more expressive) alternative to the notations used in argumentation systems (e.g., AAA and gIBIS).

Several authors of hypertext systems or digital libraries have claimed that they did not use a more structured or knowledge-oriented approach because this would scare many users and thus prevent a wide adoption of their tools. For example, this claim has been made by [Shum, Motta and Domingue \(1999\)](#) to justify the lack of explicit relations between statements. Authors of argumentation systems such as AAA (Schuler and Smith, 1990) have used it to justify the restriction to a short list of predefined relation types and concept types instead of allowing people to use and update an ontology. However, none of these systems achieved wide adoption. This may well be attributed to the fact that the restrictions deeply limited what could be done with their tools, and hence their interest and applicability. The restrictions also led to biased representations and complex turnarounds. The need for user-defined typed hyperlinks for hypertext systems has long been shown and MacWeb (Nanard & al., 1993) is an example of a user-friendly and powerful knowledge-based private hypertext system. Similarly, it is a mistake to restrict the expressivity of a *general* knowledge representation language since choices about how to handle the completeness, decidability and efficiency issues, or how to handle elements such as sets and modalities, are application-dependant (e.g., for some knowledge retrieval or filtering purposes, efficient graph-matching procedures that ignore the detailed semantics of certain elements can be used, while for other purposes exploiting all the details is essential and tractability is not an issue).

```
E: Any human_body is a body and has at most 2 arms, 2 legs and 1 head.
    Any arm, leg and head belongs to at most 1 human body. Male_body and female_body are
    exclusive subtypes of human_body and so are juvenile_body and adult_body.
```

```
FL: human_body < body,
    part: arm [0..1,0..2] leg [0..1,0..2] head [1,1],
    > {male_body female_body} {juvenile_body adult_body};
```

```
KIF: (forall ((?b human_body)) (body ?b))
      (forall ((?b human_body)) (atMostN 2 '?a arm (part ?b '?a)))
      (forall ((?a arm)) (atMostN 1 '?b human_body (part '?b ?a)))
      (forall ((?b male_body)) (not (female_body ?b)))
      ...
```

In a knowledge repository, categories and statements come from multiple sources and it is necessary to record those sources to avoid lexical and semantic conflicts (details in Section 4) and allow knowledge filtering on the sources.

```
E: According to Jun Jo (who has for user id "jj"), a body (as understood in WordNet 2.0) has for part
    (as defined by "pm") at least 1 leg (as defined by "fg") and exactly 1 head (as understood by "oc").
```

```
FL: wn#body pm#part: fg#leg [0..1](jj) oc#head [1](jj);
```

```
FL: wn#body pm#part: at least 1 fg#leg (jj) 1 oc#head (jj);
```

```
FCG: [wn#body, pm#part: at least 1 fg#leg, pm#part: 1 oc#head](jj);
```

```
KIF: (believer '(forall ((?b wn#body)) (atLeastN 1 '?l fg#leg (pm#part '?b ?l))) jj)
      (believer '(forall ((?b wn#body)) (exists1 '?h oc#head (pm#part '?b ?h))) jj)
```

Below is an excerpt from a ["structured discussion" about the use of XML for knowledge representation](#), a topic which leads to recurrent debates on many knowledge related mailing lists. The parenthesis are used for two purposes: (i) allowing the direct representation of links from the destination of a link, and (ii) representing meta-information on a link, such as its creator (e.g., "pm" or "fg") or a link on this link (e.g., an objection by "pm" on the use of an objection link by "fg", without stating anything about the destination of this link). The content of the sentences and the indentation in the example below should permit the understanding of these two different uses. (Note that in this example the creators of the statements are left implicit but that prefixes such as "pm#" could be used exactly as in the previous example). The use of dashes to list joint arguments/objections (e.g., a rule and its premise) should also be self-explanatory. The use of specialization links between informal statements may seem odd but such links are used in several argumentation systems: they are essential for modularising purposes and for checking the updates of argumentation structures, and thus guiding or exploiting these updates (e.g., the (counter-)arguments for a statement also apply to its specializations and the (counter-)arguments of the specializations are (counter-)examples for

their generalizations). Few argumentation systems allow links on links ([ArguMed](#) is one of the exceptions) and hence most of them force incorrect representations of discussions. Even fewer provide a textual notation that is not XML-based. Such a notation is nonetheless necessary whenever the use of an XML parser, editor or viewer is impossible or not desirable (this is for example the case in many text-based email editors, in text-based browsers, and in PDF or HTML documents). Argumentation structures such as the ones below cannot be expected to be the "direct" result of a discussion but they may be the result of a semi-automatic re-organization of discussions and then they may be refined by further semi-formal discussions.

```
"XML is useless for knowledge representation, exchange or storage"
  argument: ("using XML tools for KBSs is a useless additional task"
    argument: "KBSs do not use XML internally" (pm,
      objection: "XML can be used for knowledge exchange or storage" (fg,
        objection: "it is as easy to use other formats for
          knowledge exchange or storage" (pm),
        objection: "a KBS (also) has to use other formats for
          knowledge exchange or storage" (pm)))
    ) (pm);

"XML can be used for knowledge exchange or storage"
  argument: - "an XML notation permits classic XML tools (parsers, XSLT, ...) to be re-used" (pm)
    - "classic XML tools are usable even if a graph-based model is used" (pm),
  argument of: ("a KRL should (also) have an XML notation",
    specialization: "the Semantic Web KRL should have an XML notation" (pm),
    specialization of: "a KRL (Knowledge Representation Language)
      can have an XML notation" (pm),
    ) (pm);
```

To conclude, for readability reasons and to support various kinds of knowledge entering or views on the knowledge, various formal and semi-formal notations should be supported but precision and normalisation should be encouraged. The above examples (and the implementation of FCG, FE and FL in WebKB-2) show that doing this while keeping the notations readable is possible.

### 3 QUERYING AND COMPARING

From any object (category or statement) of the knowledge repository, a user should be able to see and browse the directly related objects *and* also the hierarchies of indirectly related objects via user-selected transitive links (e.g., specialization and partOf links). This remains true when the objects are presented as part of a list of results to a query. Yet, most interfaces force their users to browse in order to see the direct or indirect links from an object. In a realistic KB this makes the understanding and manual retrieval or comparison of information extremely difficult (except may be for users who have a huge short term memory). This is why, although graphical views clearly have some interest, the use of textual notations such as FL and FCG are often more interesting since they permit to provide much more detail for the same amount of space. (An hopefully convincing experience for that idea is to search the various meanings of a word in WebKB-2, and then browse, for finding a concept identifier to use in a representation). Textual notations also ease manual and automatic re-use (e.g., copy-pasting, updating, parsing). Thus, ideally, both graphical and textual views should be available.

The specialization links between object classes or statements in a KB can be manually inserted or can be re-calculated after each insertion using an inference engine. Strictly speaking, a specialization link from an object A to an object B, means that B logically implies A. For example, a polynomial graph-matching operation (Chein & Mugnier, 1997) permits to determine that [Ned, agent of: (a sell, object: a car)] specializes [a vehicle, object of: a sell] and hence permits this last graph to be used as a query graph to retrieve the first. In WebKB-2, the operator "spec" can be used as in the command spec [a person, agent of: a sell], and a similar operator can be used for retrieving "extended specializations" (extended because there is not always a logical implication between the retrieved graph and the query graph); for example, extended specializations of the last query graph are [3 cars, object of: (2 sells, agent: Ned, time: 21/1/2001)] and [John, believer of: not [Ned, agent of: a sell]] (this last graph is not a strict specialization but is definitely a "relevant" information). Similarly, graph-matching operators named "comp" and "?" are respectively usable for retrieving "comparable" graphs (graphs that either specialize or generalize the query graph) and "extended comparable" graphs (graphs that are only composed of parts comparable to parts in the query graph). These operators *only* retrieve "relevant" graphs, as opposed to graph-matching operations that also consider "sibling" categories or "cousin" categories as "matching" categories. This is why we believe that the links related to these last operators (e.g., the "extended specialization" link) should also be presented whenever they are associated to an object.

Retrieving a graph may also involve using different statements that cannot be joined/merged into one, for example because one of them has meta-statements associated to it. In such a case, the retrieved graph is the set of all the required separate statements. This case often happens when path retrieval is done, that is, when the query graph involves regular expressions, as in spec [a person, agent of: (a research, (relation: a thing)+ location: Brisbane)] (assuming that relation and thing are respectively the uppermost relation type and uppermost concept type, (relation: a thing)+ refers to any non-empty sequence of relation node

followed by a concept node). This example shows that FCG allows paths and path matching to be expressed in a simple way, and thus allows to use a very powerful knowledge retrieval mechanism.

When a list of results to a query is long, it should be structured into smaller lists, for example into specialization/partOf hierarchies. If this does not provide enough structure, additional schemes should be used: for example, the results can be grouped according to common characteristics; this is a categorization task which in the general case may be difficult to solve optimally and efficiently but important concepts (such as "process" and "physical object", or at a more detailed level, "person", "civil status", "recreational activity", etc.) provide cues for natural groupings. The ontology of WebKB-2 includes this kind of information.

Although precision should be pursued, informal documents will still be created. Allowing the embedding of commands or formal statements within informal documents, whether these statements are hidden or not, and then letting knowledge servers use these documents as inputs, has several advantages: (i) the informal elements (paragraphs, images, etc.) that document the formal representations or are indexed by them do not have to be stored in separate files: these formal and informal elements can be intertwined and hyperlinked to each other; (ii) when parsing the document, the knowledge server can simply return the command results or can also copy back the informal parts that were around the commands, thus creating a "virtual document" (this is especially interesting when calls to the server can be associated to hyperlinks with predefined commands given as parameters). Furthermore, those returned results can be formal representations or, if the user prefer, the informal elements indexed by these representations. Those ideas were explored with WebKB-1 (Martin & Eklund, 2000) and most of them are also implemented in WebKB-2.

For representing certain comparisons of objects, such as the comparison of the features of certain techniques or tools, it is useful to use tables as format. Such tables can be formal or semi-formal and can be used as input or outputs. [Fact Guru](#) (Skuce & Lethbridge, 1995) is one of the rare knowledge servers that generates comparison tables. However, its approach is not scalable since the list of features/relations from the compared objects is not structured and the cells can contain a description of the destinations of the relations. A more scalable approach (Martin, 2005) is to organize the features of the compared objects into a specialization hierarchy and to use the cells only for indicating if each compared object has or has not (or will have and when) each feature. Below is an example of table generation query, followed by its result and then (next page) by the FL and FCG statements used for generating the result. The prefixes for the relations are left implicit because this leads to no ambiguity (WebKB-2 can find the correct relations). As an intermediary work for comparing the criteria of tools workings with CGs, in the [CG tools page on Wikipedia](#) we have compared seven well-known tools in the CG community and compared them according to 160 criteria grouped into 6 sections and tables. These tables are informal but can easily be updated by the tool creators.

```
compare pm#WebKB-2 km#Ontolingua on
  (support of: a is#IR_task, output_language: a km#KR_notation, part: a is#user_interface), maxdepth 5
```

	WebKB-2	Ontolingua
support of:		
is#IR_task	+	+
is#lexical_search	+	+
is#regular_expression_based_search	+	.
km#knowledge_retrieval_task	+	.
km#specialization_structural_retrieval	+	.
(kind: {km#complete_inferencing, km#consistent_inferencing},		
input: (a km#query, expressivity: km#PCEF_logic),		
object: (several statement, expressivity: km#PCEF_logic))		
km#generalization_structural_retrieval	+	.
output_language:		
km#KR_notation	+	+
(expressivity: km#FOL)	+	+
km#FCG	+	.
km#KIF	.	+
km#XML-based notation	+	.
km#RDF	+	-
part:		
is#user_interface	+	+
is#HTML_based_interface	+	+
is#CGI-accessible_command_interface	+	.
is#OKBC_interface	.	.
is#API	+	.
is#graph_visualization_interface	-	-

```

km#CG_related_tool < km#language/structure_specific_tool,
> km#CG-based_KBMS km#CG_graphical_editor km#NL_parser_with_CG_output;

km#CG-based_KBMS < km#KBMS, > {km#CGWorld km#PROLOG\+CG km#CoGITaNT km#Notio km#WebKB};

km#WebKB > {km#WebKB-1 km#WebKB-2}, url: http://www.webkb.org;

km#input_language (*x,*y) = [*x, may be support of: (a km#parsing, input: (a statement, formalism: *y))];

[any pm#WebKB-2, part: (a is#user_interface, part: {a is#API, a is#HTML_based_interface,
a is#CGI-accessible_command_interface, no is#graph_visualization_interface}),
part: {a is#FastDB, a km#default_MSO_of_WebKB-2},
input_language: a km#FCG, output_language: {a km#FCG, a km#RDF},
support of: a is#regular_expression_based_search,
support of: a km#specialization_structural_retrieval
support of: a km#generalization_structural_retrieval,
support of: (a km#specialization_structural_retrieval,
kind: {km#complete_inferencing, km#consistent_inferencing},
input: (a km#query, expressivity: km#PCEF_logic),
object: (several km#statement, expressivity: km#PCEF_logic)
)]; // "PCEF": positive conjunctive existential formula

[any km#Ontolingua, part: {a is#HTML_based_interface, no is#graph_visualization_interface},
input_language: a km#KIF, output_language: a km#KIF,
part: {a km#ontolingua_library, no DBMS}, support of: a is#lexical_search];

```

#### 4 SUPPORTING COOPERATION BETWEEN KNOWLEDGE PROVIDERS

Here, we only consider asynchronous cooperation since it both underlies and is more scalable than exchanges of information between co-temporal users of a system. The most decentralized knowledge sharing strategy is the one the W3C envisages for the "[Semantic Web](#)": many very small KBs/ontologies, more or less independently developed and thus partially redundant, competing and very loosely interconnected. There now are many tools to align concepts from different ontologies; these tools are necessarily far from perfect although they can be sufficient for certain applications; Euzenat & al. (2005) give an evaluation. Thus, despite these tools, the above cited small ontologies have problems similar to those we listed for documents: (i) finding the relevant ontologies, choosing between them and combining them require commonsense (and hence is difficult and sub-optimal even for a knowledge engineer, let alone for a machine), (ii) a knowledge provider cannot simply add one concept or statement "at the right place" and is not guided by a large ontology into providing precise objects that complement existing objects and are more easily re-used, and (iii) the result is more or less lost to others and increases the amount of "data" to search.

A more knowledge-oriented strategy is to have a knowledge server permitting registered users to access and update a single large ontology on a domain and upload files that mix natural language sentences with knowledge representations. We know of only two knowledge servers having special protocols to support cooperation between users: [Co4](#) (Euzenat, 1996) and [WebKB-2](#). (Note: most servers support concurrency control and many servers support users' permissions on files/KBs; however, cooperation support is not so basic: it is about helping knowledge re-use, conflict prevention and the solving of each conflict once it has been detected by the system or a user). The approach of Co4 is based on peer reviewing; the result is a hierarchy of KBs, the uppermost ones containing the most consensual knowledge while the lowermost ones are the private KBs of the contributing users. We believe the approach of WebKB-2 which has a KB shared by all its users leads to more relations between categories (types or individuals) or statements from the different users and may be easier to handle (by the system and the users) for a large amount of knowledge and large number of users. Details can be found in Martin (2003a) but the next paragraph summarizes its principles.

Each category identifier is prefixed by a short identifier for the category creator (who is also represented by a category and thus may have associated statements). Each statement also has an associated creator and hence, if it is not a definition, may be considered as a belief. Any object (category or statement) may be re-used by any user within his/her statements. The removal of an object can only be done by its creator but a user may "correct" a belief by connecting it to another belief by a "corrective relation" (e.g., pm#corrective\_specialization). (Definitions cannot be corrected since they cannot be false; for example, a user such as "fg" is perfectly entitled to define fg#cat as a subtype of wn#chair; there is no inconsistency as long as the ways fg#cat is further defined or used respect the constraints associated to wn#chair). If entering a new belief introduces a redundancy or an inconsistency that is detected by the system, it is rejected. The user may then either correct his/her belief or re-enter it again but connected by a "corrective relation" to each belief it is redundant or inconsistent with: this allows and makes explicit the disagreement of one user with (her interpretation of) the belief of another user. This also technically removes the cause of the problem: a proposition A may be inconsistent with a proposition B but a belief that "A is a correction of B" is not technically inconsistent with a belief in B. (Definitions from *different* users cannot be inconsistent with each other, they simply define

different categories/meanings; a system of "category cloning" could be used to handle this situation automatically but the resulting ontology would be much more complex than via the manual handling of the situation by each category creator that is occasionally faced to it; hence, such a system has not been implemented in WebKB-2). Choices between beliefs may have to be made by people re-using the KB for an application, but then they can exploit the explicit relations between beliefs, for example by always selecting the most specialized ones. The query engine of WebKB-2 always returns a statement *with* its meta-statements, hence with the associated corrective relations. Finally, in order to avoid seeing the objects of certain creators during browsing or within query results, a user may set filters on these creators, based on their identifiers, types or descriptions.

For the construction of knowledge repositories, an interesting aspect of this approach to encourage re-use, precision and object connectivity is that it also works for semi-formal KBs. Here, regarding a statement, the adjective "semi-formal" allows the statements to use informal terms and even to be written in a natural language but entails that each statement must *at least* be related to another statement by a formal relation, for example a generalization relation (pm#corrective\_generalization, pm#summary, etc.) or an argumentation relation. Thus, to minimize redundancies and to help information retrieval within information repositories, this minimal semantic structure (which in many cases is the only one bearable by many persons) can be used to organize ideas that are otherwise repeated in many documents. For instance, for a Web site that centralizes and organizes/represents in a formal, semi-formal and informal way resources (tools, techniques, publications, mailing list, teams, etc.) related to a domain, it would be very interesting to have some space where discussions could be conducted in this minimal semi-formal way, and hence index or partly replace the mailing list: this would permit to avoid recurring discussions or presentations of arguments, show the tree of arguments and counter-arguments for an idea, permit incremental additions, encourage deeper or more systematic explorations of each idea, and record the various reached status-quo. Finally, it is fairly possible that structured discussions in different KBs or documents can be automatically aligned or merged in more successful ways than it is possible to align categories from different ontologies.

The above described knowledge sharing mechanism of WebKB-2 records and exploits annotations by individual users on statements but does not record and exploit any measure of the "usefulness" of each statement, a value representing its "global interest", acceptance, popularity, originality, etc. Yet, this seems interesting for a knowledge repository and especially for semi-formal discussions: statements that are obvious, un-argued, or for which each argument has been counter-argued, should be marked as such (e.g., via darker colours or smaller fonts) in order to make them less visible (or invisible, depending on the selected display options) and discourage the entering of such statements. More generally, the presentation of the combined efforts from the various contributors may then take into account the usefulness of each statement. Furthermore, given that the creator of each statement is recorded, (i) a value of usefulness may also be calculated for each creator (and displayed), and (ii) in return, this value may be taken into account to calculate the usefulness of the creator's contributions; these are two additional refinements to both detect and encourage argued and interesting contributions, and hence regulate them. Martin (2005) gives an algorithm that takes into account the argument tree and the votes for each statement, and the above cited two additional refinements, to calculate values of usefulness for statements and for creators. This algorithm provides reference values that should lead the information providers to be careful in their statements. However, ideally, the implementation of such an algorithm should also accept parameters from each user for allowing them to express their preferences: typically, a user may not want to see statements from people or kinds of people that have authored or backed-up ideas or arguments that were un-interesting or invalid from the viewpoint of this user (e.g., arguments referring to certain persons, books, traditions or deities). Such preferences might partially be automatically derived from the way the user votes (i.e., give values to) the statements of other users.

Despite the fact that the above described supports for cooperation rely on some centralizing mechanisms, it should be understandable that they could be supported by a peer-to-peer network (although they are easier to implement within a knowledge server). However, these supports do not solve the problem caused by the fact that one piece of information can be of interest in many domains and the fact that one knowledge server (or peer-to-peer network) clearly cannot support the knowledge sharing of all Web users; this problem is "which knowledge server should a person choose to query or update?". A server has to be specialized or to act as a broker for more specialized servers. However, if each server periodically checked *related servers* (more general servers, competing servers and slightly more specialized servers), imported the knowledge relevant to its domain and, for the rest, stored pointers to those servers, it would not matter much which server a Web user attempts to query first. For example, a Web user could try to query or update any general server and, if necessary, be redirected to use a more specialized server, and so on recursively (at each level, only one of the competing servers would have to be tried since they would mirror each other). If a Web user directly tried a specialized server, it could redirect him/her to use a more appropriate server or indicate which other servers may provide more information for his/her query (or directly forward this query to these other servers). Integrating knowledge from other servers is certainly not obvious but this is a more scalable and exploitable approach than letting people and machines select and re-use or integrate dozens or hundreds of (semi-)independently designed small ontologies. A more fundamental obstacle to the widespread use of this approach is that many industry-related servers are likely to make it difficult or illegal to mirror their KBs. However, other approaches would likely suffer from that too.

## 5 ONTOLOGIES

The users of a knowledge repository cannot be asked to update the shared ontology for declaring and defining *most of* the terms they use. A "lexical ontology" for English or another natural language should *at least* be provided to ease, check and guide knowledge entering and permit knowledge sharing and retrieval. A lexical ontology connects the words of a language to the categories representing the main meaning of these words, and connects these categories via conceptual links: unlike in foundational ontologies, no more complex definitions are generally provided; however, for important categories such as the top-level ones and the often used ones, "schemas" representing the relations that are commonly used from instances of these important categories, should be provided (for example as a resource for NLU or for the generation of combinable menus that guide and normalise knowledge representation). ([Martin, 2003b](#)) describes how such a lexical ontology was created by adapting and correcting WordNet 1.7 and also by extending it with several top-level ontologies and domain-related ontologies. However, many more corrections and additions and additions of links, schemas and ontologies deserve to be done. A knowledge repository also requires an initial domain ontology for inciting people to enter knowledge and guiding this entering. Our experiment in building a repository about CG related tools led us to partially model other related domains, and we soon had the problem of modularising the information into several files to support readability, searches, checking and systematic input. In order to be generic, we created six files: [Fields of study](#), [Systems of logic](#), [Information Sciences](#), [Knowledge Management](#), [Conceptual Graph](#) and [Formal Concept Analysis](#). The last three files specialize the others. Each of the last four files is divided into sections, the uppermost ones being "Domains and Theories", "Tasks and Methodologies", "Structures and Languages", "Tools", "Journals, Conferences, Publishers and Mailing Lists", "Articles, Books and other Documents" and "People: Researchers, Specialists, Teams/Projects, ...". This is a work in progress: the content and number of files will increase but the sections seem stable. Here are examples of the content for the first three sections.

```
// ===== Domains and Theories =====
wn#computer_science__computational_science //in FL, "__" is used for separating alternative names
(^branch of engineering science that studies computable processes and structures^)
subdomain: wn#artificial_intelligence is#software_engineering_science (is),
subdomain of: wn#engineering_science__engineering_applied_science__technology,
part: wn#information_theory, part of: wn#information_science;

// ===== Tasks and Methodologies =====
km#KM_task_knowledge_management_KM (^a K.M. (sub)task^) < is#information_sciences_task,
> km#knowledge_representation km#knowledge_extraction_and_modelling km#knowledge_comparison
km#knowledge_retrieval_task km#knowledge_creation km#classification km#KB_sharing_management
km#mapping/merging/federation_of_KBs km#knowledge_translation km#knowledge_validation
{km#monotonic_reasoning km#non_monotonic_reasoning}
{km#consistent_inferencing km#inconsistent_inferencing}
{km#complete_inferencing km#incomplete_inferencing}
{km#structure_only_based_inferencing km#rule_based_inferencing}
km#teaching_a_KM_related_subject km#language/structure_specific_task km#KM_methodology_task,
object of: km#knowledge_management_science, //between types, the default cardinality is 0..N
object: km#KM_structure; //the general relation "object" has different (more specialized) meanings
//depending on the connected categories: in the last relation, the meaning is
//"task object" (object worked on or generated by the task) not "domain object"

km#knowledge_retrieval_task < is#IR_task,
> {km#specialization_retrieval km#generalization_retrieval} km#structure_only_based_retrieval
km#analogy_retrieval {km#complete_knowledge_retrieval km#incomplete_knowledge_retrieval}
{km#consistent_knowledge_retrieval km#inconsistent_knowledge_retrieval};

// ===== Structures and Languages =====
km#KM_structure < is#symbolic_structure,
> {km#base_of_facts/beliefs km#ontology km#KB_category km#statement}
km#KB km#KA_model km#KR_language km#language_specific_structure;

km#KB_knowledge_base part: km#ontology km#base_of_facts/beliefs;

km#ontology__set_of_category_definitions/constraints
> km#lexical_ontology km#language_ontology km#domain_ontology km#top_level_ontology
km#concept_ontology km#relation_ontology km#multi_source_ontology,
part: 1..* km#KB_category 1..* km#category_definition;

km#KR_language__KRL__KR_model_or_notation
> {km#KR_model/structure km#KR_notation} //not km#semantics (not a structure)
km#predicate_logic_oriented_language km#frame_oriented_language km#graph_oriented_language
km#KR_language_with_query_commands km#KR_language_with_scripting_capabilities,
attribute: km#semantics;

km#CG_structure < km#language_specific_structure,
> km#CG_statement km#CG_language km#CG_ontology;
```

## 5 CONCLUSION

Despite the existence of the Web, publishing information still essentially implies writing documents and hence introducing related information and choosing an order and a level of detail to present this information. This linearization exercise is time-consuming and this can make information publishing/retrieval a frustrating experience since the vast majority of what has to be written/read is not new from the viewpoint of the writer/reader. Furthermore, all these redundancies make information hard to find, structure and compare, whether manually or automatically. Yet, almost all researches in information retrieval/sharing have avoided using knowledge-oriented approaches because (i) these approaches are more costly to implement, (ii) they require research on various fronts, and (iii) making semantic relationships explicit is not an exercise that people are (nowadays) accustomed to. In this article, we have described various elements of solutions for pushing those technical and psychological boundaries toward more formal, structured and normalised knowledge entering or retrieval. These elements relate to syntaxes, querying and comparison mechanisms, cooperation protocols or approaches, and ontologies. By continuing to experiment in the creation and support of knowledge repositories, we shall refine those elements and, hopefully, make such repositories popular. Of course, we do not believe that informal documents will or should disappear but we believe that structured discussions and, more generally, a normalised use of semantic relations, can become popular and enhance information retrieval. Our current main experiment relates to the representation and comparison of tools, techniques and ideas related to knowledge engineering. We shall also soon experiment with the use of such repositories for e-learning purposes.

## 6 REFERENCES

- [S. Buckingham, E. Motta and J. Domingue \(1999\)](#). *Scholarly Claims in Internet Digital Libraries: A Knowledge Modelling Approach*. Proceedings of ECDL 1999 (pp. 423-442), 3rd European Conf. Research and Advanced Technology for Digital Librarian, Paris, France, September 1999.
- M. Chein and M.L. Mugnier (1997). *Positive Nested Conceptual Graphs*. Proceedings of ICCS 1997 (Springer Verlag, LNAI 1257, pp. 95-109), Seattle, USA, August 4-8, 1997.
- [J. Euzenat \(1996\)](#). *Corporate memory through cooperative creation of knowledge bases and hyper-documents*. Proceedings of 10th KAW, (36)1-18, Banff, Canada, November 1996.
- [J. Euzenat, H. Stuckenschmidt and M. Yatskevich \(2005\)](#). *Introduction to the Ontology Alignment Evaluation 2005*. Proceedings of K-Cap 2005 (pp. 61-71), workshop on Integrating ontology, Banff, Canada, 2005.
- [W.D. Hillis \(2004\)](#). *"Aristotle" (The Knowledge Web)*. Edge Foundation, Inc., No 138, May 6, 2004.
- [Ph. Martin and P. Eklund \(2000\)](#). *Knowledge Indexation and Retrieval and the Word Wide Web*. [IEEE Intelligent Systems](#), special issue "Knowledge Management and Knowledge Distribution over the Internet", pp. 18-25, May/June 2000.
- [Martin Ph. \(2002\)](#). *Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English*. Proceedings of [ICCS 2002](#), 10th Int. Conf. on Conceptual Structures (Springer Verlag, LNAI 2393, pp. 77-91), Bulgaria, July 15-19, 2002.
- [Ph. Martin \(2003a\)](#). *Knowledge Representation, Sharing and Retrieval on the Web*. Chapter of a [book titled "Web Intelligence"](#), (Eds.: N. Zhong, J. Liu, Y. Yao; Springer-Verlag, pp. 263-297), January 2003.
- [Ph. Martin \(2003b\)](#). *Correction and Extension of WordNet 1.7*. Proceedings of [ICCS 2003](#) (Springer Verlag, LNAI 2746, pp. 160-173), Dresden, Germany, July 2003.
- [Ph. Martin, M. Blumenstein and P. Deer \(2005\)](#). *Toward cooperatively-built knowledge repositories*. Proceedings of [ICCS 2005](#), 13th Int. Conference on Conceptual Structures, (Springer Verlag, LNAI 3596, pp. 411-424), Kassel, Germany, July 18-22, 2005.
- W. Schuler and J.B. Smith (1990). *Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts*. Proceedings of ECHT'90 (Cambridge University Press, pp. 137-151), INRIA, France, Nov. 1990.
- [D. Skuce and T.C. Lethbridge \(1995\)](#). *CODE4: A Unified System for Managing Conceptual Knowledge*. International Journal of Human-Computer Studies, 42, pp. 413-451. [Fact Guru](#) is the commercial version of CODE4.
- J. Nanard, M. Nanard, A. Massotte, A. Djemaa, A. Joubert, H. Betaille and J. Chauché (1993). *Integrating Knowledge-based Hypertext and Database for Task-oriented Access to Documents*. Proceedings of DEXA 1993 (Springer Verlag, LNCS Vol. 720, pp. 721-732), Prague, 1993.
- Sowa J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA, USA, 1984.
- [Sowa J.F. \(2006\)](#). *Concept Mapping*. Web document. <http://www.jfsowa.com/talks/cmapping.pdf>