# How WebKB could contribute to PORT

Philippe Martin

Distributed System Technology Centre, Australia, phmartin@webkb.org

## 1 Introduction

PORT is a project aiming to permit the electronic storage of C.S. Peirce's writings, their conceptual annotation and organization by a restricted number of scholars, and their access, querying and navigation by Web users.

Some reasons why this undertaking is a challenge are: (i) a number of the documents are hand-written, include drawings, notes in the margins and corrections, (ii) there may be several versions, and pages may be unordered, (iii) each sentence, paragraph or page may countain some ideas about a number of subjects, (iv) the sentences or ideas may be difficult to interpret or understand, and the ideas may be explained in various different ways which may not appear consistent with each other, (v) Peirce's ideas developed during his life-time.

This article does not address the issues related to the selection of a good digital format for this work, e.g. the possibility to combine or hyperlink raw text and images, and the possibility to select or zoom parts of an image. We only assume that each document, page or image can be accessed and referred to via a URI. This is for example the case of some of Peirce documents in R.S. Robin's annotated catalog[1].

We show how a knowledge-based system such as WebKB (www.webkb.org) could be used as a support for the collaborative conceptual annotation and organization of the documents, or more precisely, of the document elements (e.g. word, sentence, part of image, section) of interest to the users. WebKB permits any Web user to contribute to this work and permit them to collaborate without having to agree on semantic or lexical issues. A user may navigate and query the knowledge of all users or selected (kinds of) users.

We first compare a few approaches for conceptual organization and retrieval, and list some features of WebKB. Then, we give examples, highlight some problems with our knowledge-intensive approach, and propose a minimal ontology for the task.

## 2 More or less knowledge-intensive approaches

Automatic keyword-based indexation (e.g. as in Altavista) or manual annotations of document elements (DEs) with raw text (e.g. as in Amaya[2]) do not permit conceptual organization and retrieval.

Manual indexation, markup or hyperlinking with a predefined set of topics, categories or rhetorical/argumentation relations is restrictive and insufficient. Markup languages also impose the modification of the annotated documents, which makes editions by multiple users difficult to handle (e.g. see the platform PIÑAS[3]).

Manual indexation or hyperlinking of the DEs, or of certain words in the annotations of the DEs, with concept/relation *types* in an ontology that can be extended by the users, may be viewed by some persons as an interesting compromise between ease of annotation and knowledge precision (and hence knowledge exploitation possibilities). Informal knowledge representation models such as Topic Maps[4] may be sufficient in this approach.

The use of formal or semi-formal *statements* (instead of types) to represent some of the content of *DEs or other statements*, connect them to other *DEs or statements*, and make judgements or hypothesis about them, is the most precise, flexible and exploitable-for-inferencing approach. It is also the most difficult and time-consuming for the users. WebKB has various features and commands to help the users produce the representations and then exploit them.

[1] http://www.iupui.edu/~peirce/web/robin/robin.htm
[2] http://www.w3.org/Amaya/
[3] http://delos.imag.fr/~decoucha/PINAS_main.html
[4] http://alexandria.sdc.ucsb.edu/~acoleman/tmaps.html

## 3 WebKB features

We must temporarily distinguish WebKB-1[1] from WebKB-2 [2]. WebKB-1 can be asked to load and interpret Web documents that include conceptual graphs (CGs) assertion or query commands, commands for accessing other Web documents or Web-based servers, DE indexation commands, Unix-style text processing commands (such as grep, awk and diff), and control structures (e.g. pipe, "if", "for", "function") for combining these commands. Scripts permit to solve problems (e.g. see our solution to Sisyphus-1[5]) or generate complex documents. Calls may be associated to HTML hyperlinks. The CGs may be expressed in CGLF, Frame-CG (FCG) or Formalized-English (FE). Some simpler formats may also be used in restricted cases. The commands or scripts are separated from the rest of the document via special delimiters, e.g. the XHTML marks `<KR>` and `</KR>`.

WebKB-2 is a (currently partial) rewriting of WebKB-1 above the OODBMS FastDB[6] and with features to permit the users to cooperatively build a large KB (on the WebKB server machine) and hence permit them to share their knowledge. To maximize knowledge re-use, retrieval and consistency checking, users' knowledge is not stored in separate, loosely interconnected modules or files but tightly integrated, and each element of the KB (category, link between categories, CG) has an associated creator/source (and optionally a creation date). To avoid lexical conflicts, each category identifier is composed of the creator identifier and a key name, e.g. `wn#domestic_dog`. A category may have several names (that may be shared by other categories) which may be included in the identifier (e.g. the previous category from WordNet could also be referred to via `wn#domestic_dog__dog__Canis_familiaris`). To avoid semantic conflicts or redundancies and maximize consistency checking, a user cannot enter a CG that is "comparable" to one already existing in the KB, unless she connects the two CGs with a relation of type `pm#corrective_specialization`, `pm#corrective_generalization` or `pm#correction` (see [2] for details). Knowledge normalization, and hence its matching and retrieval, is also encouraged via lexical, structural and ontological conventions and the use of the proposed high-level notations.

WebKB-2 currently only accepts FCG as input format for CGs and does not have all the DE indexation facilities of WebKB-1, e.g. there is no command to index (and later retrieve) the 2nd occurrence of the string "A cat is on a mat" in a certain Web file. However, thanks to the initialization of the KB with WordNet 1.7 [7] and our top-level ontology, the user does not have to spend a long time adding categories in the ontology whenever she adds a new statement, but most often simply has to retrieve and re-use the adequate categories. This re-use also permits consistency checks and eases knowledge sharing. We hope that ultimately WebKB-2 will also have all the features of WebKB-1. Both of them are accessible and usable from the WebKB site (www.webkb.org).

Like programming, knowledge modelling involves errors and revisions or reorganizations. Hence, the users of WebKB-2 are recommended to store and document their knowledge into one or several Web files (as they would have to do with WebKB-1) rather than directly try to insert it in the KB. When WebKB-2 loads a file that has syntax/semantic errors or includes the command "`no storage;`", the assertions or removals are not committed. When satified with a file, a user can commit it by removing the command "`no storage`". Then, this file can be seen as a backup or a documentation for the commited knowledge.

## 4 Examples

PORT scholars will have to represent DEs, versions, hypothesis and arguments. For clarity purposes, the following examples are not about some documents of Peirce but about the Bible and a paragraph from the Exodus. The represented hypothesis is that the divine dividing of the Red Sea described in the King James Bible may have had for actual source the temporary water receding of a reed sea due to the volcanic eruption of the Santorini at that time.

[5] http://www.webkb.org/kb/sisyphus1.html
[6] http://www.garret.ru/~knizhnik/fastdb.html
[7] http://www.cogsci.princeton.edu/~wn/

This document is an example of file mixing commands and text. Its HTML version (accessible at www.webkb.org/doc/port02.html) can be loaded (or re-loaded) in WebKB-2 via an hyperlink at this position in the file. An argument in the call to WebKB-2 specifies that the loading is done in the name of the user "jd" (John Doe) who has no password yet. This means that he will be the owner of the graphs of the next section.

Categories with identifiers beginning by '#' come from WordNet (since 95% of the 77,900 categories in WebKB-2 are from WordNet 1.7, the prefix "wn" is the default and may be omitted). WebKB-2 also permits the use of category names instead of category identifiers when there is no ambiguity on the referred category (e.g. because the name only refers to one category or because relation signatures can be used to reduce the number of alternatives). Most of the category names used below (see the terms without '#' within) are resolved to basic binary relation types created by "pm". To ease the knowledge representation task, WebKB-2 also permits a certain number of concept types (those that are subtypes of `pm#thing_that_can_be_seen_as_a_relation`) to be used as if they were relation types. Since no relation signature is associated to those types, WebKB-2 only checks that the type used in the destination concept specializes the type used as relation type.

In the following graphs, "a", "an", "the" and "some" are syntactic sugar for the existential quantifier, while "any" is for the universal quantifier. The tree-like structure used in the FCG and FE notations specifies the order and scope of the quantifiers. More details on FCG and its connections to KIF, CGLF, CGIF and FE can be found in [3]. The grammars can be found in http://www.webkb.org/doc/F_languages.html.

```
#Bible > jd#Hebrew_Bible jd#Greek_Bible; //2 subtypes of #Bible added by jd
#Exodus > jd#original_Hebrew_Exodus; //(#Exodus is a subtype of #book.section)
//these 3 subtypes are the only types that need to be declared (separately from
//the graphs) for the examples of this section!

[ [any #King_James_Bible, //(#King_James_Bible is a subtype of #Bible)
    language: an English_language,
    result of: (an interlingual_translation, time: 1611,
                material: {a Hebrew_Bible, a Greek_Bible}),
    version of: {a Hebrew_Bible, a Greek_Bible},
    version: {any American_Standard_Version, any British_Revised_Version}
  ], dc#Source: http://www.mtholyoke.edu/lits/library/guides/biblver.htm
]; //this graph is stated by "jd";  "dc" identifies the Dublin Core
```

The above FCG asserts that, according to the cited source, any King James Bible is written in English, is the result of a translation from an Hebrew bible and a Greek bible, is a version of these bibles, and has at least two versions: the American and British revised versions. It is important to note that all these bibles (and their components) have to be represented as types (not individuals) because they may have subtypes, but that most versions cannot be represented as subtypes (since they do not simply have *more* characteristics but *different* ones).

The next FCG asserts that King James Bibles have the same content (but may have different formats). The order of the quantifiers is important.

```
[a pm#string, ascii_content_except_for_spaces of: any #King_James_Bible];
```

The next three FCGs define the paragraph we are interested in. The fourth uses the URL of a DE as an identifier.

```
[type jd#KJB_Exodus (?x) := [an Exodus ?x, part of: a King_James_Bible] ];

[type jd#KJB_Exodus-14-21 (*x) := [the 21st #paragraph *x, part of:
                                   (the 14th #chapter, part of: a KJB_Exodus)] ];

[any jd#KJB_Exodus-14-21, ascii_content_except_for_spaces:
    "And Moses stretched out his hand over the sea;
     and the LORD caused the sea to go back by a strong east wind all that night,
     and made the sea dry land, and the waters were divided."];

[the jd#KJB_Exodus-14-21 http://www.bartleby.com/108/02/14.html#21];
```

The following graphs are a very tiny summary of a BBC documentary that I recently saw on TV but for which I could not find an URL (hence, "jd" is said to be the author). The idea was that many descriptions in the Exodus (the seven plagues, the ashes, the water receding due to a strong wind, etc.) were consistent with observed effects of volcanic eruptions, that the Red Sea could actually have been some near-by reed sea (much easier to divide and dry than the Red Sea), that the Santorini eruption was big and co-temporal with the Exodus, that it was only 200 miles NE of the reed sea with no island in-between, and that some pumice stone from this volcano (or very likely to be) was found in this reed sea. Unfortunately, as the few following graphs show, really "representing" the content of the previous sentence would take a much larger number of graphs and is a difficult task that can be done in many ways and the various alternatives may not be "comparable" automatically. This is the problem of the knowledge-intensive approach. (Simply indexing the words would not improve the situation but would be much quicker to do). However, the effort and the number of alternatives are reduced when objects and observations have already been represented (e.g. by other scholars) and can be re-used. In that case, the knowledge-intensive approach is the method of choice to interconnect/organize the various observations and hypothesis.

```
[jd#Red_Sea_could_have_been_translated_reed_sea //identifier given to this graph
 [any (document_element, ascii_content: "the Red Sea", part of: a KJB_Exodus-14-21),
   result of: (several interlingual_translation, object:
                (a group of #word, language: an #Hebrew, may be object of:
                  (an interlingual_translation, result: "a reed sea")))] ];

[an #hypothesis //the following instance of #hypothesis is automatically declared
     jd#the_original_bible_could_refer_to_a_reed_sea_instead_of_the_Red_Sea,
  argument: {jd#Red_Sea_could_have_been_translated_reed_sea}];

[[jd#the_Santorini_volcano_200miles_NE_of_Egypt_erupted_at_the_time_of_the_Exodus
  [a #volcanic_eruption jd#Santorini_eruption,
        object: (a #volcano, location: Mediterranean_Sea),
        time: (a #time, time of: (a pm#situation, descr: a KJB_Exodus-14-21))]
 ], dc#Source: http://bibleandscience.com/dateofexodus.htm];

[a #fact.info jd#a_strong_volcanic_eruption_can_make_water_recede_far_away];

[any KJB_Exodus-14-21, descr of: [jd#Exodus_receding_of_water_by_God
              [a #receding, object: some body_of_water, agent: a God] ] ];

[the #hypothesis
 jd#the_water_receding_of_the_Exodus_may_have_been_caused_by_the_Santorini_eruption
 [ [a #receding, object: some body_of_water],
   #interpretation: jd#Exodus_receding_of_water_by_God,
   may be consequence of: Santorini_eruption] ];

[jd#the_water_receding_of_the_Exodus_may_have_been_caused_by_the_Santorini_eruption,
  argument:
   {jd#the_original_bible_could_refer_to_a_reed_sea_instead_of_the_Red_Sea,
    jd#the_Santorini_volcano_200miles_NE_of_Egypt_erupted_at_the_time_of_the_Exodus,
    jd#a_strong_volcanic_eruption_can_make_water_recede_far_away}];
```

## 5 Some necessary types of concept and relation

### 5.1 Categories related to descriptions

Most modelling tasks, and especially those related to writings and annotations, have to distinguish between (i) a physical support of description (e.g. a paper, a stone, a wall), (ii) an abstract container of description (e.g. a paragraph in a document, an image), (iii) a content of description (e.g. a narration, an hypothesis, a definition), and (iv) a medium of description (e.g. a language, a language unit, a data structure).

Although these distinctions are clearly exclusive, classifying objects (or categories referred by common words such as "book" or "software") according to them can be difficult. There is a need for a category generalizing the last three distinctions and another category for the last two distinctions. For example, WordNet categories related to content or medium of description are often very mixed or difficult to tell apart (hence we did not further classified them during our integration of WordNet in WebKB-2). Furthermore, some relations apply to objects of different kinds, e.g. the Dublin Core relations can be used on any object of the last three distinctions. Such relations reduce the possibility of automatic semantic checking (or of using category names instead of category identifiers) but ease the knowledge representation task. We have recently generalized the signature of some relations (e.g. argumentation relations) to ease the re-use of categories from WordNet and hence make the knowledge representation task more bearable to the users. Here are WebKB-2's uppermost concept/relation types related to descriptions. (The uppermost layer of the whole ontology is given in Subsection 5.3.).

```
pm#description_content/medium/container
 > {pm#description pm#description_container}; //{..}: open subtype partition

  pm#description (^description content/medium of an entity or a situation^)
    > pm#description_content pm#description_medium sowa#form;

    pm#description_content (^e.g. a narration, an hypothesis^)
     > pm#knowledge_representation pm#narration pm#fact_generalization
       sowa#proposition sowa#intention kads#role rdf#description
       #subject_matter; //#subject_matter has 1195 subtypes from WordNet1.7

    pm#description_medium (^e.g. a syntax, a language, a script, a software^)
     > pm#abstract_data_type #communication #language_unit #symbolic_representation;

pm#relation_from_description_content/medium/container
             (pm#description_content/medium/container,*) //"*" is like "..." in C
  > pm#relation_from_description pm#version dc#Coverage dc#Contributor dc#Source
    dc#Publisher dc#Rights dc#Date pm#authoring_time pm#author dc#Language
    dc#Format pm#description_instrument pm#description_object pm#physical_support
    pm#rhetorical_relation pm#argumentation_relation;
    //there are currently 11 types of argumentation relation in WebKB-2

pm#descr (?,pm#description_content/medium/container) (^for connecting any object to
         a formal representation of it, e.g. a representation written with a fcg^);

[any pm#thing, //WebKB-2 combines such "schemas" to generate menus that ease KR
     may have for pm#descr: a pm#description,  //anything may be described
     may have for pm#descr_in: a pm#description_container   //somewhere
](pm); //according to pm, i.e. phmartin@webkb.org
```

### 5.2 Containers of descriptions: document elements

The location of a DE within a document may be formally described (as illustrated above with `jd#KJB_Exodus-14-21`) but parameters at the end of a URL could also be used if PORT members have access to a tool that can understand these parameters and display the referred DE. For example, `http://foo.org/bar.gif#quartNW` could refer to the first quart of the image, while `http://foo.org/bar.gif#l25-27` could refer to the lines 25 to 27 if the image represents some text. Below are some DE related excerpts of the current ontology in WebKB-2.

```
pm#description_container (^file, image, ... but not a disk or a piece of paper^)
  > pm#document_element #representation_container;

    pm#document_element__document (^a part of a document or the whole document^)
      > #document;

document_element > document {image_DE, textual_DE, audio_DE}
  document > {manuscript electronic_document} multi_media_document;

[any pm#description_container,
     may have for part: a pm#description_container,
     pm#physical_support: a pm#physical_entity /*e.g. a paper, a stone*/ ](pm);

pm#file_or_file_element (?,pm#description_container)
  > pm#file pm#file_element pm#descr_container;

pm#descr_in (?,pm#description_container) (^when a thing t has a description stored
       in a description container dc, there is a relation pm#descr_in from t to dc^);
```

### 5.3 Some top-level categories

For a more synthetic view of the ontology, here is another excerpt of it. Please navigate the actual ontology at www.webkb.org for details.

```
pm#thing (^anything that is not a relation^)
  > {(pm#situation pm#entity)};  // {(...)}: closed subtype partition

  pm#situation (^something "occuring" in a real/imaginary region of space/time^)
    > {(pm#state pm#process)} pm#phenomenon pm#situation_playing_some_role;

  [any pm#situation, pm#place: a pm#spatial_entity,
         pm#time: a pm#time_measure,  pm#duration: a pm#time_measure
         pm#later_situation: a pm#situation, pm#sub_situation: a pm#situation,
         may have for pm#agent: a pm#entity,
         may have for pm#experiencer: a pm#causal_entity,
         may have for pm#instrument: a pm#entity,
         may have for pm#object: a pm#thing, may have for pm#result: a pm#thing,
         may have for pm#recipient: a pm#entity](pm);

  pm#entity (^something that can be "involved" in a situation^)
    > {pm#spatial_entity pm#nonspatial_entity} pm#entity_playing_some_role;

    pm#nonspatial_entity (^e.g. knowledge, motivation, language, measure^)
      > pm#collection pm#psychological_entity
        {pm#description_content/medium/container pm#attribute_or_measure};
```

## 6 Conclusion

Tools for "cooperative conceptual annotation and organization of document elements and conceptual annotations themselves" would be useful not only for PORT but many other applications: corporate memories, repositories of concepts, techniques and tools in various domains or applications, Yellow-Pages like catalogues (for unrestricted but organized information of products or services, with comparisons and feedbacks), etc. We have illustrated a knowledge-intensive generic approach, with some of its advantages and drawbacks. (We focused on the knowledge representations; details on the actual interfaces, knowledge retrieval mechanisms, and advantages of the used notations and conventions, can be found in the referred articles).

WebKB-2 is designed to be scalable in knowledge volume and, hopefully, number of users (this last characteristic has not been tested). A unique WebKB-2 server would very probably be sufficient to support all PORT members' conceptual annotations and queries (the KB does not include the annotated document elements). However, it may be that a future easier-to-use and less knowledge-oriented version of WebKB-2 is required.

## References


1. P. Martin and P. Eklund, "Embedding Knowledge in Web Documents," *Proc. of the 8th Int'l World Wide Web Conference* (WWW8), Toronto, Canada (1999). http://www.webkb.org/doc/papers/www8/www8.ps
2. Martin, Ph., Eklund P.: Large-scale cooperatively-built heterogeneous KBs. In Proc. of ICCS 2001, 9th International Conference on Conceptual Structures, Springer Verlag, LNAI 2120, Stanford University, California (2001) 231–244. http://www.webkb.org/doc/papers/iccs01/
3. Martin, Ph.: Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. Proc. of ICCS 2002, 10th International Conference on Conceptual Structures, Borovets, Bulgaria, July 15-19, 2002. http://www.webkb.org/doc/papers/iccs02/