

Collaborative Paper Exchange (CoPE) – a community based application for sharing paper summaries

Independent study report
Semester: Fall 2005
Course: InfSci 3924
School of Information Sciences
University of Pittsburgh

Submitted by: Michael V. Yudelson
Instructor: Peter L. Brusilovsky

Introduction

In many graduate and undergraduate courses, instructors attempt to go “beyond the textbook” by assigning them research papers to read. To evaluate this work, the teachers frequently ask students to prepare brief summaries of the papers read. The summaries are graded to provide motivation for students and to evaluate their level of understanding. At the School of Information Sciences this approach has been used in several graduate courses for many years. Over the years, we have accumulated large stacks of graded student paper summaries that have been archived and left unused.

The challenge that we are addressing in this paper concerns creating a better use of these summaries. Once written, a summary becomes an important secondary resource. Being the result of the hard work of one student, it presents the core issues of a specific paper using the terminology and viewpoint of students taking this specific course. Once shared among students taking this course, these summaries can provide an important connection between the core course content found in a textbook and more recent trends that can only be distilled from research literature. In other words, a good set of organized research paper summaries could help students understand the state-of-the-art in the course’s field. Reading a few selected research papers and writing summaries of them will give a student just a sample of this state-of-the-art. Using the accumulated work of other student’s could accelerate an individual student’s progress.

Unfortunately, traditional E-Learning tools provide no support for the proper sharing of research paper summaries (Brusilovsky & Miller, 2001). Learning management systems such as Blackboard (Blackboard Inc., 2002) could automate the process of submitting a summary or delivering a grade, but they do not support the structured exchange of summaries. Encouraged by a group of faculty at our school, we attempted to develop a system that supports the process of the collaborative exchange of paper summaries. As of this writing, the first version of the system, named CoPE (Collaborative Paper Exchange), has been developed and evaluated in the context of a graduate course on Interactive Systems Design. Results show that the system achieved a range of its goals and provided valuable feedback for further development of the system. This paper presents the system and summarizes the results of the study.

CoPE v.1 Interface

The window of CoPE v.1 consists of three frames: a navigation frame on the left, a content frame on the center and right, and a menu frame on top (Figure 1). The CoPE’s interface follows the ‘folder-document’ paradigm: the content is organized as a directory tree with documents stored in the tree nodes. Users browse papers and summaries by expanding/collapsing nodes of the directory tree and selecting documents.

CoPE provides the following functionality to its users:

- add and edit research paper summaries;
- browse the summaries;
- index them following rubrics from a defined set;
- give feedback (grades) for a summary;
- review feedback for a summary.

There are 3 user types and multiple user groups. The user types are: administrator, instructor, and student. There is only one user group for each of the administrator and instructor types. There can be multiple student groups. Depending on their type, users have different access rights to the system.

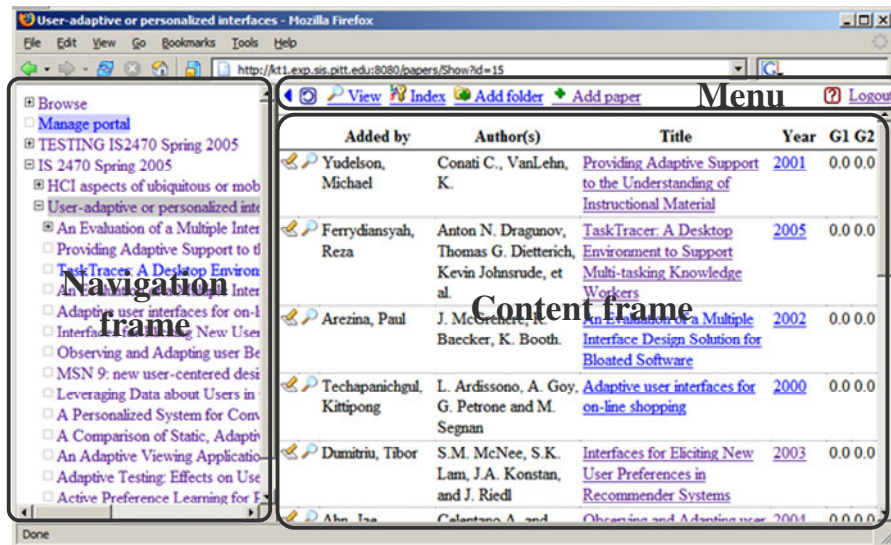


Figure 1 CoPE interface: Navigation tree on the left, contents of the folder on the right, and menu on the top.

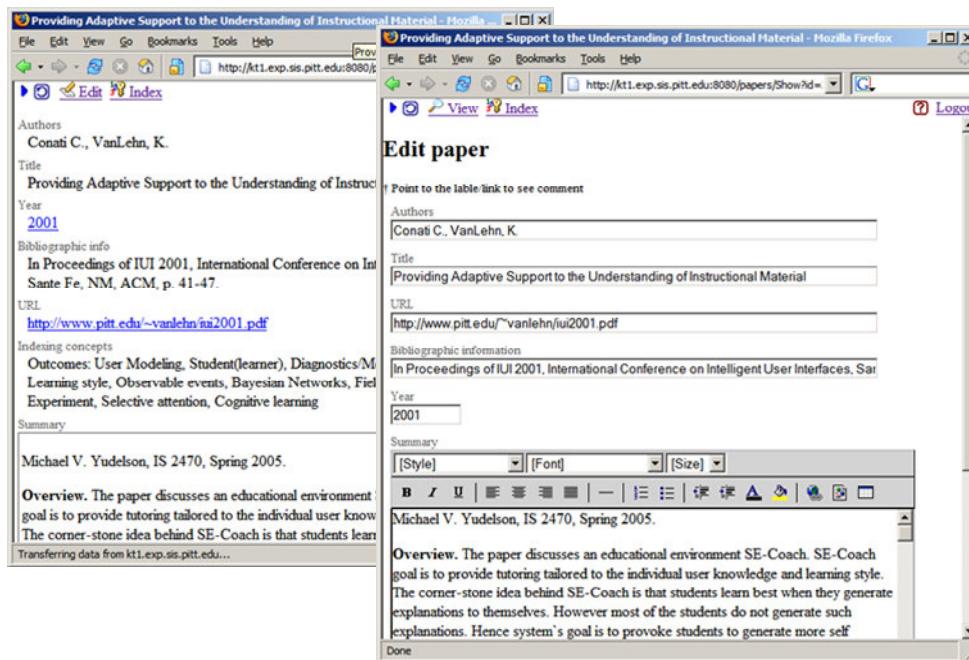


Figure 2 View (left) and add/edit paper (right) in CoPE.

Users of type administrator have privileges to do any possible task. They can modify and view all of the information. In addition to the rights listed above, they can create folders, assign folders to instructors, and modify folder properties (name, submission quota). Administrators can also create new users and user groups, and assign student groups to

instructors. Instructors have an intermediate access level. They can give grades for student paper summaries that are submitted to the folders assigned to them and also create subfolders and modify their properties. However, instructors cannot modify the body of the student's submission, only the grade. Students can edit submissions of their own and view grades that were given for those submissions.

All users can navigate through papers using the explorer-style left frame (Figure 1). By default, each paper summary is opened in view mode. Users can switch to edit mode if they submitted the paper or are an administrator of the CoPE system. To switch to edit mode one has to click on the 'paper-and-pencil' icon next to the paper title in the list of papers or by using the top menu. In addition to navigation using the folder hierarchy, papers can also be accessed by year of publication. This can be done in the folder list view (Figure 1 content frame), from a paper document view (Figure 2 left), or in a separate folder view: Browse→By Year→<year>. Navigation based on the author's name and concepts-keywords (see Section 3 for details) associated with the paper are future features we plan to implement as soon as possible.

When users add a new paper summary, they provide basic bibliographic information about the paper (such as title, list of authors, list of publication), plus they submit a URL of the paper (if such exists) so that it can be retrieved and reviewed (Figure 2). Paper summaries can either be entered using the supplied rich text editor or pasted from an MS Word or RTF file. Since the summary will be stored in HTML format, it can also be edited/ entered directly in HTML format.

3. Indexing Papers in CoPE v.1

Indexing is one of the principal features of CoPE. In an academic setting, it is used in conjunction with the summary to demonstrate how well the student understands the gist of the paper, since indexing shows whether s/he can correctly identify the relationship of the paper to the chosen domain. According to the constructivism paradigm, learning is considered to be the process of actively building the concept structure of the domain. When a student reads a paper it is important for her/him not only to verbalize the summary of the paper, but also to define the main topics the paper is about. A paper is considered a source, from which a student extracts concepts describing knowledge being acquired. By asking students to index papers we motivate them to actively explore the set of available concepts, focus on specific ones and build the connections between concepts and the content of the paper.

The pilot version of CoPE implemented traditional list-based indexing (Figure 3 left). This interface required users to select concepts relevant to the paper from an ordered list of available concepts and place them into a list of prerequisites – concepts necessary to be known to a person prior to reading the paper or a list of outcomes – concepts that the paper is defining, extending or actively using in some other way. Preliminary user evaluation has shown that list interface is too simplistic and a more elaborate interface is necessary.

An alternative to the flat-list concept organization is a hierarchy of concepts, where students can navigate in a more natural way using simple mouse clicking and dragging. Hierarchical structures represent an intuitive basis for creation of navigable hypermedia interfaces. The first step in this direction was done in 1995 by Gaines and Shaw when

they created the WebMap – a system which integrates concept maps with websites (Gaines and Shaw, 1995). We have been exploring the possibility of using hierarchies along with the flat lists of concepts in order to index intelligent content (Brusilovsky et al, 2005). In the case of CoPE, we want to find out which of these two interfaces is more desirable for users and more efficient for indexing; therefore we use them separately. To represent the domain hierarchy we have developed the HCI ontology (see Figure 3 right).

The design principles guiding manual ontology development depend very much on the intended goal of this ontology and the role it will play in the information system. Taking into account the intention to use the developed ontology not as a knowledge base component, but rather as a cognitive tool for manual information search and navigation, our designs were influenced by the principle of "good shape" (Gavrilova et al., 1999). We tried, where possible to preserve the balance of ontology and to avoid large branching factors of the links, cross-links, and relations of multiple types. The set of leave-concepts of the ontology is based mainly on several HCI glossaries available on-line (<http://www.meanswhat.com/weblog/glossary.html>; <http://www.usabilitysa.co.za/hcigloss.htm>; <http://www.usabilitysa.co.za/hcigloss.htm>). The meta-concepts of these higher levels reflect our view on the structure of the domain and its main areas of knowledge.

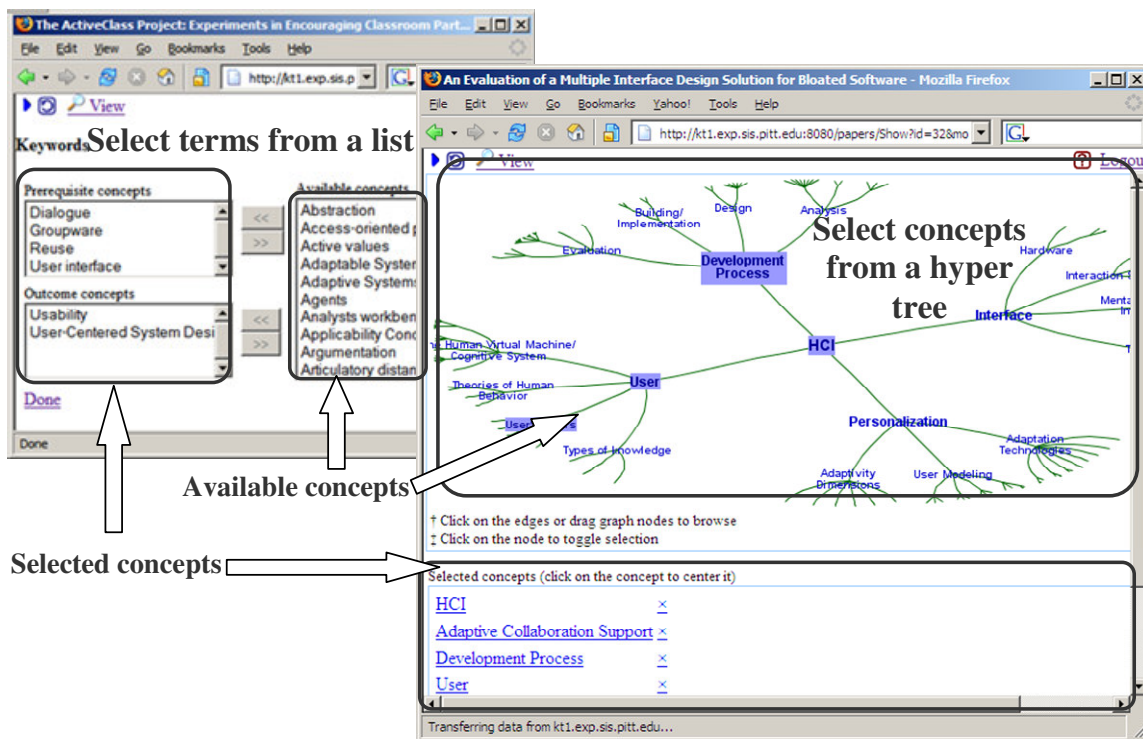


Figure 3 Indexing in CoPE. Left – list-based interface (abandoned), right – hierarchy-based interface

For ontology visualization we use the hypergraph software (HyperGraph, 2005), which provides an open source, easily-tunable platform for manipulating hyperbolic trees (Lamping et al, 1995). A number of research and practical projects are conducted currently on developing different types of tools for the visualization of large concept structures (Uther and Kay, 2003; Ziegler et al. 2002). Hyperbolic trees allow one to shift

the focus away from unnecessary information while preserving the entire structure of the tree (or its sufficient part) on the screen. Since, our choice of ontology type is a simple hierarchy; tree structure is the best choice for representing the relationships of the domain concepts and organizing them into helpful navigational components.

To browse the hierarchy, users have to drag the nodes or click on the edges between the nodes. Clicking on the nodes toggles the selection of the underlying concepts. To simplify the search for the terms already selected users can click on the concept name in the 'selected concepts' area below the tree. That will center the tree on that term. To remove a selected concept, the user has to click on the 'X' next to the selected concept.

4. CoPE v.1 Evaluation

A classroom evaluation of CoPE was performed during the Spring 2005 semester in an Interactive Systems Design class. There were 20 students registered for this course. All of the students had to submit 2 paper summaries and indexing for the papers as class assignments. The topic of the first paper was required to address research related to ubiquitous and mobile computing. The second paper was required to address research related to human computer interaction. The first paper had to be indexed using the list interface. The concepts glossary for the list interface was borrowed from an online dictionaries (HCI Glossary, 2000), by Hugo. Students had access to these dictionaries and could retrieve corresponding concept definitions. The second paper had to be indexed using the hierarchy interface. The concept ontology for it was developed by the authors of this paper. Students did not have access to concept definitions.

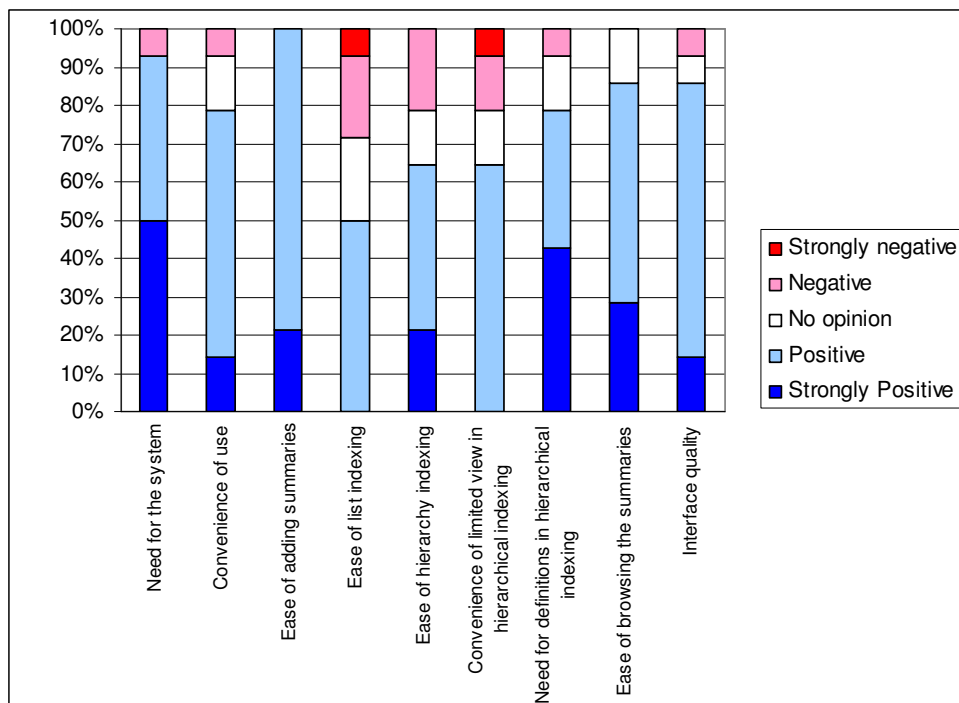


Figure 4. Subjective student evaluation of different features of CoPE

At the end of the semester, subjective student feedback was collected in the form of answers to a non-mandatory questionnaire. Out of 20 students 14 submitted their

questionnaires. Most questions in the questionnaire were liker-style rating questions. In addition, students had the opportunity to express their opinions in free form. Each question contained space for “additional comments” and at the end of the questionnaire, a section for “additional comments” regarding the system in general was also included. The questionnaire was composed of 11 questions. Part of the data from submitted questionnaires is shown in Figure 4. As Figure 4 shows, the users believe there is a need for such a system (over 90% of students are positive and strongly positive about that) and that it is convenient to use in general (about 80% of students are positive and strongly positive about that). 100% and 85% of students respectively were positive and strongly positive that adding and browsing summaries is easy. Although in the current version there is no navigation logging, students did report that they were browsing through the summaries submitted by their peers. 10 out of 14 students reviewed works of others, including 4 out of 14 that reviewed many of those.

A significant portion of the questionnaire was related to the procedure of indexing papers with concepts. As mentioned before, two methods of indexing were introduced: list based and hierarchy based. Almost two thirds of the students who submitted questionnaires preferred the hierarchy-based indexing (9/14), while the rest preferred the list-based interface (5/14). The ease of indexing in general however was more controversial. Only 50% of students were positively convinced that the indexing (regardless of the interface) was easy. About 20% of students had no opinion about it and about 27% were negative or strongly negative.

Further analysis of students’ feedback, including the freeform answers and informal conversations, convinced us to continue investing efforts in refining CoPE’s interface. Interface in general needed more work and indexing interface in specific that was the main source of the negative attitudes toward CoPE.

5. CoPE v.2 Interface

CoPE v.2 interface uses the same folder-document paradigm just like CoPE v.1. The main window of CoPE v.2 is separated into left navigation frame with Microsoft Explorer-like tree structure and main frame where the top portion is occupied by menu and main area is where content of the folders and documents is displayed (Figure 5). The major modifications of the navigation frame are:

- every object such as folder, paper, summary, or set of indexing concepts is represented as an element of the tree, and
- every type of objects (e.g. folder type or paper type) has a unique icon that distinguishes it from others.

This minor modification achieves better outlook in general and faster separation of various types of objects.

Menu has also been change towards unification (Figure 6). There is a button to toggle show/hide the left frame, reload the content of the system (this might be useful while some modifications happen in parallel, e.g. two users are working at the same time), button for going to the upper level folder, and logoff button. The two remaining buttons are for adding objects (folders and documents) and editing them.

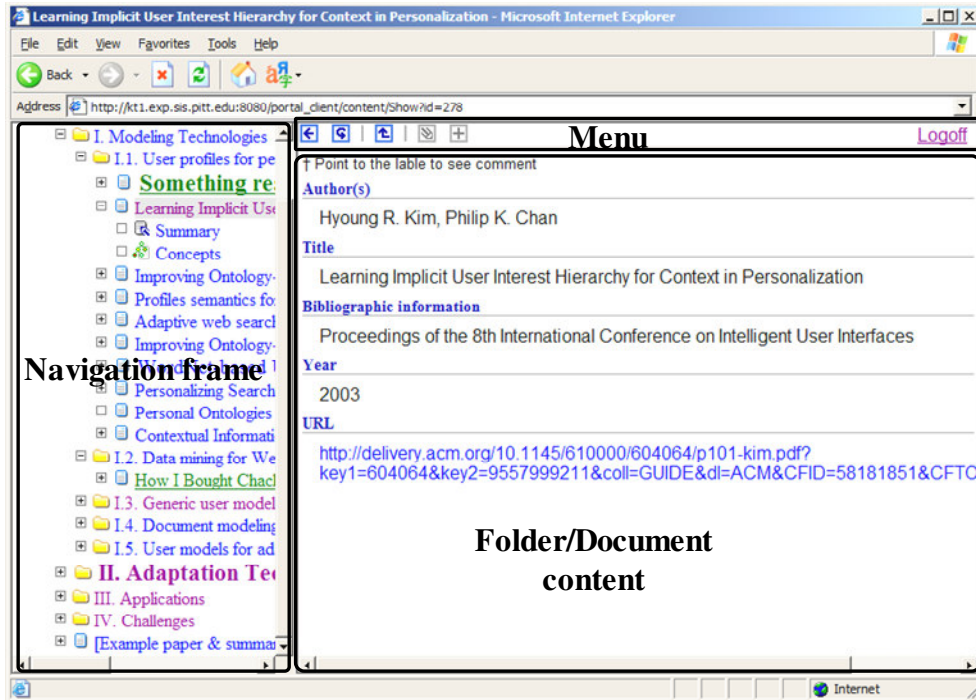


Figure 5 CoPE v.2 interface

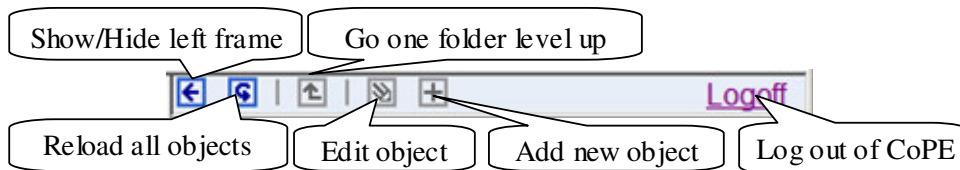


Figure 6 CoPE v.2 menu

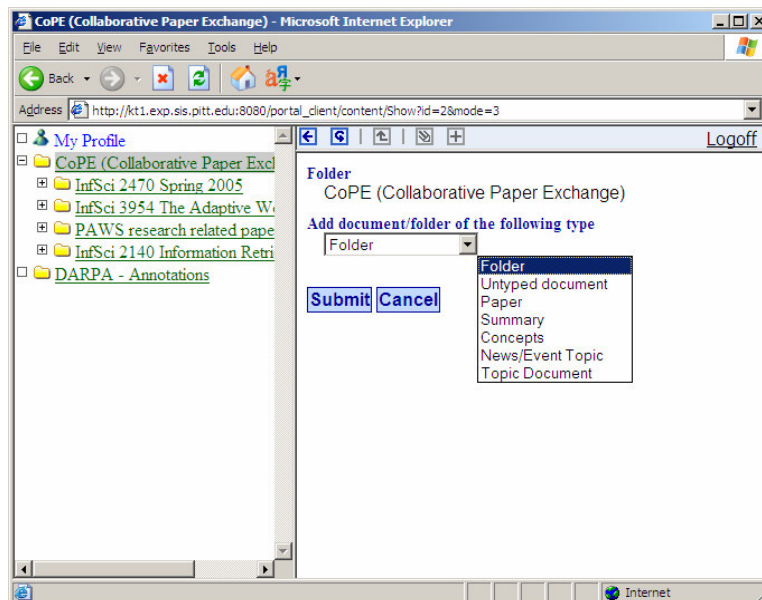
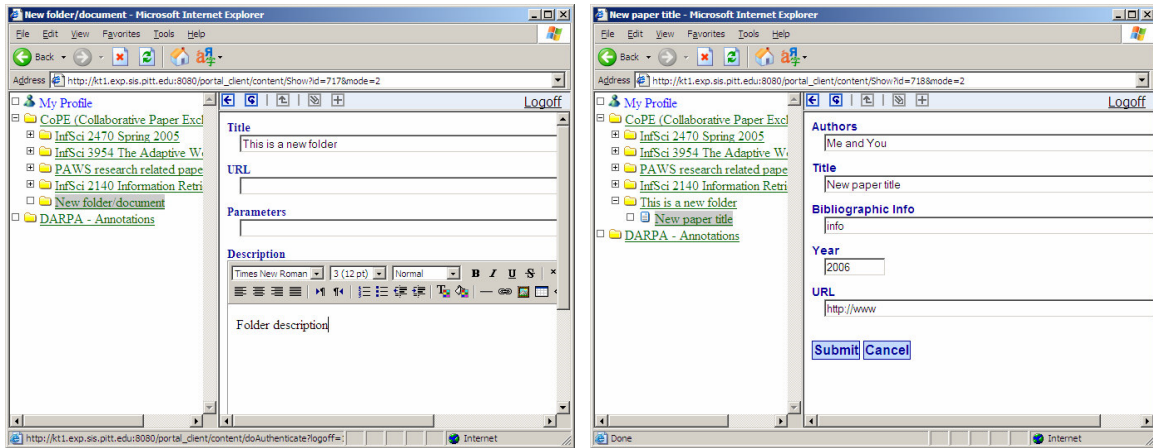


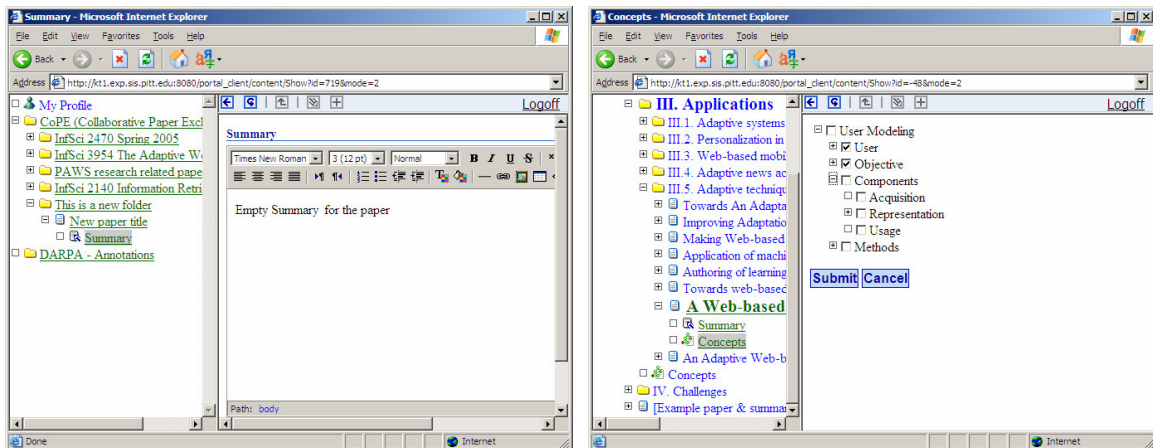
Figure 7. Adding an object in CoPE v.2

Interface for adding an object is shown in Figure 7. User should choose an object s/he wants to add. Then s/he should hit submit and a desired object will be created and opened in the edit mode. Editing interface depends on the object user is currently working on. In case of folder user enters title and description (Figure 8a), for paper user enters title, authors, bibliographic information and URL (Figure 8b), for summary user enters the summary text (Figure 8c), and the case of indexing concepts (Figure 8d) will be covered later. To commit changes user has to click of Submit button, to discard – on Cancel.



(a)

(b)



(c)

(d)

Figure 8 Editing objects in CoPE v.2

6. Indexing Papers in CoPE v.2

Interface for indexing papers with ontology concepts has gone through major revision. Many of the CoPE v.1 complained that both indexing interfaces – list interface and hypertree interface – are inconvenient. The list interface is too bulky – requires too many clicks to have work done. And the hypertree interface visualization is not trivial and hard to understand sometimes. Also CoPE v.2 allows using more than one ontology at the same time.

Instead, CoPE v.2 offers a collapsible tree indexing interface (Figure 9). Concepts are indented according to their hierarchy level. Next to each concept there is a check box. To

select a concept user must check its box, to unselect—uncheck it. Again to submit changes user should click submit button, to discard changes – Cancel button. The choice of collapsible tree was driven by (Chimera and Shneiderman, 1994) where authors have compared multiple interfaces for browsing large volumes of hierarchical data and have found that collapsible trees are the optimal choice.

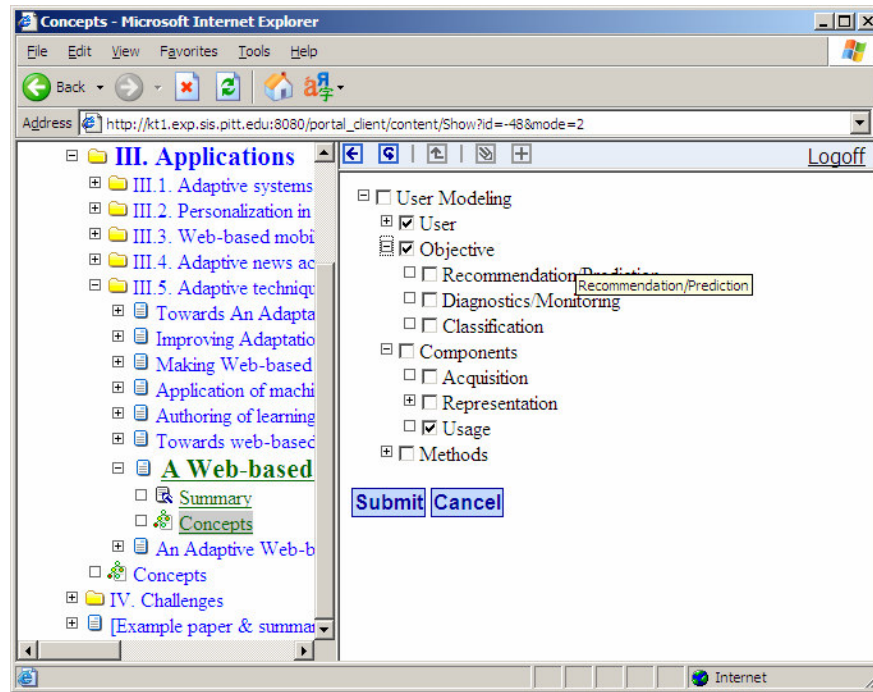


Figure 9 Indexing in CoPE v.2

7. Access Rights in CoPE v.2

In CoPE v.2 most of adding/editing activity is constrained by the access rights. Access rights are applied to subtrees of the objects (folders and documents) and prescribe which users or groups can perform adding and or editing activity on which types of objects. Each right is of the following format:

{User/Group, RightType, ParentType, ChildType, Node, Quantity, Description, Owner},

where:

- User/Group – determine an individual user or a group that exercises the right; a right available to all users/groups would have an ALL_USERS value specified;
- RightType – view, edit, add; if edit or add right is granted view right is assumed; the list of rights can be extended; a right allowing all operations will have an ALL_RIGHTS specified here;
- ParentType, ChildType – the types of the objects the right is applied to (ALL_TYPES will be specified to denote all possible types of parent or child objects):
 - e.g.1 if ParentType=Folder, ChildType=Paper, and RightType=Add, it means that a user can add Papers to a Folder;
 - e.g.2 if ParentType=Folder, ChildType=Paper, and RightType=Edit, it means that a user can edit Papers that are inside Folders;

- Node – the object of the portal that the right is recursively applies to; ALL_NODES macros differentiates rights applicable to all objects of the portal;
- Quantity – useful for RightType=Add, denotes how many objects can be added, -1 denotes infinitely many (unbounded), 0 is meaningless;
- Description – contains textual explanation for the right;
- Owner – when this flag is set to true the right applies only to objects that a user is an author of (created him/herself):
 - e.g. if ParentType=Paper, ChildType=Summary, RightType=Add, and Owner=true – a user would only be able to add summaries for the papers s/he added.

When a user is logging into a system the following rights are loaded:

- rights that apply to the user personally;
- rights that apply to the group the users logged into;
- rights that apply to all users/groups (specified by ALL_USERS macro).

Loaded rights are used to select a subset of the CoPE resources that a user can view. Loaded rights are kept in the user session cache and every time user changes his focus (selects different object to work with) actions the user can perform are checked against them.

8. Future Plans

CoPE has a lot of room for future improvement and. It was designed as an architecture that can be extended in various directions to service various need of instructors and students. One of our immediate goals is to run an evaluation study with CoPE v.2 and determine the most needed features that have to be implemented. Here is a tentative list of functionalities we think are most crucial for improving CoPE v.2, all but the last of them were actually suggested by CoPE users (both students and instructors):

- provide interface for access rights delegation; right now rights are entered manually which puts a great responsibility on an administrator, decreases the usability of the system and increases the risk of the mistake;
- allow users to see the names of the authors of CoPE objects (papers, summaries, etc) or create a special access right “Allow to see an Author”; this might require extending object level access rights with object property level rights;
- allow users to assign ratings to CoPE resources, e.g. “vote” for papers;
- allow descriptions of CoPE papers according to standard formats like BibTeX or EndNote;
- allow links between CoPE papers and publisher or digital library online resources, such as ACM Digital Library, Springer, or DOI;
- allow not only folder-document based structural navigation (that often is a “course organization” structure) but also ontology based navigation, author (of the paper) based navigation, and year based navigation of the papers;
- in addition to delegative rights that “allow” introduce restrictive rights that “forbid”;

References

- Blackboard Inc. (2002) Blackboard Course Management System 5.1, Blackboard Inc.
<http://www.blackboard.com/> (Accessed 21 January, 2002)
- Brusilovsky, P. and Miller, P. (2001) Course Delivery Systems for the Virtual University. In: T. Tschang and T. Della Senta (eds.): *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University*. Amsterdam: Elsevier Science, pp. 167-206., also available at
<http://www2.sis.pitt.edu/~peterb/papers/UNU.html>.
- Brusilovsky, P., Sosnovsky, S., Yudelso, M., Chavan, G. Interactive Authoring Support for Adaptive Educational Systems. In *Proceedings of 12th International Conference On Artificial Intelligence in Education*, Amsterdam, 2005 (accepted).
- Chimera, R., Shneiderman, B. An Exploratory Evaluation of Three Interfaces for Browsing Large Hierarchical Tables of Contents, In: *ACM Transactions on Information Systems*, 12(4):383-406, October 1994
- Gaines, B. R. and Shaw, M. L. G. (1995) Concept maps as hypermedia components. *International Journal of Human Computer Studies*, 43(3), 1995, pp. 323--361.
- Gavrilova, T., Voinov, A., Vasilyeva, E. Visual Knowledge Engineering as a Cognitive Tool. In *Proceedings of International Conference on Artificial and Natural Networks IWANN'99*, Benicassim, Spain, 1999, 123-128.
- HCI Glossary. (2000). Retrieved March 10, 2005, from
<http://www.meanswhat.com/weblog/glossary.html>
- Hugo, J. (2000). HCI Glossary. Retrieved March 10, 2005, from
<http://www.usabilitysa.co.za/hcigloss.htm>
- HyperGraph. (2005) Retrieved January 25, 2005, from <http://hypergraph.sourceforge.net/>
- Lamping, R. and Pirolli, P. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In: Katz, I., Mack, R. and Marks, L. (eds.) *Proc. of CHI'95*, (Denver, May 7-11, 1995), ACM, 401-408.
- Uther, J. and Kay, J. VIUM, a Web-based visualization of large user models. In: Brusilovsky, P., Corbett, A. and Rosis, F.d. (eds.) *User Modeling 2003. Lecture Notes in Artificial Intelligence*, Vol. 2702. Springer Verlag, Berlin, 2003, 198-202.
- Ziegler, J., Kunz, C., Botsch, V., and Schneeberger, J. Visualizing and exploring large networked information spaces with Matrix Browser. In: *Proc. of 6th International Conference on Information Visualization, IV'02*, (London, UK, July 10-12, 2002), IEEE Computer Society.