Report on past research works

Helping knowledge representation, sharing and retrieval

Dr Philippe MARTIN

My research so far has been focused on integrating and developing methodologies, techniques, tools and resources to support a precise manual knowledge representation, sharing and retrieval (KRSS). The resulting models, notations and normalization techniques for representing, indexing, querying, evaluating and integrating knowledge are however also important to follow for precision-oriented (semi-)automatic knowledge extraction and integration. From the end of 1992 to the end of 1999, the context of my work was the "semi-independent knowledge representation approach" where knowledge providers do not have to represent their information into a shared knowledge base (KB). This approach is still the only one considered by most knowledge representation, sharing and retrieval (KRSS) related works today. Nonetheless, as detailed below, this approach is very restricting for precise KRSS. Hence, from 2000 to 2007, my work focused more on also enabling people to easily update a very large consistent well-organized KB while avoiding the classic problem of either allowing any user to modify any part of the KB (as in wikis) or having the bottleneck and restrictions associated to the existence of a selection committee.

Table of Contents

1. Research from 1992 to 1999

1.1. Extending a knowledge acquisition methodogy to acquire explanatory information (Master thesis; 1992)

Tackled problem. In 1992 KADS was the most well known and used Knowledge Acquisition (KA) methodology for building knowledge bases (KBs) or KB systems (KBSs) - in this research report, KA covers the phases of knowledge extraction and modelling to those of the implementation and testing of a KBS. However, neither KADS nor any other KA methodology was precise enough to guide a knowledge engineer into acquiring information from sources of expertise (documents, experts) for the KB to be self-explanatory or for the KBS to be able to generate good explanations on its knowledge and reasonings. Thus, the KB or KBS was hard to understand and trust.

Found solution. A "model of cooperation expertise" and a "model of communication expertise" were added to the KADS Conceptual Model for enabling the modelling of explanation related knowledge. The content of these two new models of expertise and the relationships that should occur between both models were specified. A list of questions to acquire problem solving knowledge and explanatory knowledge related to

each type of element of an interpretation model was also provided. To come up with this result I synthesized the various knowledge acquisition and explanatory techniques used so far. The difficulty relied in making that synthesis and instantiating it into the KADS framework. This research can however be used in other KA methodologies and has been published in (Martin, 1993, 1993a, 1994).

1.2. Integrating and easing knowledge management and document management with CGKAT, a knowledge acquisition tool integrating a structured document editor, an inference engine and ontologies (PhD thesis; 1993-1996)

Tackled problems. The usability of a KBS or KA tool is greatly enhanced if it enables its users

- to manage, organize, query, and generate documents as well as knowledge representations (KRs), in both structural and semantic ways,
- to cross-link KRs and document elements (DEs) for documenting the KRs or indexing the DEs, and to mix KRs and DEs within documents,
- to see and use intuitive and expressive KR notations, graphical as well as textual, as well as script languages to combine query commands on DEs or KRs, and
- to re-use not only task models but also a general ontology (semantically organized dictionary of conceptual categories such as concept types/instances and relation types, with associated definitions or axioms), thus easing and guiding knowledge modelling as well as indexing it (without such an ontology, the user has to re-create parts of it and, for knowledge modelling, sharing, re-usability and retrieval purposes, the result is not as good as if the user extended the ontology).

In 1993, KA tools offered only very partial solutions to achieve these features. The Semantic Web related tools now offer many possibilities with respect to these problems although they are still not ideal.

Found solutions. I designed CGKAT, a KA tool that

- integrated the structured document editor Grif/Thot (Quint & Vatton, 1992) with the Conceptual Graph (CG) workbench CoGITo (Haemmerlé & Guinaldo, 1999), via 10,000 lines of C++ and the definition of models for a CG (a data model, a presentation model and an event model) in the languages of Thot, thus enabling (i) a CG to be added within Thot documents, graphically edited, and hyperlinked, indexed or more structurally exploited via Thot, and (ii) a CG (and/or the DE indexed by it) to be retrieved via semantic queries,
- proposed commands to retrieve or combine CGs or DEs, and a script language to combine these commands which was usable from outside CGKAT as well as from within it (thus allowing to generate "virtual documents" for example),
- integrated WordNet 1.5 with some top-level ontologies and represented some models of the KADS library in CGs, thus allowing their search and combination via semantic queries, and
- permitted the dynamic update of the large default ontology (via a system of memory caches).

These solutions were original and solved the four above cited problems. I showed how the exploitation of ontologies, and especially large ontologies, by knowledge engineers improves the consistency, extendibility and re-usability of their KRs. This idea is now well accepted. The re-use of WordNet for KA and the availability of queries and script languages to combine structural or semantic queries are now not uncommon, although this can mainly only be done via graphical interfaces, not also via intuitive textual languages. The integration of a structured document editor to an inference engine has not been replicated. Indeed, Thot had data/presentation/event model specification languages and graph editing capabilities that even Amaya (the W3C browser based on Thot and created by the W3C and WAM project at INRIA) still does not have now that it re-uses XML, CSS, XSLT and SVG. When it does, my work could be re-used to create an API for Amaya to allow the graphical display, editing and embedding XML/HTML documents of XML-based KRs. Given the number and popularity of XML-based KR notations, with a Web-accessible API (for example like the one provided for Google Maps) would likely be very popular.

Conclusion. During and after my PhD in the Acacia team of the INRIA, CGKAT and my extension of CoGITo to use it for KA purposes were re-used by other PhD students of Acacia. I tested CGKAT via the representation and management of KADS models as well as various interviews from experts on accidentology. My PhD thesis covered three domains: KA, KRs (especially Conceptual Graphs) and

structured/hypermedia document management tools (CGKAT can also be viewed as a knowledge-oriented instance of such tools or, more generally, as a precision-oriented desktop Information Retrieval tool). Its main difficulty relied in the choice, extension and integration of many unrelated techniques to solve the four above cited problems. This research has been published in (Martin, 1995, 1995a, 1995b, 1996, 1997a) and Martin & Alpay (1996).

1.3. Enabling the use of readable, expressive or concise knowledge representation/indexation/querying languages within Web documents with WebKB-1, a personal KB server adapting CGKAT to the Web (1997-1999)

Tackled problems. The advantages for a KA or Information Retrieval (IR) tool to exploit the possibilities of the Web and be accessible from the Web became clear by the end of 1996. However, at that time, none of the Web-based tools for KA, IR or Computer Supported Cooperative Work (CSCW) were *knowledge-based* (i.e., precision-oriented and flexible) and able to exploit expressive and readable KRs embedded within documents. Nowadays, many Semantic Web tools are precision-oriented KA, IR and/or CSCW tools but, as in 1997, the exploited models and notations use lead to problems:

- most of these tools - e.g., RDF-based tools - exploit knowledge representation and querying languages which have an XML syntax (and hence are difficult to use without a graphical interface), do not include a script language to enable queries and assertions to be combined and are difficult to mix with other DEs within documents, or
- like Ontobroker (Fensel, 1998) or, nowadays, like tools exploiting micro-formats or the RDFa notation, these tools use simple but quite restricting notations embedded within HTML or XML tags; although nowadays well considered, these notations cannot be displayed or indexed and hence do not genuinely reach one of their goals, which is to avoid the redundancies between formal and informal information (Fensel, 1998).

In both cases, these notations follow models which, from a *general* knowledge representation and sharing viewpoint, are low-level and poorly expressive, at best the RDF+OWL model (this point is detailed below), and hence are hard to use except for simple KRs or queries, and lead to write to ad-hoc or biased KRs that are difficult to compare automatically and hence to retrieve (Marshall & Shipman, 2003). It should be noted that the issues of computability and decidability are *not* related to notations or models but to applications and inference engines: each application or engine developer can decide which parts in expressive KRs to ignore in order to deal with the problems of efficiency, consistency and completeness (e.g., some IR applications may chose to ignore meta-statements). Thus, restricting notations without knowing exactly for which application it will be used is not an advantage, it can only limit and bias knowledge modelling and the many possible inferencing techniques (Sowa, 2000). Even if OWL-Full is expressive enough to be undecidable, it is insufficient for general knowledge representation and sharing (e.g., the representation of natural language sentences) since for example it does not handle full negation, sets nor meta-statements, and it is bound to be extended (Patel-Schneider, 2005). OWL has a low-level model since it does not include high-level constructs (such as numerical quantifiers) which are needed for general knowledge representation and which even knowledge engineer would often struggle to define.

Core solutions. Given the above problems and since no Web browser had editing and presentation capabilities similar to those of Thot, based upon my work on CoGITo I developed WebKB-1, a Web-accessible KB server that could be sent concise and powerful commands (Web document fetching and searching commands, KRs or queries, scripts) via the HTTP protocol (i.e., via both GET and POST parameters) and parse or execute commands included in fetched Web documents (these commands or KRs are isolated from the informal document elements by special marks). The possibility to use a powerful language of commands within GET parameters permits calls to WebKB-1 to be included within a Web document and combined together or with calls to other applications. Thus, for example, virtual documents can be easily created and queries can be associated to hyperlinks. Nowadays, many Semantic Web tools propose special "semantic tags" to include the result of certain predefined semantic queries within a document, although such tags are often aimed to be used by Java programmers, not end-users. Some other Semantic Web tools accept XML-based parameters via the SOAP protocol which is only usable by programmers. Thus, from an end-user viewpoint, both solutions are less flexible and user-friendly than the solution adopted in WebKB-1 which follows a REST Web service architectural approach (Fielding, 2000);

for example, in both solutions, commands cannot be easily easily associated to hyperlinks nor combined with pipes (as in the Unix shell or WebKB-1). Furthermore, the language of commands of WebKB-1 is more powerful than those of current Semantic Web tools. First, unlike the indexation languages developed by the W3C, the one accepted by WebKB-1 can index *any* part of a Web document (to that end, the *full textual content or representation of a document element* and its occurrence number in the document can be indicated in an indexation). Second, there is a unique high-level CG-based language for asserting, indexing and querying information. Third, I invented various search operators/commands for the query language of WebKB-1 (the most basic one being an operator to search for both the specializations and the generalizations of a CG) and extended the CG notation to allow a concise specification of a search for paths within one or several CGs (Martin & Eklund, 2001); nowadays, for similar purposes, regular expressions are also added to SPARQL, the most commonly used RDF-related knowledge querying language. Fourth, I allowed the use of semi-formal elements, for example the use of ambiguous words instead of unambiguous categories identifiers. Fifth, the language of commands used in WebKB-1 is also procedural one, not just a declarative one, and is an extension of the one used in CGKAT to support more reasoning; for example, it was used to solve the KA test problem named Sisyphus-I (Martin & Eklund, 1999a). Sixth, as in CGKAT, a large ontology based on WordNet can be exploited.

Conclusion regarding the core solutions. WebKB-1 is a "personal" KB server - as opposed to a shared KB server - in the sense that the KBs are Web accessible documents created by users and that users ask the server to access and execute for answering queries. Many Semantic Web tools only allow the exploitation of documents stored on the server machine. WebKB-1 integrates KB management with document management as much as current Web browsers permit, was one of the very first Semantic Web tools and, with respect to many features, still advantageously competes with current Semantic Web tools. However, WebKB-1 should not be seen as a competitor to these tools. First, it can be seen as a complement to them since it can be called from them, for example as a way to offer complementary ways to query, enter, index and display knowledge. Second, it can be seen as a show-case tool for other architectural and presentation avenues, since WebKB-1 remains a research prototype (like most Semantic Web tools); most of its features are now included in WebKB-2 (Martin & Eklund, 2001) (Martin, 2003a) although not all of them yet. I applied WebKB-1 to the representation, indexation and querying of a base of images from the Club Med, various ontologies, and interviews from experts on accidentology.
Recognition. WebKB-1 was finalist of the 1999 Asia-Pacific Oracle Queensland IT&T Awards for Excellence in the Intelligent Technologies category. My research related to WebKB-1 - and not already cited above - was published in (Martin, 1997, 2003), (Martin & Eklund, 1999, 1999a, 1999b, 1999c, 2000a) and (Eklund et al., 1998, 1999).

Refinement of the CG linear form with a family of languages. An important work related to WebKB-1 is the design and implementation of three notations - FL (For-Links), Frame-CG (FCG) and Formalized English (FE) - which I derived from the CG Linear Form (CGLF) in order to improve on the qualities that made its success: its intuitiveness, conciseness, and "knowledge normalization" effect. I began designing and implementing these notations in 1997 but I often refined this work (Martin & Eklund, 1999, 1999b) (Martin, 2000, 2002, 2006c, 2007); hence, I present this work in this separate paragraph and I do not present it again in the next section ("Research from 2000 to 2007 ..."). These three notations, which can be used for both asserting and querying, are complementary:

- FL is very concise but does not support quantifiers: it can only represent description-logics like statements, e.g., OWL-full statements, but without restrictions on their associated context (meta-statements). FL permits to represent a large volume of knowledge in a structured way and a small amount of space, which is important to reduce navigation, visually compare objects (categories or statements) and their relationships, and thus more easily find and understand them.
- FCG is a concise notation with the expressiveness of KIF (i.e., first-order logic, plus meta-statements and sets). In 1998, Tim Berners-Lee (the director of the W3C) began to design N3 as "a readable language for data on the Web". N3 looks a bit like FCG but is not as concise, regular, readable and expressive.
- FE has the same structure as FCG but look likes English, although currently still a very odd English. FE is the most expressive and formal of the currently used controlled languages (Pool, 2006).

A simple example illustrating these languages is given below in Table 1.

My work on these notations went together with my gathering and refinement of good "lexical, structural and ontological principles" (Martin & Eklund, 2000) (Martin, 2000, 2004b). Indeed, for normalization, precision and re-use purposes, these languages and their semantic parsers are designed to encourage people to follow these principles. Table 2 gives some examples of these principles, some that are encouraged by FCG, FE and FL, and others that the users is simply advised to follow.
In (Martin, 2002) and (Martin, 2007a), I compared these languages with commonly used notations (KIF, N3, RDF+OWL, UML, CGLF, CGIF). However, the import and export functions of WebKB-1 and WebKB-2 from or to these other notations currently only cover a subset of these languages, generally the OWL-DL subset.
Recognition. Related to this work on notations is my participation to the CGIF&KIF sub-committees of the ISO/IEC JTC1 SC32 from 2000 to 2003, my participation to the specification of the Ontology Definition MetaModel of the Object Management Group (Raymond et al., 2003), and my participation to the definition of a notation for UML Activity Diagrams (Flater et al., 2007).

Table 1. Very simple representations in English, FE, FCG, CGLF, FL and KIF.

KIF is a low-level but expressive and common knowledge representation language. Since the translation in RDF+OWL/XML is 25 lines long without even representing who created those statements, this translation is not included below. This example shows how FL better permits to view and compare the various relationships between the objects than the other languages. Providing easy-to-read-and-use constructs for all the features needed for general knowledge representation (e.g., the representation of natural language sentences) is the main advantage of FE and FCG for precision, normalization and re-usability purposes. Example of such features are contexts (meta-statements), higher-order statements, numerical quantifiers (such as "at least 3" or "2%"), as well as the various kinds of definitions, sets and set interpretations. Low-level languages such as KIF and RDF+OWL do not offer this possibility. In the following representations, for readability purposes, the source or creator of each category is not indicated: no category identifier includes the identifier of its creator or source, as a prefix or suffix.

```
En:  According to John, any human_body is a body and has at most 1 head and 2 arms.
     According to Jack, any human_body has exactly 1 head and conversely.
     According to Jo, male_body and female_body are exclusive subtypes of human_body.

FE:  `Any human_body is a body and has for part {at most 1 head, at most 2 arms}'(John).
     `Any human_body has for part 1 head'(Jack).  'Any head is part of 1 head'(Jack).
     `Human_body has for subtype excl{male_body, female_body}.

FCG: [any human_body, instance of: a body, part: {at most 1 head, at most 2 arms}](John);
     [any human_body, part: 1 head](Jack).  [any head, part of: 1 head](Jack);
     [human_body, subtype: excl{male_body, female_body}];

CGLF:[situation: [type: human_body]->(supertype)->[type: body] ]->(believer)->[Person: John]
     [situation: [human_body: @forall]->(part)->[head: {*}@<=1] ]->(believer)->[Person: John]
     [situation: [human_body: @forall]->(part)->[arm : {*}@<=2] ]->(believer)->[Person: John]
     [situation: [human_body: @forall]->(part)->[head: {*}@1] ]->(believer)->[Person: Jack]
     [situation: [head: @forall]->(part)->[human_body: {*}@1] ]->(believer)->[Person: Jack]
     [situation: [human_body]->(subtype)->[type: {male_body, female_body}]
                 [male_body]->(exclusion)->[female_body] ]->(believer)->[Person: Jo]

FL:  human_body supertype: body (John),
                part     : head [any->0..1(John), any->1(Jack), 1<-any(Jack)]
                           arm  [any->0..2(John)],
                subtype  : excl{male_body female_body}(Jo);

KIF: (believer '(forall ((?b human_body))  (body ?b)) John)
     (believer '(forall ((?b human_body))  (atMostN  1 '?a head (part ?b '?a))) John)
     (believer '(forall ((?b human_body))  (atMostN  2 '?a arm  (part ?b '?a))) John)
     (believer '(forall ((?b human_body))  (exactlyN 1 '?a head (part ?b '?a))) Jack)
     (believer '(forall ((?a head))        (atMostN  1 '?b human_body (part '?b ?a)))  John)
     (believer '(forall ((?a head))        (exactlyN 1 '?b human_body (part '?b ?a)))  Jack)
     (believer '(forall ((?b male_body))   (and (human_body) (not (female_body ?b))))  Jo)
     (believer '(forall ((?b female_body)) (and (human_body) (not (male_body ?b))))    Jo)
     ;; with:
     (defrelation atMostN (?num ?var ?type ?predicate) :=
     (exists ((?s set)(?n)) (and (size ?s ?n) (=< ?n ?num)
     (truth ^(forall (,?var) (=> (member ,?var ,?s)
                                 (and (,?type ,?var) ,?predicate)))))))
```

Table 2. Examples of normalization guidelines (best knowledge modelling practices).
Rationales are given in (Martin, 2002, 2004b).

Examples of best knowledge modelling practices encouraged by FCG, FE and FL.

- use singular nouns (nor a verb, adverb, etc.) for (at least one of) the names of a concept type or relation type (hence, for example, a concept type name such as "Define" and relation type names such as "define", "defined_by" or "has_definition" are inadequate; in both cases, "definition" is better)
- use basic binary relations and avoid introducing new relations (in particular, relations representing processes)
- when defining a category, use all adequate transitive relations (e.g., specialization relations and part relations)
- do not define a category using "instance" relations when "subtype" relations can be used (this also means not defining a category as a individual when it can be defined as a type)
- give the most precise supertypes or types of a new category
- contextualize beliefs in time and space

Examples of best practices not encouraged by FCG, FE and FL.

- explicitly represent action categories with their subtask+specialization relations and re-use them as much as possible, even in informal sentences; for example,
  - do not write "governments should enforce animal rights"
    but "the enforcement_of_animal_rights_by_governments should happen"
  - do not write "the more precise the representations, the more precise the information retrieval"
    but ` "the increase of the precision of knowledge_representation" has for consequence
    "the increase of the precision of information_retrieval" '
    or ``an increase with object the precision characteristic of a knowledge_representation
    with object some information *i ' has for consequence `an increase with object the
    precision characteristic of the information_retrieval with object *i ' '
- store, organize and document your knowledge inside Web-accessible files then ask the shared KB server to load and check these input files
- provide sections such as "Process", "Structures", "Tool", "Agents", ... in your input files and organize the presentation of the knowledge in partOf/subtypeOf hierarchy

Table 3. Interconnection of semi-formal normalized statements in FL.

1) In this example, only the creators of the relations have been made explicit, not the creators of the statements. The terms used below for the relations and the terms including an underscore are informal but the relevant related formal terms/categories for these informal terms can be automatically found.
2) The statement beginning by a back quote is in FE; it connects two informal statements.
3) The terms used below for the relations and with an underscore inside are informal but the relevant related formal terms/categories for them can be automatically found.
4) To normalize the formulation of the statements and ease their organization and retrieval, most of the statements begin by a process and all the processes have related formal terms/categories.
5) The parenthesis are used for two different purposes which the indentation help understand: (i) allowing the direct representation of relations from the destination of a relation, and (ii) representing meta-information on a relation, such as its creator (e.g., "pm" or "fg") or a relation on this relation.
6) Dashes are used for joint arguments/objections (e.g., a rule and its premise).
7) Most notations proposed by argumentation systems do not have this expressiveness and compactness, and hence restrict or bias the work of their users.

```
"knowledge_sharing_with_an_XML-based_language is advantageous"
   generalization: "knowledge_sharing_with_an_XML-based_language
                                is possible" (pm),
   specialization: ("knowledge_sharing_on_the_Web_with_an_XML-based_language
                     is advantageous"
                       argument of: "the Semantic Web should have an XML notation" (pm)
                   )(pm),
   argument: - "XML is a standard" (pm)
             - ("knowledge_management_with_classic_XML_tools is possible"
                   corrective_restriction:
                     "syntactic_knowledge_management_with_classic_XML_tools is possible" (pm,
                       argument: ("there is no exploitation_of_semantics by classic_XML_tools"
                                     example: "there is no taking_into_account by classic_XML_tools
                                               of the fact that RDF/XML has multiple equivalent
                                               serialisations" (pm)
                                  )(pm)                                                    )
               )(pm),
   argument: "the use of URIs and Unicode is possible in XML"
             (fg, objection: "the use of URIs and Unicode can easily be made possible in
                              most syntaxes" (tbl, pm)  //According to pm, the last statement
                                     // is an objection by Tim Berners Lee on F.G.'s argument
                                     // (the use of the relation, not its destination)
             ),
   objection: - ("the use_of_XML_by_KBSs implies several tasks to manage"
                    argument: "the internal_model_of_KBSs is rarely XML" (pm)
                 )(pm),
              - ` "an increase of the number of tasks *t to_manage" has for consequence
                  "an increase of the difficulty to develop a software to manage *t" '(pm),
   objection: - "knowledge_sharing_with_an_XML-based_language will force
                 many persons (developers, specialists, etc.) to understand
                 complex_XML-based_knowledge_representations" (pm)
              - ("understanding complex_XML-based_knowledge_representations is difficult"
                    argument: "XML is verbose" (pm)
                 )(pm);
```

2. Research from 2000 to 2007: enabling the cooperative building of a large knowledge base with WebKB-2, a large-scale shared knowledge base server

General problem. Storing formal, semi-formal or informal information in a personal document or KB, rather than in a large shared KB, is choosing a "semi-independent knowledge representation approach". This approach is restricting, difficult and sub-optimal from a knowledge entering and sharing viewpoint because:

- instead of simply finding and specializing conceptual categories or statements within a large well-organized KB, each knowledge provider must (i) create a whole new document, ontology or KB, (ii) try to find one or several ontologies (or KBs) to re-use and specialize, and (iii) try to understand and merge (parts of) the selected ontologies and hence solve their redundancies and inconsistencies;
- the result of this work is yet another document, ontology or KB, and hence increases the amount partially redundant, inconsistent and unrelated "data" for other users to search and annotate or extend (and these annotations or extensions will therefore themselves likely be independent, hence compounding the problem);
- the smaller the re-used ontologies or KBs, the more heterogeneous in quality and design they are likely to be, the more likely their design purposes, assumptions, authors, and temporal context will not be made in an explicit way (that is, using KRs), the less interconnected they are likely to be, the more the user will have to add, the less the user will be guided by the ontology either directly (that is, by its structure and the many categories which document or define each other) or indirectly (that is, by a KA tool exploiting the definitions and constraints to guide or validate knowledge entering), and hence the less connected to other ontologies and the less re-usable the new ontology or KB is likely to be;
- documents, ontologies or KBs that have not been pre-indexed or loaded by a KA or IR tool may be long to load and hence exploit dynamically;
- the approach cannot escape the various problems related to the evolution of documents, ontologies or KBs and hence to their multiple versions (within a KB, this can be avoided);
- the merging of ontologies and the extraction of deep KRs from general documents are difficult task to automatize (these are indeed also difficult tasks even for a knowledge engineer) and hence, nowadays, have understandably poor results (Euzenat et al., 2005) (Wang et al., 2003) (Dang et al., 2006);
- automatic or manual information retrieval is restricted by the redundancies and rare connections that have been manually set (or that can be automatically extracted) between concepts or statements within or across the various documents, ontologies or KBs, as well as the absence of explicit information on the context (author, assumptions, temporal constraints) of the KB or on each piece of knowledge; indeed, under those conditions, information retrieval results will have to be at least partly based on lexical matching and/or will be a list of documents, ontologies or KBs instead of being a list of KRs precisely answering a question and connected (i) to each other by semantic relations, and (ii) to other categories or statements (e.g., arguments, objections, specializations, subtasks, etc.), thus supporting knowledge comparison and further information retrieval by browsing or queries.

Thus, the "semi-independent approach" is unscalable for precision-oriented large-scale IR or KA. This is why the number of projects based on wikis, semantic wikis, and large KBs is rapidly growing. However, shared documents, ontologies or KBs commonly suffer from the following problems which make their shared approaches unscalable for large-scale IR or KA:

- there is a committee of special users that decide what should or should not be in the KB (and sometimes they are the only ones allowed to make additions), or any user is allowed to remove other users' knowledge, or the knowledge of each user is stored in semi-independent modules (this is the "semi-independent approach" within a document, ontology or KB, , for a particular community);
- the structure or ontology is fixed and cannot be dynamically extended by the users;
- the supporting system does not guide people into adding knowledge in normalized way or "at the right place" (and hence avoid undetected partial redundancies or inconsistencies) because the system has no mechanisms or normalization conventions to keep it well organized and/or its KB has not been initialized with a sufficiently organized content (with respect to what the KB is aimed to cover) for people to know where the "right place" is - this point is understandably not fully addressed by the

guidelines, protocols and ontologies of WebKB-2 and hence is one of the theme of my research project;
- the supporting system does not have mechanisms (knowledge presentation, evaluation and filtering mechanisms) to let each user see only what she wants to see in the possibly overwhelming amount of information.

Introduction about the core solutions of WebKB-2. The following sub-sections introduce the core ideas of my solutions for addressing these last problems (not yet implemented ideas or implementation-only related ideas are not presented; although implemented, interface-related works - such as the presentation of knowledge according to a wide range of presentation options, or the generation of combinable "cascading knowledge entering/querying forms" based on definitions or schemas associated by users to categories - are not presented either). These solutions still need to be refined and these are minimal elements for scalable IR or KA, which is why my research project proposes to extend these solutions and to develop ways to integrate the "semi-independent approach" with a "scalable KB sharing approach" since the co-existence of these two approaches is unavoidable. WebKB-2 includes many of the features of WebKB-1 and its mechanisms work on formal as well as informal information. Although interesting to use in its own right, WebKB-2 can also be seen as a show case tool for techniques that bigger projects could integrate to reduce some of their problems. Like WebKB-1, it is usable via a Web-accessible language of commands and hence can be combined with other applications. WebKB-2 has been tested on a wide range of domains (teaching materials, philosophical materials, all kind of ontologies etc.) and applications (vocabulary specification, tourism, e-learning, etc.). Other examples of potential applications are: collaboratively-built states of the art, precision-oriented corporate memories and catalogs, e-government and e-science. WebKB-2 has been tested or used by many of my students, several researchers and, anonymously, by a greater number of Web users. Finally, I co-supervised two PhD students for whom I defined PhD subjects related to my own work, and one undergraduate student from the University of Technologie of Belfort-Montbéliard did his 6-month-long full-time training course on the extension of the import-export features of WebKB-2.
Recognition. WebKB-2 won the 2001 Asia-Pacific Oracle Queensland IT&T Awards for Excellence in the Research and Development category. The "multi-source ontology" of WebKB-2 was voted a "candidate material for a standard" by the voting members of the IEEE SUO (SUO, 2004). Some employees of IBM Washington use the short and intuitive category identifiers of this ontology in their technical documentations to specify the meaning of some words.

2.1. Enabling and encouraging the integration of knowledge without semantic conflicts, redundancies nor agreement on terminology and beliefs

Tackled problem. Supporting the collaborative building of a shared KB requires much more than (i) providing tools guiding the merge of ontologies, and (ii) supporting persistency, concurrency control, synchronous cooperation (e.g., via meeting spaces) and user-dependent read/write permissions on modules, categories or statements. It requires protocols to keep the KB well organized, without detected redundancies or contradictions, without forcing the users to agree on terminology or beliefs, nor even discuss with each other (this is essential for scalability purposes). Only two KBSs seem to have special protocols to support cooperation between people: Co4 (Euzenat, 1996) and WebKB-2 (Martin & Eklund, 2001) (Martin, 2003a). Co4 is not used anymore. Its approach was based on peer reviewing, the result of which was a hierarchy of KBs, the uppermost ones containing the most consensual knowledge and the lowermost ones being the private KBs of the contributing users. This approach was intuitive but based on organized discussions and led to the creation of separate partially redundant KBs only for storing how consensual each piece of information was, and thus could not guide nor exploit a tight semantic organization between all the objects of all these KBs.

Found solution. For inferencing or presentation purposes, all (meta-)information related to an object (category, relation or statement) - for example its creator, source document, creation date, and the statements using it - should readily accessible. From a programmer's viewpoint, this is often easier if all the information is stored in the same KB. Furthermore, when different KBs are used, for example one per knowledge creator, the context of the knowledge in each KB (e.g., its creator, certain temporal or spatial constraints, the creation date of each object) are not always made explicit. This context should be made explicit along each

information whenever the object is exported. For example, to avoid lexical conflicts, each category identifier should include an identifier for the creator of the category. My core solution to keep a KB "minimally organized" and, technically, without detected semantic conflicts, is to support and encourage the manual or automatic setting of relations of specialization or correction relations between inconsistent or partially redundant statements (this solution is a simple one, it does not require the use of non-monotonic logics). The user can then use filtering mechanisms on the KRs (including their relationships and meta-information) to see only what she wants and, if needed, generate an adequate module, KB or even a "lattice of ontologies" (an often referred-to architecture for knowledge sharing purposes but which suffer from the same above mentioned problems as all other module-based solutions). For example, for a particular application in a certain domain, a user may wish to select only statements from people who have a degree in this domain and, to choose amongst competing statements, select only the most specialized or that correct the other ones.
Table 4 summarizes the core ideas behind the editing protocols used in WebKB-2 for keeping the KB technically consistent and without redundancies while allowing the users to explicit their own beliefs. The statements do not have to be formal and can be arbitrarily large or small (in order to allow for incremental refinements) but have to be connected by relations: at least relations of specialization or correction, and as many relations as the knowledge providers are willing to enter. For scalable knowledge sharing and retrieval purposes, knowledge (within a KB and, ideally, across KBs too) should actually be more than "minimally organized", for example, each category or statement should have a unique place in each hierarchy composed of transitive relations[1], not just specialization relations. This is not yet enforced by WebKB-2 but is strongly advised by its normalization guidelines. In the domain of formal specifications for softwares, Dromey (2006) also came to this conclusion, although he only referred to the few hierarchies he thought necessary to create for scalability purposes. The partOf hierarchies (e.g., those using subtaskOf or physicalPartOf relations) are the second most important and ubiquitous kinds of hierarchies.
The approach and editing protocols illustrated in Table 4 could be used in any structured cooperatively updated repository (e.g., structured wikis, semantic wikis, structured corporate memories) to solve the problems related to the existence of a committee or to right of removal by any user. The problems related to multiple versions is also avoided: since all the objects (statements, relations, and categories - hence the terminology) are individually contextualised, the removal of past beliefs or past categories is of no benefit, it is only a loss of information.

Conclusion. This research has been published in (Martin et al., 2001) and summarized (as well as partially refined) in (Martin et al., 2005, 2006, 2007) and (Martin & Eboueya, 2008). WebKB-2 currently only detects inconsistencies or partial redundancies via graph matching and the exploitation of the ontology. WebKB-2 does not detect them between completely informal statements. In the future, it could use heuristics to perform such a detection and hence suggest (instead of enforce) the use of corrective relations. In one annexe of my PhD thesis (Martin, 1996), to deal with various interpretations of a category by multiple users, I describe an approach based on an automatic system of "category cloning" that complements the approach introduced in Table 4. This system has not been implemented in WebKB-2 because of the complexity and disorganization it would lead to in the long term, compared to the manual setting of relation of correction by users. However, since forcing such a manual setting may annoy some users, also integrating this system can be of value.

[1]For example, if tasks have been recorded in a KB and, as they should, have been organized in a subtask hierarchy, a user that wants to declare a new task should represent all the subtask relationships (that she is aware of) between this new task and the already represented ones. If each user represents all such subtask relationships, the place of each task in the subtask hierarchy is unique and there is no independent extensions of this subtask hierarchy, hence, no redundancies within it.

Table 4. Core ideas behind the editing protocols.

- Any user can add any object, but an object can only be removed by its creator.
- By convention, a category definition is neither true nor false (since the meaning of the category is specified by the definition) and hence cannot be "corrected" by someone else than its creator. For example, a user identified as "fg" is perfectly entitled to define a category `fg#cat` as a subtype of the WordNet category `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 a user tries to enter a new belief that would introduce a redundancy or an inconsistency detected by the system, this addition is rejected by the system. The user may then either correct this belief or re-enter it again but connected by specialization relations (e.g. "example") or "corrective relations" (e.g., "corrective_restriction") to each belief it is redundant or inconsistent with.
  For example, here is a Formalized-English statement by Joe that corrects an earlier statement by John:
  ` `any bird is agent of a flight'(John) has for corrective_restriction
  `most healthy French birds are able to be agent of a flight' '(Joe).
  The use of corrective relations 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.

In all my experiments, the hypothesis behind this approach proved to be be correct:

- Semantic conflicts can always be solved by adding more precision until they boil down to different "observations" or "preferences". Solving conflicts increases the organization of the KB.
- Solving conflicts (or, more precisely, avoiding them by making them explicit) can be done incrementally, without the users having to discuss or change their beliefs.
- Different, internally consistent, ontologies do not have to be structurally modified to be integrated (strongly inter-related) into a unique consistent semantic network. It it happens that two ontologies provide mutually inconsistent definitions for a same category, a manual or automatic "cloning" of one of the categories can solve the problem. I give an algorithm for a "minimal cloning approach" in the Annex 2 of my PhD thesis (Martin, 1996).

2.2. Proposing a large and dynamically updatable default ontology that merges and extends other ontologies without loss of information

Tackled problem. As previously noted, a large default ontology is needed to ease knowledge modelling, sharing and retrieval, and users should be able to complement this ontology. Such an ontology should include as many conceptual distinctions as possible, and hence should integrate many other ontologies: top-level ontologies, lexical ontologies, KA ontologies, language ontologies (e.g., OWL) and, for modelling within particular domains, ontologies related related to theses domains. Because all conceptual categories are useful, and in order to ease future integrations of newer versions of source ontologies, this integration should not modify the meaning of the categories from the source ontologies. Finally, each category should have at least one intuitive identifier for KRs using this category to be easy to read. This implies that an identifier should not be nor include numbers but, apart from the prefix or suffix corresponding to the creator or source of this category, should either be a common word or, in case of ambiguity, an easy-to-read expression.

Found solutions.

- WebKB-2 is a Web-accessible KBMS (KB Management System) implemented on top of a DBMS but in a sufficiently generic way to permit dynamic changes to the ontology (Martin et al., 2005). For efficiency purposes, most other KBMSs or KB servers that re-use a DBMS (e.g., Powerloom) have a fixed ontology because this one is encoded into the conceptual schema of a database. Some generic and relatively efficient KBMSs (e.g., many RDF-based DBMSs) are developed on top of a relational database and translate knowledge queries into SQL queries but this solution only works with a restricted knowledge model (e.g., most RDF-based DBMSs do not take OWL into account and many

do not even take specialization relations into account, hence cannot do any inferencing). WebKB-2 has a complex knowledge model and is efficient because it re-uses the object-relational main/virtual memory DBMS FastDB/Gigabase (Knizhnik, 2007) which has all the classic features of a DBMS.

- The implementation of the solutions introduced in the previous sub-sections allow the creation of a "multi-source ontology" and its extension by Web users. The term "multi-source ontology" was initially coined by Sowa (1993) to refer to the large default ontology I created by converting WordNet 1.7 into a genuine lexical ontology, integrating various top-level ontologies into it, and complementing them with conceptual categories and schemas or definitions. These complements were needed for (i) knowledge checking, (ii) structuring the ontology and thus easing its browsing, understanding, use and extension, and (iii) easing general knowledge representation (many of the added concept types and 200 relation types are necessary for quickly creating precise and normalized KRs; the combinable "cascading knowledge entering/querying forms" of WebKB-2 exploit schemas or definitions). Examples of top-level ontologies that I integrated are: those of Sowa, Dolce, the SUMO, the Lifecycle Integration Schema, the Natural Semantic Metalanguage, OWL, DAML+OIL, KIF and the Dublin Core. The result was voted a "candidate material for a standard" by the voting members of the IEEE SUO (SUO, 2004).
  Transforming WordNet into a genuine lexical ontology implied (i) correcting the 300 semantic inconsistencies that WebKB-2 detected, especially after I added a top-level ontology on top of WordNet (without over-interpreting the meaning of its categories), and (ii) distinguishing its generalization relations into subtypeOf and instanceOf links (I isolated 6211 instance relations by following the knowledge sharing best practices I gathered). I also created an algorithm to generate an intuitive identifier for each category (word meaning) based on the words that the categories represent the meanings of. Finally, I associated various meta-information to those categories to ease or, on the contrary, prevent their use since some categories of WordNet have a lexical nature rather than a semantic nature. No other work on WordNet has been so extensive: most of them, for example, OntoWordNet, only worked on its top-level ontology.
  The result has been exported in static files (which have been used by several researchers), including a RDF file, as for most of the "adaptations" that have been made of WordNet, e.g., OntoWordNet. However, any category or any portion of that ontology can also be queried from within WebKB-2 or from a distinct application, and displayed or exported in various languages and according to various options. A consequence is that each category has a URI, a point which was the only purpose of the "xmlns.com Wordnet vocabulary service". More detail on this work, its difficulties, results and the methodology I used for it can be found in (Martin, 2003, 2003b). A similar approach can be used for the integration of other ontologies or WordNet-like lexical databases.
- I added knowledge about various domains to the default KB of WebKB-2. For example, I represented and organized the content of the learning resources for three courses given at Griffith University (Multimedia, Systems Analysis & Design and Workflow Management) in order to help their students better relate and understand the various pieces of information scattered all along the lecture materials (Martin, 2006, 2006b). This help was recognized by the students even if the current absence of graphical interface or structured interface for knowledge visualization in WebKB-2 proved to be an obstacle. I have also represented "structured discussions" (as illustrated in Table 3) about various subjects (Martin, 2006b), begun an ontology of Knowledge Engineering and, as part of this task, found a way to structure information in a visually scalable way within static files (Martin et al., 2005) (Martin & Eboueya, 2007). The ultimate goal of these last two works is to create a sufficiently organized formal and semi-formal KB for researchers and lecturers in Knowledge Engineering to be able to complement it in a scalable way with their own ideas and knowledge, thus creating a collaboratively built semi-formal state of the art in Knowledge Engineering. This project is further described in my research proposal. Then, other states of the art or corporate memories could be similarly built. In 2005, I organized a workshop about such semi-formal repositories and the use of structured discussions to build them (Martin, 2005).

2.3. Enabling a precise valuation and filtering of knowledge and knowledge sources

Tackled problem. Current approaches to valuating information or ontologies are restricted to the association of formal or informal annotations or the averaging numerical attributes - the approach of Euzenat (1996) for generating consensual sub-KBs can also be seen as a restricted form of valuation. For example, Knowledge Zone (Lewen et al., 2006) allows its users to rate ontologies with numerical or free text values for criteria such as "usage", "coverage", "correctness" and "mappings to other ontologies", also allows its users to rate each other users' ratings, and uses all these ratings to retrieve and rank ontologies. This approach has the problems of the "semi-independent approach" or, seen from another viewpoint, the related problems entailed by annotating *large* blocks of information (here, whole ontologies) instead of individual categories or statements: (i) whole ontologies are rarely genuinely/intuitively comparable (given two randomly selected ontologies, it is very rare that one fully includes or specializes the other), (ii) giving numerical values for such criteria is rather meaningless, (iii) textual values for each of such criteria cannot be automatically organized into a semantic network, (iv) two sets of criteria are rarely comparable (one set rarely includes all the criteria of the other set and has higher values for all these criteria), and (v) similarity measures on criteria only permit to retrieve possibly "related" ontologies: the work of understanding, comparing or merging their statements still has to be (re-)done by each user.
The languages, normalization guidelines and editing protocols presented so far ease and encourage the semantic interconnection of statements but need to be complemented by a voting system to permit a cooperative evaluation of the originality, popularity, acceptation and other characteristics related to the "usefulness" of a statement or KB user. Such valuations provide useful knowledge filtering or presentation criteria and hence are important for IR or to remove the need for committees or special users to judge what is of interest or not. They should also provide incentives for knowledge providers to create precise and original statements, or refine them when necessary. These last two points entail that there should be a default valuation mechanism but that users should be able to personalize it for their own purposes.

Found solutions. In (Martin et al., 2006), I gave a template algorithm that satisfies the above specifications. It quantifies the usefulness of each statement in a KB, and then also on each of their creators, based on votes from users on statements and on how each statement is (counter-)argued using argumentation relations. With this algorithm, using a different pseudo when providing low quality statements is not an effective turn-around strategy since this reduces the number of authored statements for other pseudos. When a belief is counter-argued, the usefulness of its author decreases, and hence this user is incited to refine its faulty statements, argue for them, or remove them. The core ideas behind this algorithm are presented in Table 6.
A primitive and informal version of our statement valuation approach was implemented in SYNVIEW (Lowe, 1985). For purposes similar to those of Lowe, Buckingham et al. (2007) continue to extend ScholOnto with the ultimate dream (which I share) of creating an Internet based infrastructure supporting a more effective dissemination, debate, and analysis of ideas than the current system of article publication. However, their approach does not yet include a voting system, it is currently only based on enabling argumentation structures such as those in Table 3, with several supporting tools and more guidelines for the choice of argumentation but with no specialization relations (hence, no inheritance), no meta-statement (hence, no distinction between "an objection to a statement" and "an objection to the use of a statement as an argument or objection to another statement") and no normalization guidelines - hence, without scalable way to design an organized network of statements. In his description of a "Digital Aristotle", Hillis (2004) describes a "Knowledge Web" to which teachers and researchers could add "isolated ideas" and "single explanations" at the right place, and suggests that this Knowledge Web could and should "include the mechanisms for credit assignment, usage tracking, and annotation that the Web lacks" (pp. 4-5), thus supporting a much better re-use and evaluation of the work of a researcher than the current system of article publishing and reviewing. Hillis does not give any indication on such mechanisms but those proposed in this sub-section and the previous ones seem to form a basis. In my research proposal, I plan to extend this basis.

Table 6. Core ideas for the valuation of knowledge and knowledge sources.

Let the users relate statements - e.g., (counter-)arguments - to statements, as in Table 3.
Let each user personalize the default valuation of the **usefulness of a statement**
based on its "weighted average interest" and, if the statement is not a definition, its
"state of confirmation". Let each user use this criteria in its filtering and display specifications.

- Calculate the **weighted average interest** of a statement by taking into account
  the users' "individual interests" (their votes) for this statement
  and the "usefulness" of these users
  - Calculate the **usefulness of a user** based on the "usefulness of her
    statements" and her participation to valuating statements of other authors
- Calculate a value for the **state of confirmation** of a belief
  based on how its (counter-)arguments are (counter-)argued

3. References

Since this document refers to almost all my publications, their references are not included here, they can be found in my Curriculum Vitae (which is included in this application).

Buckingham Shum S.J., Uren V., Li G., Sereno B. & Mancini C. (2007). *Modelling Naturalistic Argumentation in Research Literatures: Representation and Interaction Design Issues*. International Journal of Intelligent Systems, (Special Issue on Computational Models of Natural Argument, Eds: C. Reed and F. Grasso, 22, (1), pp. 17-47.

Dang H.T., Lin J. & Kelly D. (2006). *Overview of the TREC 2006 Question Answering Track*. Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006), NIST Special Publication SP 500-272.

Dromey G. R. (2006). *Scaleable Formalization of Imperfect Knowledge.* Proceedings of AWCVS-2006, 1st Asian Working Conference on Verified Software, 29-31 October 2006, Macao SAR, China.

Euzenat J. (1996). *Corporate memory through cooperative creation of knowledge bases and hyper-documents*. Proceedings of 10th KAW, (36)1-18, Banff, Canada, Nov. 1996.

Euzenat J., Stuckenschmidt H. & Yatskevich M. (2005). *Introduction to the Ontology Alignment Evaluation 2005* Proceedings of K-Cap 2005 (pp. 61-71), workshop on Integrating ontology, Banff, Canada, 2005.

Fensel D., Decker S., Erdmann M. & Studer M. (1998). *Ontobroker: Or How to Enable Intelligent Access to the WWW*. Proceedings of KAW'98 (11th Knowledge Acquisition Workshop), pp. 8-23, Banff, Canada, 1998.

Fielding R.T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Ph.D. Thesis, University of California, Irvine, Irvine, California, 2000.

Haemmerlé O. & Guinaldo O. (1999). *CoGITo v3.3 : plate-forme de développement d'applications sur les graphes conceptuels.* Technique et Science Informatique, 18 (9), pp. 933-965, November 1999.

Hillis W.D. (2004). *Aristotle (The Knowledge Web)*. Edge Foundation, Inc., No 138, May 6, 2004.

Knizhnik K. (2007). *FastDB: a main-memory database object-relational database system.* Available at http://www.garret.ru/knizhnik/fastdb.htm

Lewen H., Supekar K.S., Noy N.F. & Musen M.A. (2006). *Topic-Specific Trust and Open Rating Systems: An Approach for Ontology Evaluation*. Proceedings of EON'06 (Evaluation of Ontologies for the Web) at WWW'06, Edinburgh, UK.

Lowe D. (1985). *Co-operative Structuring of Information: The Representation of reasoning and debate*. International Journal of Man-Machine Studies, 23(2), pp. 97-111., August 1985.

Marshall C.C. &. Shipman F.M. (2003). *Which Semantic Web?*. Proceedings of ACM Hypertext 2003, pp. 57-66.

Patel-Schneider P.F. (2005). A Revised Architecture for Semantic Web Reasoning. *PPSWR 2005* (LNCS 3703, pp. 32-36), Principles and Practice of Semantic Web Reasoning: Third International Workshop, Dagstuhl Castle, Germany, September 11-16, 2005.

Pool J. (2006). *Can Controlled Languages Scale to the Web?*. Proceedings of CLAW 2006 (5th International Workshop on Controlled Language Applications), August 12, 2006.

Quint V. & Vatton I. (1992). *Combining Hypertext and Structured Documents in Grif.* Proceedings of ECHT'92, D. Lucarella, ed., pp. 23-32, ACM Press, Milan, December 1992.

Sowa J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove, CA.

Sowa J.F. (2003). *Re: Ontology Registry*. Message 11841 of the IEEE Standard Upper Ontology list, 26 Nov 2003. http://suo.ieee.org/email/msg111841.html

SUO (2004). *MSO Ballot Results*. Message 12552 of the IEEE Standard Upper Ontology list, 12 May 2004. http://suo.ieee.org/email/msg12552.html

Wang B., Mckay R.I., Abbass H.A. & Barlow M. (2003). *A comparative study for domain ontology guided feature extraction.* Proceedings of ACSC-2003, 26th Australian Computer Science Conference, pages 69-78. Australian Computer Society, 2003.