

Ontology Repositories with Only One Large Shared Cooperatively-built and Evaluated Ontology

Philippe A. Martin

ESIROI STIM, University of La Réunion, France

and adjunct researcher of the School of I.C.T. at Griffith Uni., Australia

Abstract. This article first list reasons why an ontology repository - or, more generally a knowledge base (KB) server - should permit the collaborative building of one well organized KB rather than solely be a repository for heterogeneous KBs. To that end, the article proposes KB editing protocol that keeps the KB free of automatically/manually detected inconsistencies - and lead knowledge providers to semantically organize their terms and statements - while not forcing them to discuss or agree on terminology and beliefs nor requiring a selection committee. Then, the article gives ideas on how to extend this support to allow a precision-oriented collaborative evaluation of each information provider and piece of information.

Keywords: knowledge sharing/integration/retrieval/evaluation

1 Introduction

Ontology repositories are often only imagined as being collections of static formal files (e.g., RDF documents) more or less independently developed, hence loosely interconnected and mutually partially redundant or inconsistent. Section 2 shows that this "static file based approach" - as opposed to a "collaboratively-built well-organized large knowledge base (cbwoKB) server approach" - makes knowledge sharing/re-use tasks complex to support and do correctly or efficiently, especially in a collaborative way. Most Semantic Web related research works are intended to support such tasks (ontology creation, retrieval, comparison and merging) and hence are useful. However, most often, they lead people to create new formal files - thus contributing to the problems of knowledge re-use - instead of inserting their knowledge into a cbwoKB server. Indeed, it seems that WebKB-2 [10] (webkb.org) is the only ontology server that has protocols supporting a governance-free loss-less well-organized knowledge sharing - and a large general ontology - and hence has at least two of the elements necessary to build a cbwoKB. Other "shared ontology" servers/editors i) let any authorized users make any change in the shared ontology (this discourages information entering or leads to edit wars), or ii) rely on each user or some privileged users to accept or reject changes made in the shared ontology (this is bothersome for the evaluators, sometimes forces them to make arbitrary selections, and is a bottleneck of information sharing that can cause strong delays or discourage information providers).

Section 3 gives protocols - with many yet unpublished ideas - to avoid these governance problems and thus support a scalable collaborative building of a cbwoKB, i.e., a KB where detected *partial* redundancies or inconsistencies are prevented or made explicit via relations of specialization, identity and/or correction; thus, in a cbwoKB, each object has one "right place" in the specialization hierarchy and is then easily retrievable and comparable to the other objects. Section 4 gives ideas on how this support can be extended to allow collaborative knowledge evaluation.

2 Approaches based on Files Versus cwoKB Servers

With files, information retrieval (IR) often leads to a list of possibly relevant files or *pieces of information* (*objects*, e.g., a formal term or an informal sentence) whereas it leads to an exact answer in a cbwoKB or with the content of one formal file. Such an answer may be a portion of the cbwoKB, e.g., a part/subtask/specialization hierarchy (with associated argumentation structures) if the query is of the kind "what are the resources/tools/methods to do ...". Such semantically structured answers permit to find and compare all relevant objects instead of getting a long redundant list of objects/files where original/precise ones are hidden among/behind objects that are more general, mainstream or from big organizations. This is also why IR quality decreases when the size and number of the files increases, but not when the number of objects increases in a cbwoKB.

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. (Some of these criteria may be used for the internal organization of the cbwoKB but the resulting "views" or "contexts" are language/representation dependent choices and, unlike (semi-)independently created static files, do not lead the users to set only few relations between objects of different views). Ontology libraries, from the first ones such as the Ontolingua library to imagined ones such as "The Lattice of Theories" [15], 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 [15], if the objects are organized into a cbwoKB, such (lattices of) theories can be generated via queries.

With files, change management requires version management (which leads to more files and many information management complications); not within a cbwoKB, as Section 3 shows.

With formal files as inputs and outputs, knowledge re-use or integration approaches 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 [5] and for rejecting or integrating changes made in other ontologies, e.g., [2][12][13]. In a cbwoKB, no adaptation or integration has to be done for each re-use: the most important/defining relations from an object to other ones have to be entered by its creators and then they can be incrementally complemented or corrected by any user. Indeed, it is often the case that only the object authors know what their objects really mean or have some other kinds of information required for relating their objects to other ones.

Only a cbwoKB has one "right place" for each new object and hence only a KB server with a large cbwoKB can permit a knowledge provider to simply/directly add one new object "at its right place" and guide her to provide precise and re-usable objects that complement the already stored objects. The protocols of the next two sections work only with a cbwoKB.

3 Collaborative Editing of a KB

The next points describe the principles behind the editing protocols implemented in WebKB-2 for it to be a cbwoKB server, and make some comparisons with features of RDF (which only supports a personal-file based approach). WebKB-2 allows the use of several knowledge representation languages (KRLs): RDF/XML (an XML format for knowledge using the RDF model), KIF and other ones which are here collectively called KRLX and that were specially designed to ease knowledge sharing: they are expressive, intuitive and normalizing (i.e., they guide users to represent things in ways that are automatically comparable). One of them is named Formalized English (FE). It will be used for the examples.

1. In WebKB-2, every object is a term or a statement (generally, a relation between two quantified terms or some relations within a same *context*, i.e., *meta-statement*). A term refers to a concept/relation type or an individual (a type instance that cannot have instances). A statement is an individual and is either informal, formal or semi-formal (when it uses a formal syntax and some terms/objects that are informal or referring to informal/semi-formal objects). A (semi-)formal term is a unique *identifier* for a (semi-)formal object. An informal term is a *name* for an object. Different objects may have common names, not common identifiers.

Every (semi-)formal object has an associated source: creator or source file. The exact unique meaning of a (semi-)formal object may be left implicit and hence might be known only by its creator. Informal objects may also have an associated creator: their meanings are those that their source has implicitly given them. These distinctions permits to differentiate (in-)formal objects and create one specialization/generalization hierarchy categorizing all objects. More precisely, this is an "extended specialization/generalization" hierarchy since in WebKB-2 the

classic "generalization" relation between formal objects (the logical implication) has been extended to apply to informal objects too.

In KRLX, informal objects are double quoted, and object identifiers are either URIs or include their source identifiers as prefixes or suffixes. This is a common solution to avoid lexical conflicts. As in RDF/XML, shortcuts for a source may be used, e.g., `wn#bird` refers to one of the WordNet categories for the English word "bird". The informal statement `"birds fly"_[u1]` was created by the user `u1`. A slight difference with RDF/XML is that the lexical declaration of a shortcut is also a semantic declaration of a term for the source, thus encouraging the creator of the declaration to specify what the source is (a person, a file, etc.); this is also possible in RDF but is not mandatory.

The real difference is that RDF offers no standard way to associate creators to objects and hence, for example, has no notion of "belief". In WebKB-2, each object is, in a sense, contextualized by its source; for example, a statement that is not a definition is a belief and a statement by a user `U` on another user's statement `S` is actually `U`'s belief on his interpretation of `S`.

Similarly, in KRLX, a user has an easy way to i) represent his belief that certain statements belong to a certain source, or ii) associate a private key to its user identifier for preventing another user to impersonate him, and iii) use an encrypted form of this key (i.e., the related public key) for identifying himself.

A knowledge sharing supporting syntax should offer normalized ways to allow this so that knowledge sharing tools can support reasoning or collaboration based on the knowledge sources. RDF and RDF/XML does not yet offer a standard way to allow this. This will come: SPARQL and N3 already offer a way to specify that a statement belongs to a source.

2. *Any user can add any object and use it in any statement (as in RDF) but an object may only be modified or removed by its creator.* This last part has no equivalent in RDF since it is a knowledge model, not a collaboration model.
3. *Each statement has an associated source `S`, and hence, if it is not a definition of a term created by `S`, is considered as a belief of `S`.* When the creator of an object is not explicitly specified, WebKB-2 exploits its "default creator" related rules and variables to find this creator during the parsing. Similarly, unless already explicitly specified by the creator, WebKB-2 uses the "parsing date" for the creation date of a new object. Unless already specified, the creator of a belief is encouraged to add restrictive contextualizing relations on it (at least temporal and spatial relations must be specified).

A definition of a term `T` by the creator `C` of `T` may be said to be "neither true nor false" or "always true by definition": a definition may be changed by its creator but then the meaning of the defined term is changed rather than corrected. No one (including `C`) is allowed to state something about `T` that is inconsistent with the definition(s) of `T`. A user `u1`, is perfectly entitled to define `u1#cat` as a subtype of `wn#chair`; there is no inconsistency as long as the ways `u1#cat` is further defined or used respect the constraints associated to `wn#chair`. A definition associated to `T` by a source `S` that is not `C` is actually a belief of `S` about the

meaning of T. At parsing time, WebKB-2 rejects such a belief if it is found (logically) inconsistent with a definition of T by S.

Universally quantified statements are *not* definitions. RDF+OWL does not make the distinction but N3 does. In WebKB-2, it leads to very different conflict resolution strategies (conflict between two statements of different sources).

- A conflict that involves two definitions by two sources S1 and S2 is a misinterpretation by one of the sources, say S2, of the meaning of a term S1#T created by the other source, and hence is solved by *automatic term cloning* of S1#T, i.e., by creating S2#T with the same definitions except for one and then replacing S1#T by S2#T in the statements of S2. The difficulty is to automatically guess a relevant candidate for S1#T and a relevant definition to remove for the overall change to be minimal. In Annex 2 of [11] I gave some algorithms to do so in common cases.
- Otherwise, a *loss-less correction* is used (details in Point 6).

4. *If adding, modifying or removing a **statement** introduces an **implicit redundancy** (detected by the system) in the shared KB, or if this introduces an **inconsistency between statements believed by the user** having done this action, this **action is rejected**.* Thus, in 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 in case of total redundancy and an extended specialization relation (e.g., an "example" relation) in case of partial redundancy.

In WebKB-2, a statement is seen as a graph with an interpretation in 1st-order logic and graph matching is used for detecting if one graph (Y) is an extended specialization of the other (X), i.e., if X structurally matches a part of Y and if each of the terms in this part is identical or an extended specialization of its counterpart term in X. For example, WebKB-2 can detect that the FE sentence ``Tweety can be agent of a flight with duration at least 2.5 hours'_[u2]` (which means "u2 believes that Tweety can fly for at least 2.5 hours") is an extended specialization (and an "extended instantiation") of both ``every bird can be agent of a flight'_[u1]` and ``2 bird can be agent of a flight'_[u1]`. Furthermore, these last two statements are respectively extended specializations of ``75% of bird can be agent of a flight'_[u2]` and ``at least 1 bird can be agent of a flight'_[u2]`. (Similarly, this last graph can be found to be exclusive with ``no bird can be agent of a flight'_[u3]`).

Except for the fact that it takes into account numerical quantifiers and measures instead of just the existential and universal quantifiers, the graph matching for detecting an extended specialization is similar to the classic graph matching for a specialization (or conversely, a generalization which is a logical deduction) between positive conjunctive existential formulas (with or without an associated positive context, i.e., a meta-statement that does not restrict its truth domain). This last operation is sound and complete with respect to 1st-order logic and can be

computed in a polynomial way if Y has no cycle [3]. Outside this restricted case, graph matching for detecting an extended specialization is not always sound and complete. However, results of searches for extended specializations of a query graph always seem "relevant".

In WebKB-2, no rule based system or theorem prover is used yet. The violation of relation signatures or exclusion relations is detected. The detection of implicit extended specializations between two objects also reveals an inconsistency or a total/partial redundancy, and it is often not needed to distinguish between these two cases to reject the newly entered object. Extended instantiations are exceptions: since adding an instantiation is giving an example for a more general statement, it does not reveal a redundancy or inconsistency (here, an inconsistent belief or incorrect interpretation of a term) 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 (from 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.

Adding, modifying or removing a term is done by adding, modifying or removing at least one statement (generally, one relation) that uses this term. *A new term can only be added by specializing another term* (e.g., via a definition), except for process types which for convenience purposes can also be added via subprocess/superprocess relations. *A new statement is automatically added by WebKB-2 into the extended specialization hierarchy* via graph matching or, for informal statements, solely based on the extended specialization between the words they include). An automatic categorization may be "corrected in a loss-less way" by any user. A new informal statements must also be connected via an argumentation relation to an already stored statement. To sum up, all objects are manually or automatically inserted in the extended specialization hierarchy and/or the subprocess hierarchy, and hence are easy to search and compare.

5. *If adding, modifying or removing (a statement defining) a term T introduces an inconsistency involving statements created or believed by other users* (i.e., users different from the one having done this action), *T is automatically cloned* to ensure that its interpretation by these other users is still represented. In case of term removal, term cloning simply means changing the creator's identifier in this term by the identifier of one the other users (if this generated term already exists, some suffix can be added). In a cbwoKB server, since statements points to the terms they use, changing an identifier does not require changing the statements. In a global virtual cbwoKB, identifier changes in one server need to be replicated to other servers using this identifier.

In a cbwoKB, it is not true that beliefs and formal terms (or their definitions, as well as what they refer to, e.g., concepts) "have to be updated sooner or later". Indeed, in a cbwoKB, every belief must be contextualized in space and time, as in '75% of bird can be agent of a flight' in place France and in period 2005 to 2006'[_[u3], even though such

contexts are not shown in the other examples of this article. If needed, to this belief u3 can associate the term u3#75%-of-birds-fly--in-France-from-2005-to-2006. 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 wordnet1.7#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 certainly 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.

6. *If adding, modifying or removing a belief introduces an implicit inconsistency involving beliefs created by other creators, it is rejected.* However, a user may "loss-less correct" a belief (that he does not believe in) by connecting it to a belief (that he believes in) via a *corrective relation*. E.g., here are FE statements by u2 that correct a statement made earlier by u1:

```
` `every bird is agent of a flight'_[u1] has for
  corrective-restriction `most healthy flying-bird are
  able to be agent of a flight' '_[u2] and
` `every bird can be agent of a flight'_[u1] has for
  corrective-generalization `75% of bird can be agent
  of a flight' '_[u2].
```

If instead of the *belief* `every bird can be agent of a flight', 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:

- u1 originally created this type (u1#bird); then, u2's attempt to correct the definition is rejected, or
- u1 added a definition to another source's type - say, wn#bird since this type from WordNet has no associated constraint preventing the adding of such a definition - and hence i) the types u1#bird and u2#bird are automatically created as clones (and subtypes of) wn#bird, ii) the definition of u1 is automatically changed into `any u1#bird is agent of a flight'_[u1], and iii) the belief of u2 is automatically changed into `75% of u2#bird can be agent of a flight'_[u2].

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, they are encouraged not to use an objection relation but a "corrective relation with argument relations on them". Thus, not only the objections are stated but a correction is given and may be agreed to 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 to better organize 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. Technically, this also removes the 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 may (and should) remain consistent.

For problem-solving purposes, i.e., for an application, choices between contradictory beliefs must 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 point). 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. To conclude, this approach is unrelated to defeasible logics and avoids the problems associated to classic "version management" (furthermore, as above explained, in a cbwoKB, neither formal terms nor statements 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 as 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 likes flowers"_[u2] and "u2 is writing this sentence"_[u2]. Instead of trying to formalize this into exceptions, the editing protocols of WebKB-2 rely on the reluctance of people to argue against such beliefs that should not be argued against.

7. Like all descriptions of techniques, statement/creator evaluation techniques are considered as term definitions and are automatically organized into the extended specialization hierarchy. To support more knowledge filtering or decision making possibilities and incite the users to be careful and precise in their contributions, a cbwoKB server must 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. Details are given in the next section. These measures should not be hard-coded but explicitly represented (and hence be executable by the cbwoKB) to let each user specialize them for its goals and preferences. Indeed, only the user can find 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. Furthermore, before browsing or querying the cbwoKB, a user should be given the possibilities 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 like automatically executed queries on the results of queries. In WebKB-2, like conceptual querying, filtering is based on a search for extended specializations. Filters are useful when the user

is overwhelmed by the amount of information in an insufficiently organized part of the KB.

8. The approach described in the above points is incremental and *works on semi-formal KBs* since the users can set corrective or specialization relations between objects even when WebKB-2 cannot detect an inconsistency or redundancy. As above noted, a new informal statement must be connected via an argumentation relation (e.g., a corrective 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 one and permits to do it only when needed. This is necessary for the acceptance of the approach by people.

These editing protocols encourage or enforce at least a minimal re-use, precision or connectivity between objects: each object is connected to at least another object via relations of specialization/generalization, identity and/or argumentation. They permit a loss-less information integration approach since no knowledge selection has to be made. They can be seen as enabling a precise asynchronous dialogue between the knowledge providers. To sum up, they permit, enforce or encourage people to interconnect their knowledge into a shared KB, while keeping the KB consistent but without having to discuss and agree on terminology or beliefs.

Since the techniques described in this article work on semi-formal KBs and are not particularly difficult for information technology amateurs - since the minimum these techniques require is for the users to set the above mentioned relations from/to each term or statement - they can 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 listed in the previous section, without creating new problems. Its use would permit to merge (the information discussed or provided by the members of) many communities with similar interests, e.g., the numerous different communities working on the Semantic Web. From an application viewpoint, the approach seems interesting for collaboratively-built states of the art, corporate memories, catalogues, e-learning, e-government, e-science, research, etc.

The hypothesis of this approach 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 maintain 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 or mappings of ontologies that I made invalidated these hypothesis.

4 Evaluating Objects and 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 [8] 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, these 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. As previously noted, a cbwoKB should provide "default global measures" for the evaluation of each statement/creator (based on each user's individual evaluations) and allow the users to refine it. Here are comments (only general ones due to space restrictions) on the global measures that are at the implementation stage in WebKB-2.

- 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 [9]. The KB server Co4 [4] 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.
- 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).
- A global measure of *the usefulness of a statement* should exploit (at least) the above two measures.

- 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. However 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.) For example, 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.

[6] 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). [6] did not give indications on such mechanisms but those proposed in this article seem one initial basis for them.

6 Conclusion

This article aimed to show that a cbwoKB - and hence a cbwoKB based ontology repository - is technically and socially possible, and - on the long term or when creating a new KB for *general* knowledge sharing purposes - provides more possibilities, with *on the whole* no more costs, than the mainstream approach [14][1] where knowledge creation and re-use involves searching, merging and creating (semi-) independent (relatively small) ontologies. However, research results on knowledge extraction or merging may ease the creation of a cbwoKB. Conversely, the results of applying these techniques with a cbwoKB as one of the inputs would be better and they would not be lost if stored in a cbwoKB.

This article showed that a cbwoKB can be collaboratively built and evaluated without a selection committee and without forcing the users to discuss or agree on terminology and beliefs. However, to guide users into collaboratively representing knowledge in a normalized and organized way and hence inserting it "at the right places", other elements are also needed: expressive and normalizing notations, methodological guidance, a large general ontology, and an initial cbwoKB core for the application domain of the intended cbwoKB. WebKB-2 proposes research results for all these elements, with for application domain the "Semantic Web related techniques".

7 References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5, vol. 3, pp. 1-22 (2010)
2. Casanovas, P., Casellas, N., Tempich, C., Vrandecic, D., Benjamins, R.: Opik and diligent: ontology modeling in a distributed environment. *Artificial Intelligence Law*, 15, vol. 2, pp. 171-186 (2007)
3. Chein, M., Mugnier, M.-L.: Positive nested conceptual graphs. In: *ICCS 1997, LNAI 1257*, pp. 95-109 (1997)
4. Euzenat, J.: Corporate memory through cooperative creation of knowledge bases and hyperdocuments. In *KAW 1996*, pp. (36)1-18 (1996)
5. Euzenat, J., Mbanefo, O., Sharma, A.: 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)
6. Hillis, W.D.: *Aristotle (the knowledge web)*, Edge Foundation, Inc., 138 (2004)
7. Lee, J., Park, J., Park, M., Chung, C., Min, J.: An intelligent query processing for distributed ontologies. *Systems and Software*, 83, vol. 1, pp. 85-95 (2010)
8. Lewen, H., Supekar, K.S., Noy, N.F., Musen, M.A.: Topic-specific trust and open rating systems: An approach for ontology evaluation. In *EON 2006 at WWW 2006*.
9. Lowe, D.: Co-operative structuring of information: The representation of reasoning and debate. *International Journal of Man-Machine Studies*, 23, pp. 97-111 (1985)
10. Martin, Ph., Eboueya, M.: For the ultimate accessibility and re-usability. Chapter 29 (14 pages) of the *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies*, IGI Global (2008)
11. Martin Ph.: *Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations*. Ph.D. thesis (378 pages), University of Nice - Sophia Antipolis, France (1996)
12. Noy, N.F., Tudorache, T.: Collaborative ontology development on the (semantic) web. In: *AAAI Spring Symposium on Semantic Web and Knowledge Engineering (SWKE 2008)*
13. Palma, R., Haase, P., Wang, Y., d'Aquin, M.: Propagation models and strategies. Deliverable 1.3.1 of *NeOn - Lifecycle Support for Networked Ontologies*; NEON EU-IST-2005-027595 (2008)
14. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. *IEEE Intelligent Systems*, 21, vol. 3, pp. 96-101 (2006)
15. Sowa, J.: Theories, models, reasoning, language, and truth. <http://www.jfsowa.com/logic/theories.htm> (2005)