Collaborative Ontology Modelling

Collaboratively Built, Evaluated and Distributed Ontologies

Philippe A. Martin
ESIROI STIM, University of La Réunion
(+ adjunct researcher of Griffith Uni., Australia)
Sainte Clotilde, La Réunion, France

Abstract—**This article first summarizes problems of the classic Semantic Web approaches to knowledge sharing and, to solve this problems, presents i) a protocol permitting people to collaboratively build a well-organized KB without having to discuss or agree, and ii) a protocol for a global virtual KB well-organized to be based on individual well-organized KBs via a partially mirroring between them. This article also presents some applications and a way to allow a precise collaborative evaluation of information providers and pieces of information.**

Knowledge sharing/integration/retrieval/evaluation/distribution; collaboration; Semantic Web

I. Introduction

The more organized information are in a repository, that is, the more explicitly related by semantic relations the stored objects (i.e., the pieces of information) are, the easier it is to retrieve, compare, interpret and manage them, manually and automatically. In this viewpoint, the ideal repository is a logic-based formal (and yet easy-to-read) "semantic network" whose objects are *maximally-connected* at least according to the most important and well known kinds of semantic relations, such as those about equivalence, generalization (logical deduction, super-type, instance-of, more-general-term, ...), part-hood (e.g., subprocess, substance, physical-part, …), theme/case (agent, object, recipient, instrument, …), argumentation, time (duration, date, before, …), space, source (author, …) and other "context description" relations.

Current repositories of ontologies – or, more generally, of knowledge bases (KBs) - and the current Semantic Web, are extremely far from such an ideal. First, within most ontologies, objects are hardly systematically organized. Second, most ontologies are small, designed according to different and implicit principles, and very poorly interconnected, hence with partial/total redundancies or inconsistencies between that are not made explicit via semantic relations and that often cannot be detected automatically in a logic-grounded way. Indeed, this is how the Semantic Web is nowadays generally expected to be [3] [21] [22]: based on rather independently developed static formal files. The implicit (i.e., not explicitly represented) relations between the objects of different files are difficult to guess/discover by people, let alone machines: often, only the object authors know what their objects really mean and hence only these authors can often correctly relate them to other objects. Most *current semi-automatic Semantic Web tools* are intended to alleviate the difficulty of retrieving, comparing and merging semi-independently developed ontologies. Such tools are useful but , in a sense, contribute to the problem they are partially alleviating since their outputs are new (additional) formal files whose objects are not related via explicit semantic relations to the objects of most other existing formal files. Most *current Semantic Web tools that are not semi automatic* are private KB editors or shared KB servers/editors (e.g., Ontolingua, OntoWeb, Ontosaurus, Freebase, and semantic wiki servers) which either i) let every authorized user modify what other ones have entered in the shared KB (this discourages information entering or leads to edit wars), or ii) require all/some users to approve or not changes made in the KB, possibly via a workflow system (this is bothersome for the evaluators, may force them to make arbitrary selections, and this is a bottleneck in information sharing that often discourages information providers).

This article presents ways to support a *complementary* approach that relies less on semi-automatic tools but more directly tends to the above cited ideal repository. These are supports for a "collaboratively-built at-least-minimally-well-organized large KB" (cbwoKB) where – to allow incremental collaborative construction – the *objects (statements and concept/relations terms for statements)* may be formal (i.e., with a unique meaning), informal or – for statements – semi-formal, i.e., with a formal grammar but with formal or informal terms. In this article, ***at-least-minimally-well-organized*** means manually or automatically detected *partial redundancies or inconsistencies are prevented or made explicit via relations of i) specialization and part-hood, and/or ii) identity or correction*. This implies that every object of the KB has a unique place in the global specialization hierarchy and global part-hood hierarchy (which, in other words, are unique, i.e., fully connected, and are not "trees" but, to ease knowledge comparison, may have the classic added constraint of being "lattices"; this option is not explored in this article). This "unique place", i.e., the absence of implicit redundancies, is a minimal requirement for knowledge insertion and retrieval to be done in a scalable way in the hierarchies and hence in the semantic network of which they are the backbones [6]. Furthermore, for scalability purposes and for the reasons given at the end of the previous paragraph, ***information integration*** in a (general-purpose) cbwoKB has to be "***loss-less***": i) no choice between conflicting knowledge should be made by people other than by end-users themselves to suit their own preferences or the requirements of their application, ii) people should not be forced to meet, discuss or agree on terminology or beliefs. Finally, for decentralization purposes, it should be possible to inter-relate physical cbwoKBs (i.e., those managed by a unique Web server, e.g., the servers of communities or the servers on machines of a peer-to-peer network) to create a ***global virtual cbwoKB*** (gv-cbwoKB).

Section II quickly compares different approaches to knowledge sharing. Section III presents rules that support and enforces the collaborative edition of a cbwoKB. Section IV introduces a framework for the collaborative evaluation of knowledge representations. These supports have been fully or partially implemented in a shared KB server named WebKB-2 [16] (webkb.org) along with other supports which are necessary for a general cbwoKB but which are not represented here because of space restrictions: i) a set of "best practices" for knowledge organization and normalization [18], iii) a large general ontology for English that also integrates many current top-level ontologies [15], and iii) several complementary expressive-but-intuitive knowledge representation notations [14] [18]. Section V presents some applications of WebKB-2 for (e-)learning. Section VI shows a way to support a gv-cbwoKB. Section VII concludes.

II. Quick Overview of Knowledge Sharing Approaches

Current knowledge sharing approaches are generally based on i) the independent creation of formal documents (e.g., in RDF/XML and OWL) or documents including some knowledge representations (e.g., in RDFa), ii) the shared use of database servers the schema of which are represented in formal files, iii) the shared use of KB/ontology servers which, as seen in the introduction, have no loss-less knowledge-integration protocol, and iv) the shared creation of mostly informal documents, as in semantic wikis. Within one fully/partially formal document/KB, knowledge retrieval can lead to an exact answer (e.g., when the query is of the kind "what are the resources/tools/methods to do ...", a sub-network of the KB composed of subtask/specialization/argumentation relations). When several files (documents or KBs) are involved, due to the lack of a unique semantic network (or, at least, a common ontology), information retrieval (IR) leads to a list of possibly relevant documents or statements where original/precise ones are often hidden among/behind those that are more general, mainstream or from big organizations. More generally, IR quality decreases when the size and number of the files increases, but not when the number of objects increases in a well-organized KB.

The more objects two files contain, the more difficult it is to link these files via semantic relations and hence to semantically compare, organize and evaluate them. Instead, similarity/distance (statistical) measures have to be used. In a cbwoKB, when needed, semantic queries can be used to filter objects or generate files, according to arbitrary complex combinations of criteria, e.g., about the creators of the objects. Ontology libraries, from early ones such as the Ontolingua library to imagined ones such as "The Lattice of Theories" [23], are often organized into "minimal and internally consistent theories" to maximize their re-use. However, this also leads to few relations between objects of different ontologies, as well as implicit redundancies or inconsistencies between them, and hence more difficulties to compare, merge or relate them. On the other hand, as acknowledged by the author of [23], if the objects are organized into a cbwoKB, such (lattices of) theories can be generated via queries.

With formal files as inputs and outputs, knowledge re-use or integration leads to the creation of even more files and requires people to select, compare, relate, merge, adapt and combine (parts of) files. Except for simple applications where fully automatic tools can deliver good-enough results, these are complex tasks that have to be done by trained people who know the domain. Most works in collaborative knowledge sharing or "ontology evolution in collaborative environments" are about (semi-)automatic procedures for integrating two ontologies [8] and for rejecting or integrating changes made in other ontologies, e.g., [3] [19] [21]. In a cbwoKB, no adaptation or integration has to be done for each re-use: important relations from an object simply have to be entered by its creators and can then be complemented by any user.

III. KB Sharing Protocol

This protocol is intended to keep the cbwoKB "at-least-minimally-well-organized" in the sense given in the introduction. It is not tied to any particular knowledge representation language (KRL) or inference mechanism. All it requires is for *"actual/potential conflicts" (i.e., total/partial inconsistencies/redundancies between statements of the KB)* to be detected by some inference engine (as shown below, this does not imply that the KR language should be restricted), or for users to tell the system about conflicts they believe exist. This KRL independence is clear in the high-level algorithms which are given below in Java (and, for clarity purposes, in an object-oriented way) and then discussed.

These algorithms present some checks on a user's attempt to remove or add a statement and the resulting system decision: rejecting the action ("return false") or accepting it, with possibly some repair step before accepting. Only statement removal and adding are considered since i) updating is considered as removal followed by adding, ii) reading or re-using an object (term or statement) is always accepted (privacy control is not dealt with in this article), and iii) term removal or adding must be made via the removal or addition of a statement. Indeed, an additional rule of the protocol – not presented in the algorithms below – is that a new term can only be added (vs. removed) by specializing or aliasing another term (vs. removing this specialization or identity relation), except for processes which, for convenience purposes, can also be added (vs. removed) via sub-process relations. Giving a definition is equivalent to using a specialization/identity relation, except that the system can exploit the definition to better place the term in the specialization hierarchy. A statement is either a definitions or a belief. Every belief is also automatically inserted in the specialization hierarchy and its place may be refined by its creator if this does not introduce an inconsistency in the KB.

In order to have a unique specialization/generalization hierarchy and hence be able to compare any pair of formal or informal objects (i.e., know if one generalizes or specializes the other), this hierarchy must actually use several kinds of specialization relations (all of which being subtypes of an "extended-specialization" relation type): i) the classic "subtype" and "instance" relations between formal terms, ii) the classic "logical-deduction-of" between formal statements (which, when formal terms have definitions, permits to calculate or check subtype/instance relations between these terms), and iii) an "informal-generalization" from a formal or informal object to an informal one.

WebKB-2 uses the same graph-matching technique for calculating partial or total extended-specialization relations between formal/informal statements, and therefore also "actual or potential conflicts". Other inference mechanisms could be used instead *or in addition for detecting more specialization relations.* One advantage of this graph-matching is that it can be computed in polynomial time if one of the graphs (statements) has no cycle [4]. Another advantage is that although from a logical viewpoint it is not sound and complete when the graphs are not simply "positive conjunctive existential formulas, possibly including a positive context (i.e., a meta-statement that does not restrict its truth domain)", it can be applied to any graph, however expressive it is, and always be "relevant" from an "extended-specialization" viewpoint. A statement Y is an extended specialization of a statement X if X structurally matches a part of Y and if each of the terms in this part of Y is identical or an extended specialization of its counterpart term in X. For example, WebKB-2 can detect that the Formalized English (FE) statement `u2#`Tweety can be agent of a flight with duration at least 2.5 hour´` (which means "u2 believes that Tweety can fly for at least 2.5 hours") is an extended specialization (and an "extended instantiation") of both `u1#`2 bird can be agent of a flight´` and `u1#`every bird can be agent of a flight´`. In KIF (Knowledge Interchange Format) [9], a representation of this last statement can be:

```
(believer u1 '(modality possible '(forall ((?b bird))
                (exists ((?f flight)) (agent ?b ?f)))))
```

Furthermore, these last two FE statements can respectively be found to be extended specializations of `u2#`75% of bird can be agent of a flight´` and `u2#`at least 1 bird can be agent of a flight´`. Similarly, this last graph can be found to be exclusive with `u3#`no bird can be agent of a flight´`.

To avoid lexical conflicts and permit knowledge filtering on the creator of objects, every object in the KB has at least one associated "source" (creator, believer, interpreter, source file or language) which itself is represented by a formal term. In the above example knowledge representations, the terms are formal but, for readability purposes, their creators are not shown; `wn2.1#flight` is an example in FE of formal term in WordNet 2.1 that represents of of the meaning of the informal English term `en#"flight"`; such a prefixing avoid lexical conflicts between homonym formal/informal terms from different sources). In the following algorithms, the word "user" is used as a synonym for "source".

```
boolean statement.removal_by (User agent)
{ if (object.creator != agent) return false;
  if (agent.created_statements.are_inconsistent_with(this))
    return false;
  if (agent.created_statements.are_redundant_with(this))
    return false;
  if (this.is_definition())
  { if (KB.statements_without(this).are_inconsistent())
      KB.clone_term_in_statements_using(this.defined_term());
  }
  else if (KB.statements_without(this).are_inconsistent())
         this.clone_for_other_believers();
  KB.remove(this,agent);  return true;
}
```

```
boolean statement.adding_by (User agent)
{ if (this.is_informal_statement() &&
      !this.has_associated_argumentation_relation())
     return false;
  if (agent.created_statements.are_inconsistent_with(this))
    return false;
  if (agent.created_statements.are_redundant_with(this))
    return false;
  if (this.is_definition())
  { if (this.is_definition_of_new_term() &&
        KB.statements.are_inconsistent_with(this))
      return false;
    if (this.is_new_definition_of_already_declared_term() &&
        KB.statements.are_inconsistent_with(this))
      KB.clone_term_in_statement_inconsistent_with(this);
  }
  else if (KB.statements.are_partially_conflicting_with(this))
         return false;  //"implicitly redundant/inconsistent"
  KB.add(this,agent);  return true;
}
```

Here are the informal rules enforced by these algorithms.

1) *Any user can add and use any object* but *an object may only be modified or removed by its creator*.

2) *If adding, modifying or removing a* ***statement*** *introduces a detected* ***implicit redundancy*** *in the shared KB,* ***or*** *if this introduces an* ***inconsistency between statements believed by the user*** *having done this action*, this ***action is rejected*** by the system. Thus, in the case of an addition, the user must refine his statement before trying to add it again or he must first modify at least one of his already entered statements. An "implicit" redundancy is a redundancy between two statements without a relation between them making the redundancy explicit, typically an equivalence relation or an extended specialization relation (e.g., an "example" relation). The detection of implicit extended specializations between two objects reveals an inconsistency or a total/partial redundancy. It is often not necessary to distinguish between these two cases to reject the newly entered object. Extended "instantiations" are exceptions (see the example given above): since adding an instantiation is giving an example for a more general statement, it does not reveal a redundancy or inconsistency that needs to be made explicit. It is important to reject an action introducing a redundancy instead of silently ignoring it because this often permits the author of the action to detect a mistake, a bad interpretation or a lack of precision (on his part or not). At the very least, this reminds the users that they should check what has already been represented on a subject before adding something on this subject.

3) *If the addition of a* ***new term u1#T*** by a user u1 *introduces an* ***inconsistency with statements of other users***, this ***action is rejected by the system***. Indeed, such a conflict reveals that u1 has directly or indirectly used at least one term from another user in his definition of u1#T and has misunderstood the meaning of this term. The *addition by a user u2 of a definition to u1#T is actually a belief of u2* about the meaning of u1#T. This belief should be *rejected if it is found (logically) inconsistent with the definition(s) of u1#T by u1*. An example is given in Point 6.

4) *If the addition, modification or removal of a statement* ***defining an already existing term u1#T*** by a user u1 *introduces an* ***inconsistency involving statements directly or indirectly re-using u1#T and created or believed by other users*** (i.e., users different from u1), ***u1#T is automatically cloned*** to solve this conflict and ensure that the original interpretation of u1#T by these other users is still represented. Indeed, such a conflict reveals that these other users had a more general interpretation of u1#T than u1 had or now has. Assuming that u2 is this other user or one of these other users, the term cloning of u1#T consists in creating u2#T with the same definitions as u1#T except for one, and then replacing u1#T by u2#T in the statements of u2. The difficulty is to chose a relevant definition to remove for the overall change of the KB to be minimal. In the case of term removal by u1, term cloning simply means changing the creator's identifier in this term to the identifier of one of the other users (if this generated term already exists, some suffix can be added). In a cbwoKB server, since statements point to the terms they use, changing an identifier does not require changing the statements. In a global virtual cbwoKB distributed on several servers, identifier changes in one server must be replicated to other servers using this identifier. Manual term cloning is also used in knowledge integrations that are not loss-less [5].

In a cbwoKB, it is not true that beliefs and term definitions "have to be updated sooner or later". Indeed, in a cbwoKB, every belief must be contextualized in time and space as in `u3#` `` `75% of bird can be agent of a flight´ in place France and in period 2005 to 2006´ `` (such contexts are not shown in the other examples of this article). If needed, u3 can associate the term `u3#75%_of_birds_fly__in_France_from_2005_to_2006` with this last belief. Due to the possibility of contextualizing beliefs, it is rarely necessary to create formal terms such as `u2#Sydney_in_2010`. Most common formal terms, e.g., `u3#bird` and `wn1.7#bird` never need to be modified by their creators. They are specializations of more general formal terms, e.g., `wn#bird` (the fuzzy concept of bird shared by all versions of the WordNet ontologies). What evolves in time is the popularity of a belief or the popularity of the association between an informal term and a concept. If needed, this changing popularity can be represented by different statements contextualized in time and space.

5) *If adding, modifying or removing a* ***belief*** *introduces an* ***implicit potential conflict (partial/total inconsistency or redundancy) involving beliefs created by other creators***, it is ***rejected***. However, a user may represent his belief (say, b1) – and thus "loss-less correct" another user's belief that he does not believe in (say, b2) – by connecting b1 to b2 via a ***corrective relation***. E.g., here are two FE statements by u2, each of which corrects a statement made earlier by u1:

```
u2#` u1#`every bird is agent of a flight´
      has for corrective_restriction
       u2#`most healthy flying_bird are
            able to be agent of a flight´ ´ and
u2#` u1#`every bird can be agent of a flight´
      has for corrective_generalization
       u2#`75% of bird can be agent of a flight´ ´.
```

If instead of the *belief* `every bird can be agent of a flight´ (all birds can fly), u1 entered the *definition* `any bird can be agent of a flight´, i.e., if he gave a *definition* to the type named "bird", there are two cases (as implied by the rules of the two previous points):

a) u1 originally created this type (`u1#bird`); then, u2's attempt to correct the definition is rejected, or

b) u1 added a definition to another user's type, say `wn#bird` since this WordNet type has no associated constraint preventing the adding of such a definition; then, i) the types `u1#bird` and `u2#bird` are automatically created as clones (and subtypes of) `wn#bird`, ii) the definition of u1 is changed into `` `any u1#bird is agent of a flight´ ``, and iii) the belief of u2 is (automatically) changed into `` u2#`75% of u2#bird can be agent of a flight´ ``.

In WebKB-2, users are encouraged to provide argumentation relations on corrective relations, i.e., a meta-statement using argument/objection relations on the statement using the corrective relation. However, to normalize the shared KB, people are encouraged not to use an objection relation but a "corrective relation with argument relations on them". Thus, not only are the objections stated but a correction is given and may be agreed with by several persons, including the author of the corrected statement (who may then remove it). Even more importantly, unlike objection relations, most corrective relations are transitive relations and hence their use permits better organization of argumentation structures, thus avoiding redundancies and easing information retrieval. The use of corrective relations makes explicit the disagreement of one user with (his interpretation of) the belief of another user. There is no inconsistency: an assertion A may be inconsistent with an assertion B but a belief that "A is a correction of B" is technically consistent with a belief in B. Thus, the shared KB can remain consistent.

For problem-solving purposes, application-dependent choices between contradictory beliefs often have to be made. To make them, an application designer can exploit i) the statements describing or evaluating the creators of the beliefs, ii) the corrective/argumentation and specialization relations between the beliefs, and more generally, iii) their evaluations via meta-statements (see the next section). For example, an application designer may choose to select only the most specialized or restricted beliefs of knowledge providers having worked for more than 10 years in a certain domain. Thus, the approach of this protocol is unrelated to de-feasible logics and avoids the problems associated with classic "version management" (furthermore, as above explained, in a cbwoKB, formal objects do not have to evolve in time).

This approach assumes that all beliefs can be argued against and hence be "corrected". This is true only in a certain sense. Indeed, among beliefs, one can distinguish "observations", "interpretations" ("deductions" or "assumptions"; in this approach, axioms are considered to be definitions) and "preferences"; although all these kinds of beliefs can be false (their authors can lie, make a mistake or assume a wrong fact), most people would be reluctant to argue against self-referencing beliefs such as `u2#"u2 likes flowers"` and `u2#"u2 is writing this sentence"`. Instead of formalizing this into exceptions, the editing protocol of WebKB-2 relies on the reluctance of people to argue against such beliefs that should not be argued against.

Before browsing or querying the cbwoKB, a user should be given the opportunity to set "filters for certain objects not to be displayed (or be displayed only in small fonts)". These filters may set conditions on statements about these objects or on the creators of these objects. They are automatically executed queries over the results of queries. In WebKB-2, filtering is based on a search for extended specialization, as for conceptual querying. Filters are useful when the user is overwhelmed by information in an insufficiently organized part of the KB.

The approach described by the previous points is incremental and *works on semi-formal KBs.* Indeed, the users can set corrective or specialization relations between objects even when the system does not detect an inconsistency or redundancy. As noted, a new informal statement must be connected via an argumentation relation (e.g., a corrective relation) or an extended specialization relation to an already stored statement. For this relation to be correct, this new statement should generally not be composed of several sub-statements. However, allowing the storing of (small) paragraphs within a statement eases the incremental transformation of informal knowledge into (semi-)formal knowledge and allows doing so only when needed. This is necessary for the general acceptance of the approach. The techniques described in this article do not seem particularly difficult for information technology amateurs, since the minimum they require is for the users to set the above mentioned relations from/to each term or statement. Hence, these techniques could be used in semantic wikis to avoid their governance problems cited in the introduction and other problems caused by their lack of structure. More generally, the presented approach removes or reduces the file-based approach problems without creating new problems. Its use would allow merging of (information discussed or provided by members of) many communities with similar interests, e.g., the numerous different communities working on the Semantic Web.

The hypotheses of this protocol are that i) conflicts can always be solved by adding more precision (e.g., by making their sources explicit: different "observations", "interpretations" or "preferences"), ii) solving conflicts in a loss-less way most often increases or maintains the precision and organization of the KB, and iii) different, internally consistent, ontologies do not have to be structurally modified to be integrated (strongly inter-related) into a unique consistent semantic network. None of the various kinds of integrations of ontologies that I made invalidated these hypotheses.

IV. Evaluating Objects and Their Sources

Many information repositories support free-text/numerical evaluations on objects or files by people and then display them or statistical measures on them. For example, Knowledge Zone [12] allows each of its users to i) rate ontologies with numerical or free text values for criteria such as "usage", "coverage", "correctness" and "mappings to other ontologies", ii) rate other users' ratings, and iii) uses all these ratings to retrieve and rank ontologies. Such evaluations have several problems: i) the evaluations are not organized into a semantic network, ii) the above examples of criteria and their numerical values are not about objects in the ontologies and hence do not help choosing between objects, iii) multi-criteria decision making is difficult since two sets of (values for) criteria are rarely comparable (indeed, one set rarely includes all the criteria of the other set and, at the same time, has higher values for all these criteria), and iv) 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. In a cbwoKB, such problems are strongly reduced since evaluations are on objects and are themselves objects: they are managed/manageable like other objects and are integrated into a network of specialization, correction and argumentation relations.

To support more knowledge filtering or decision making possibilities and lead the users to be careful and precise in their contributions, a cbwoKB server should propose "default measures" deriving a global evaluation of each statement/creator from i) users' individual evaluations of these objects, and ii) global evaluations of these users. These measures should not be hard-coded but explicitly represented (and hence be executable) to let each user adapt them - i.e., combine their basic functions - according to his goals or preferences. Indeed, only the user knows the criteria (e.g., originality, popularity, acceptance, ..., number of arguments without objections on them) and weighting schemes that suit him. Then, since the results of these evaluations are also statements, they can be exploited by queries on the objects and/or their creators. Here are comments (only general ones due to space restrictions) on the global measures that are implemented in WebKB-2.

a) A global measure of ***how consensual a belief is*** should take into account i) the number of times it has been re-used or marked as co-believed, and ii) its argumentation structure (i.e., how its arguments/objections are themselves (counter-)argued). A simple version of such a measure was implemented in the hypertext system SYNVIEW [13]. The KB server Co4 [7] had protocols based on peer-reviewing for finding consensual knowledge; the result was a hierarchy of KBs, the uppermost ones containing the most consensual knowledge while the lowermost ones were the private KBs of contributing users. Establishing "how consensual a belief is" is more flexible in a cbwoKB: i) each user can design his own global measure for what it means to be consensual, and ii) KBs of consensual knowledge need not be generated.

b) A global measure of ***how interesting a statement is*** should be based on its type (if it has one, e.g., observation, deduction, assumption, preference, ...), on its relations (especially those arguing for/against it or representing its originality, acceptance, ...), and on the usefulness of the authors of these relations (see below).

c) A global measure of ***the usefulness of a statement*** should exploit (at least) the above two measures.

d) A global measure of **the *usefulness of a user U*** should use the global measures of usefulness of U's statements and, to encourage participation to evaluations, the number of objects he evaluated.

Given these comments, the interest of enabling end-users to adapt the default measures is clear. Whichever way it is done, taking into account the above cited elements should incite information providers to be careful and precise in their contributions and give arguments for them. Indeed, unlike in traditional discussions or anonymous reviews, careless statements here penalize their authors. This may lead users not to make statements outside their domain of expertise or without verifying their facts. (Using a different pseudo when providing low quality statements does not seem to be an helpful strategy to escape the above approach since this reduces the number of authored statements for the first pseudo.) E.g., when a belief is objected to, the usefulness of its author decreases, he is incited to deepen the argumentation structure on its belief or remove it.

[10] 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). The author of [10] did not give indications on such mechanisms but those proposed in this article seem one initial basis for them.

V. Examples Of Applications

WebKB-2 has been used for integrating many ontologies [15] [18] and representing many domains. In particular, it has been used for representing and inter-connecting the most important concepts of four different courses that I gave: "Workflow Management", "Systems Analysis & Design", "Introduction to Multimedia" and "Client-Server Architecture". Nearly each sentence of each slide for these courses has been represented into a semantic network of tasks, data structures, properties, definitions, etc. Figure 1 shows an extract of a Web file that was an input file for WebKB-2 and that mixed formal and informal elements; the formal ones are in the FL notation and represent important statements (here, relations between important concepts) from a book in Workflow Management. Each statement in these figures follow the generic schema:

```
CONCEPT1 RELATION1: CONCEPT2  CONCEPT3,
         RELATION2: CONCEPT4 (sourceForRel2) ...;
```

Such a statement should be read: "any CONCEPT1 may have for RELATION1 one or many CONCEPT2, and may have for RELATION1 one or many CONCEPT3, and may have for RELATION2 one or many CONCEPT4 (relation which can be found at sourceForRel2), ...". The sources of those relations in the book and the persons who created those representations (e.g., pm and the student s162557) are indicated. When the creator of relations is not indicated, I (the user "pm") was the creator.

The students of these courses have recognised the help that the semantic network provides them in relating and comparing information otherwise scattered in many different slides and other lecture materials (an analysis of their evaluation of this teaching approach is given in [17]). However, having to learn the FL notation was perceived as a problem, especially by the students who were evaluated on their contributions to the semantic network. An intuitive table-based knowledge entering/display interface for FL should reduce this problem. Compared to an (informal) learning journal, evaluating the students on their contributions permitted a much better evaluation of whether or not they understood the nature of the important concepts and their relationships. To enter these contributions, i.e., to collaboratively complete the initial "course formal summary" that I designed for them, the students used WebKB-2. The KB editing protocols were not a problem but entering meaningful knowledge representations proved to be very difficult for the students and highlighted the necessity for very strong and very advanced semantic checking (due to its knowledge normalization procedures, WebKB-2 has stronger semantic checks than RDF+OWL inference engines but this still proved to be very insufficient).

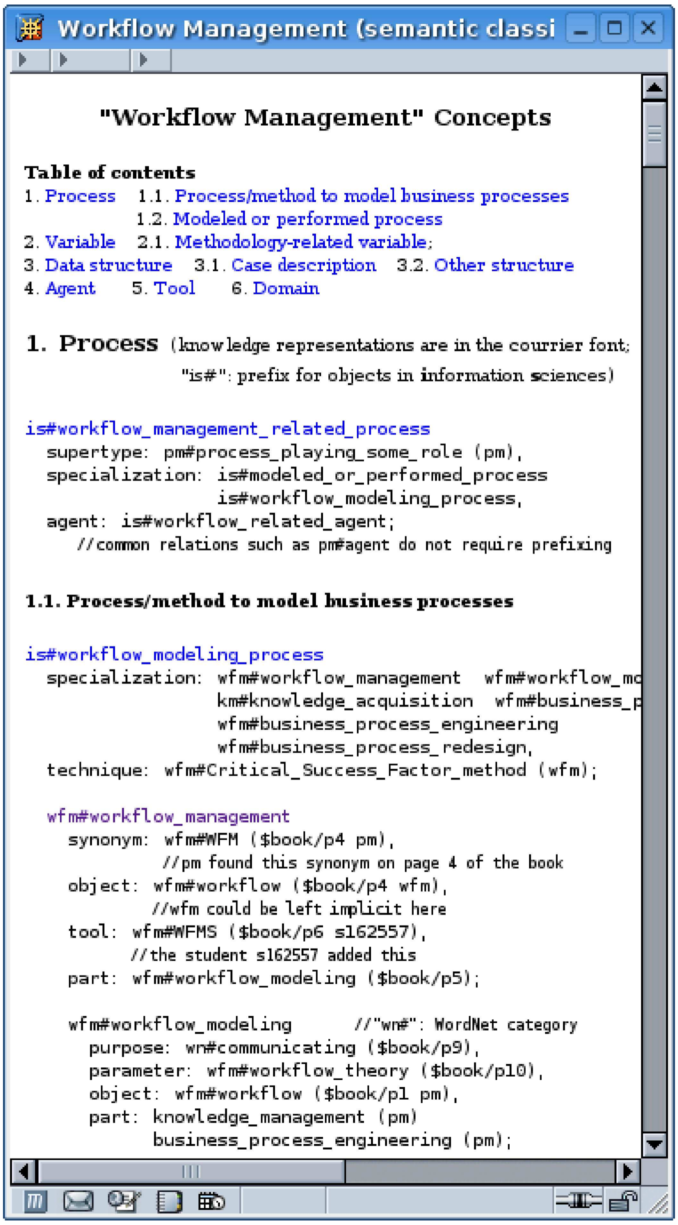

Figure 1. Extract from a Web file including some formal representations of statements from a book in Workflow Management (referred to via the variable $book).

VI. Distribution in a Virtual KB

One cbwoKB server cannot support knowledge sharing for all communities. For scalability purposes, the cbwoKB servers of communities or persons should be able to interact to act as one global virtual cbwoKB (gv_cbwoKB), without a central brokering system, without restrictions on the content of each KB, and without necessarily asking each server to register to a particular super-community or peer-to-peer (P2P) network. For several cbwoKB servers to be seen as a gv_cbwoKB, it should not matter which KB a user or agent chooses to query or update first. Hence, object additions/updates made in one KB should be replicated into all the other KBs that have a scope which covers these objects; idem for queries when this is relevant.

Given these specifications, current approaches for collaboration between KB servers/owners (e.g., the approaches of [3] [19] which are based on integrating changes made in other KBs, and those of [21] which also use a workflow system) or distributed querying between a few KB servers (e.g., as described by [11]) are insufficient. Indeed, they are based on partial descriptions of the content of each KB or on predefined roles for each KB owner or user, and the redundancies or inconsistencies between the KBs are not made explicit. This often makes difficult to find the relevant KBs to search/add in and to integrate query results.

As in the previous sections, a solution is to let the knowledge indexation and distribution be made at the object level instead of the document/KB/community/owner level. The requirement is that for every term T stored in a cbwoKB server, the KB must ***either***

a) have a ***Web-accessible formal description specifying that it is committed to be a "nexus" for T***, i.e., that i) it stores any statement S on T (if S is inserted in another KB of this gv_cbwoKB, it is also inserted in this KB), or ii) it associates to T the URLs of cbwoKB servers permitting to find or store any statement on T, ***or***

b) not be a "nexus" for T, and hence associate to T either i) the URLs of all cbwoKB servers that have advertised themselves to be a nexus for T, or ii) the URL of at least one server that stores these URLs of nexus servers for T.

Thus, via forwards between servers, all objects using T can be added or found in all the nexus for T. This requirement refines the 4th rule of the Linked Data approach [2]: "link things to their related ones in some other data sets". Indeed, to obtain a gv_cbwoKB, the data sets must be cbwoKB servers and there must be at least one nexus for each term. A consequence is that when the scopes of two nexus overlap, they share common knowledge and there is no *implicit* redundancies or inconsistencies between them. Thus, the gv_cbwoKB has a unique ontology distributed on the various cbwoKB servers.

The difficult task is, whenever the owners of a new cbwoKB server want to join a gv_cbwoKB, to integrate their ontology into the global one (they must find some nexus of the gv_cbwoKB, only one if it has a nexus for its top level type). This integration task is at the core of most knowledge sharing/re-use approaches. In this one, it is done only by the owners of the new cbwoKB; once this is done, regularly and (semi-)automatically integrating new knowledge from/to other nexus is much easier since a common ontology is shared. Thus, it can be envisaged that one initial cbwoKB server be progressively joined by other ones to form a more and more general gv_cbwoKB.

The key point of the approach is the formal commitment to be a nexus for a term (and hence to be a cbwoKB since direct searches/additions by people must be allowed). There is currently no standard vocabulary to specify this, e.g., from the W3C, the Dublin Core and voiD (a vocabulary for discovering linked datasets). To specify - and commit to - the processes related to i) being a nexus, and ii) being a cbwoKB, the ontology of WebKB-2 proposes the process types named "integrating-all-published-information-specified-as-parameter" and "supporting-the-collaborative-building-of-a-KB". The described cbwoKB server is specified via an "agent" relation to those kinds of processes. For the first kind, a "parameter" relation is used for specifying the set of types for which the cbwoKB is a nexus. For the first kind, the KB is specified via an "output" relation. Any other "relation-fom/to-a-process" (defined in the ontology of the cbwoKB, e.g., input and period) can be used for further describing what the cbwoKB commits to do. Thus, this specification approach is flexible. It would not have been the case if relation types had been used instead of process types since all the parameters to those relation types would have had to be predefined.

It is in the interest of a competitive company to advertise that it hosts a nexus for a certain term, e.g., `apartment-for-rent-in-Sydney` for a real estate agent covering the whole of Sydney. If the actual coverage of a nexus is less than the advertised one, a competitor may publish this. In a business environment, it is in the interest of a competitive company to check what its competitors or related companies offer and, if it is legal, integrate their public information in its cbwoKB. It is also in its interest to refer to the most comprehensive KBs/nexus of its related companies. To sum up, the approach could be technically and socially adopted. Since its result is a gv_cbwoKB, it can be seen as a way to combine advantages commonly attributed to "distributed approaches" and "centralized approaches".

VII. Conclusion

This article first aimed to show that a (gv_)cbwoKB is *technically and socially* possible. To that end, Section III presented a protocol permitting, enforcing or encouraging people to *incrementally* interconnect their knowledge into a well-organized (formal or *semi-formal*) KB without having to discuss and agree on terminology or beliefs. As noted, it seems that all other knowledge-based cooperation protocols that currently exists work on the comparison or integration of whole KBs, not on the comparison and loss-less integration of all their objects into a same KB. Other required elements for a (gv_)cbwoKB - and for which WebKB-2 implements research results - were also introduced (Section IV and Section VI) or simply mentioned: expressive and normalizing notations, methodological guidance, a large general ontology, and an initial cbwoKB core for the application domain of the intended cbwoKB.

Already explored kinds of applications were cited. One currently explored is the collaborative representation and classification by Semantic Web experts of "Semantic Web related techniques". This means that in the medium term Semantic Web researchers will be able and invited to represent and compare their techniques in WebKB-2, instead of just indexing their research via domain related terms, as was the case in the KA(2) project [1] or with the Semantic Web Topics Ontology [20]. More generally, the approach proposed in this article seems interesting for collaboratively-built corporate memories or catalogues, e-learning, e-government, e-science, e-research, etc.

A second aim of this article was to show that - in the long term or when creating a *new* KB for *general* knowledge sharing purposes - using a cbwoKB does/can provide more possibilities, with *on the whole* no more costs, than the mainstream approach [2] [21] [22] where knowledge creation and re-use involves searching, merging and creating (semi-)independent (relatively small) ontologies or semi-formal documents. The problem - and related debate - is more social (which formalism and methodology will people accept to learn and use?) than technical. A cbwoKB is much more likely to be adopted by a small communities of researchers but could incrementally grow to a larger and larger community. In any case, research on the two approaches are complementary: i) techniques of knowledge extraction or merging ease the creation of a cbwoKB, ii) the results of applying these techniques with a cbwoKB as input would be better, and iii) these results would be easier to retrieve, compare, combine and re-use if they were stored in a cbwoKB.

References

[1] V.R. Benjamins et al., "Knowledge Annotation Initiative of the Knowledge Acquisition Community: (KA)2," in KAW 1998, 11th Knowledge Acquisition for Knowledge Based System Workshop, Banff, Canada, April 18-23, 1998.

[2] C. Bizer, T. Heath, and T., Berners-Lee, "Linked data - the story so far," International Journal on Semantic Web and Information Systems, 5, vol. 3, pp. 1-22, 2010.

[3] P. Casanovas, N. Casellas, C. Tempich, D. Vrandecic, and R. Benjamins, "Opjk and diligent: ontology modeling in a distributed environment," Artificial Intelligence Law, 15, vol. 2, pp. 171-186, 2007.

[4] M. Chein, and M.-L. Mugnier, "Positive nested conceptual graphs," in ICCS 1997, LNAI 1257, pp. 95-109, 1997.

[5] R. Djedidi, and A. Aufaur, "Define Hybrid Class Resolving Disjointness due to Subsumption," http://ontologydesignpatterns.org/wiki/Submissions:Define_Hybrid_Class_Resolving_Disjointness_due_to_Subsumption

[6] G.R. Dromey, "Scaleable Formalization of Imperfect Knowledge," in AWCVS-2006, 1st Asian Working Conference on Verified Software, Macao SAR, China, October 29-31 2006.

[7] J. Euzenat, "Corporate memory through cooperative creation of knowledge bases and hyper-documents," in KAW 1996, pp. (36)1-18, 1996.

[8] J. Euzenat, O. Mbanefo, and A. Sharma, "Sharing resources through ontology alignment in a semantic peer-to-peer system,"in "Cases on semantic interoperability for information systems integration: practice and applications", pp. 107-126, 2009.

[9] M.R. Genesereth, "Knowledge Interchange Format," draft proposed American National Standard (dpANS), NCITS.T2/98-004, http://logic.stanford.edu/kif/dpans.html

[10] W.D. Hillis, "Aristotle (the knowledge web)," Edge Foundation, Inc., 138, May 2004.

[11] J. Lee, J. Park, M. Park, C. Chung, and J. Min, "An intelligent query processing for distributed ontologies," Systems and Software, 83, vol. 1, pp. 85-95, January 2010.

[12] H. Lewen, K.S. Supekar, N.F. Noy, and M.A. Musen, "Topic-specific trust and open rating systems: An approach for ontology evaluation," in EON 2006 at WWW 2006.

[13] D. Lowe, "Co-operative structuring of information: The representation of reasoning and debate," International Journal of Man-Machine Studies, 23, pp. 97-111, August 1985.

[14] P. Martin, "Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English," in ICCS 2002, Springer Verlag, LNAI 2393, pp. 77-91.

[15] P. Martin, "Correction and Extension of WordNet 1.7," in ICCS 2003, Springer Verlag, LNAI 2746, pp. 160-173.

[16] P. Martin, and M. Eboueya, "For the ultimate accessibility and re-usability," Chapter 29 of the Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, IGI Global, pp. 589-606, 2008.

[17] P. Martin, "Managing Knowledge to Enhance Learning," International Journal of Knowledge Management & E-Learning (ISSN 2073-7904), Vol.1, No.2, 2009, pp. 103-119.

[18] P. Martin, "Towards a collaboratively-built knowledge base of&for scalable knowledge sharing and retrieval," HDR thesis, University of La Réunion, France, December 8, 2009.

[19] N.F. Noy, and T. Tudorache, "Collaborative ontology development on the (semantic) web," in SWKE 2008, AAAI Spring Symposium on Semantic Web and Knowledge Engineering, March 2008.

[20] OWL specification of the Semantic Web Topics Ontology, 2006.*lsdis.cs.uga.edu/library/resources/ontologies/swtopics.owl*

[21] R. Palma, P. Haase, Y. Wang, and M. d'Aquin, "Propagation models and strategies," Deliverable 1.3.1 of NeOn - Lifecycle Support for Networked Ontologies; NEON EU-IST-2005-027595, January 2008.

[22] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," IEEE Intelligent Systems, 21, vol. 3, pp. 96-101, May/June 2006.

[23] J. Sowa, "Theories, models, reasoning, language, and truth," http://www.jfsowa.com/logic/theories.htm, December 2005.