Databases and Search Engines: Tools for Reuse of Course Materials

Edward F. Gehringer, Tony M. Louca
North Carolina State University
{efg, tmlouca}@eos.ncsu.edu

Abstract

We have developed software for creating databases of course materials on the World-Wide Web. The goal is to allow instructors at different institutions to share materials and develop them jointly. Our first two databases, in computer architecture and object technology, comprise thousands of problems and lectures downloaded (with permission) from course Websites around the world. The database is searchable by classification or fulltext string.

To populate the databases with up-to-date material, we started by building a list of course Websites. Using several sources, we came up with a list of 73 sites in computer architecture and 40 sites in object technology. However, only a minority of the instructors have allowed us to use their material. To provide access to a larger amount of material, we are extending our search capability to include material on the Web as well as in the database. Users of the database will not, of course, have an automatic right to reuse and adapt material that is not in the database; however, they will be able to ask the copyright holders for permission individually.

The search engine that we are integrating with the database finds course Websites by searching a filtered set of educational domains for sites containing keywords characteristic of course material in the target discipline. We present preliminary results of using this search engine.

1. Introduction

With the advent of the World-Wide Web in the early '90s, instructors began to place course material on line. In 1995, academic attendees from the International Symposium on Computer Architecture indicated great interest in developing a Website of reusable course materials. By 1997, approximately half of the object-technology (OT) instructors attending a workshop organized by the first author had developed course Websites. Contributions were sought, and approximately 500 problems were obtained from nine different contributors. The database went online in 1998. In the beginning, questions were inserted by cutting and pasting them into a browser interface to the database. To automate this time-consuming process, a set of Perl scripts was developed. The scripts would take a URL pointing to the course Website and a regular expression. Based on the regular expression, the script would download all the homeworks (or other course material) it found and try to separate them into individual problems. Then, these problems would be inserted in the database. These scripts, however, often required user intervention and did not keep track of where the material had been loaded.

2. WebAssign

The database schema and search functionality of our database, on the other hand, were robust from the outset. Material is stored [1] in WebAssign [2], a Web-based multimedia exam and homework-grading system developed at NCSU. This gives our course database the search functionality and Web accessibility of the physics database. Although they share software with WebAssign, our course databases are totally separate from the physics database.

Teaming up with an existing on-line testing system freed us from the need to do database programming, and thus permitted us to bring up a small system with only a few thousand dollars of internal funding. Eventually, it will allow the problems in the database to be used for quizzes administered over the Web and graded automatically.

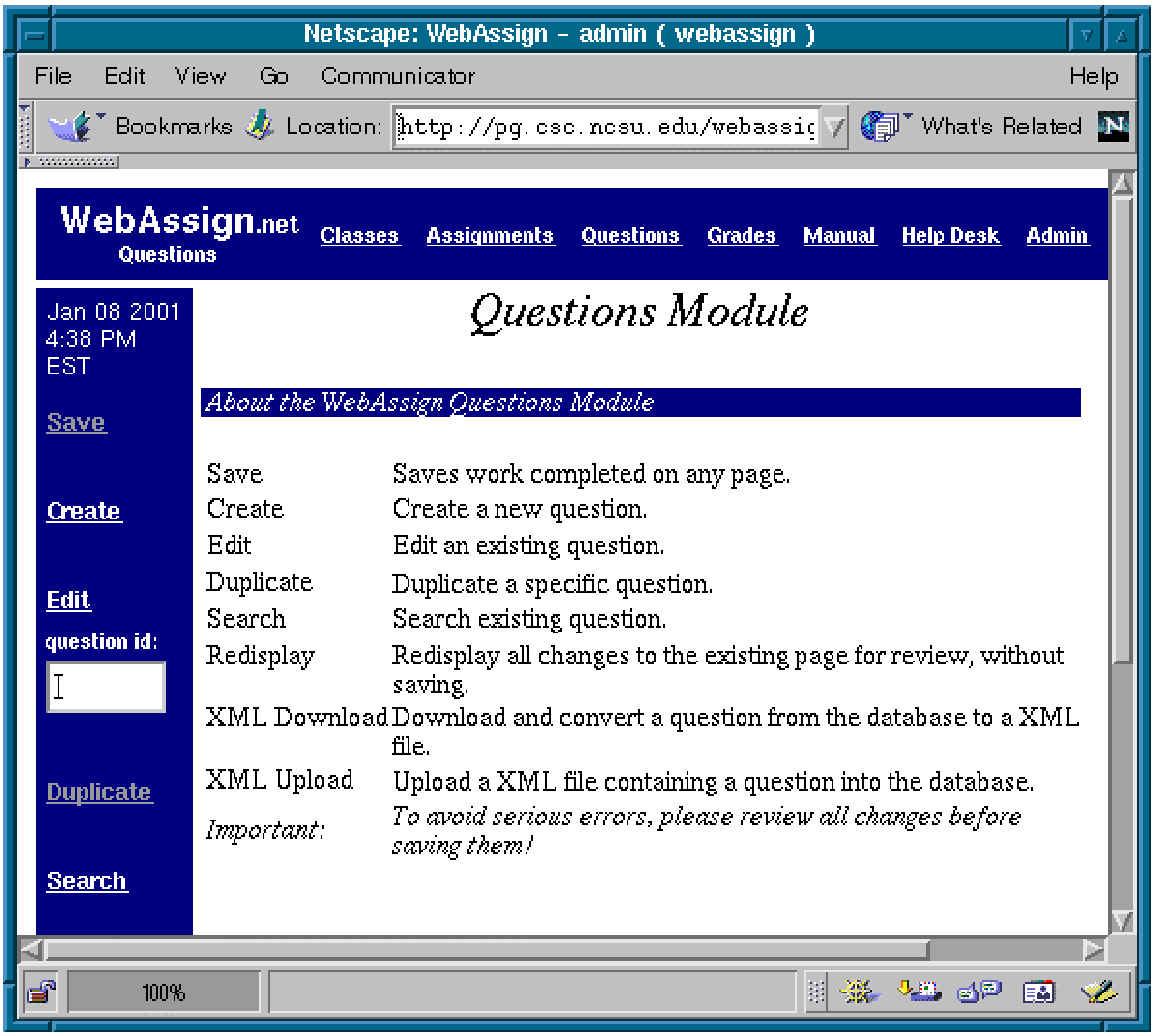

Figure 1. Questions module

As our database progresses, however, the limitations of WebAssign are becoming more apparent. First, information can only be entered into WebAssign through a limited number of fields defined by the database schema. These encode much of the information we need to store (e.g., author, degree of difficulty, related textbook), but do so in their own specific way. We would prefer to encode this information as IMS metadata [3], which is expected to achieve widespread usage. But IMS metadata is based on the XML standard, which, unfortunately, was not supported by WebAssign. Second, WebAssign does not offer printing and download functions, which would useful for instructors trying to reuse the material. Therefore, we have reimplemented the "front end" of WebAssign to meet our needs and implement some of our users' suggestions. We have also extended WebAssign to support XML. We are continuing to store questions in WebAssign format, however, so that the ones that have an objective answer can later be used in machine-graded assignments.

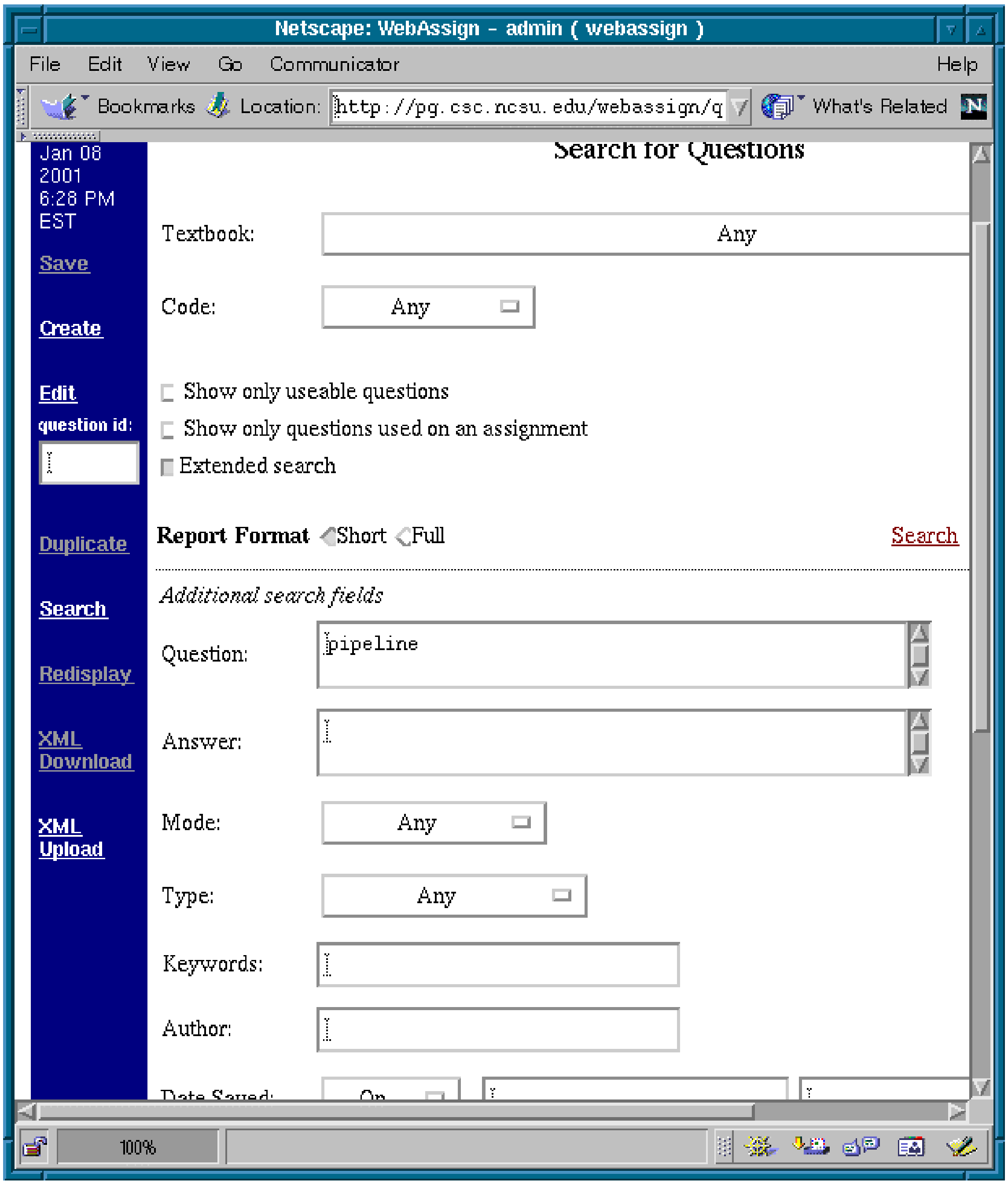

Figure 2. Search screen

Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition

3. A short tour of the database

A user logging into the database is taken directly to the WebAssign questions module (Figure 1). Clicking on “Search” on the left-hand menu brings up a form that gives several search options. A search may be performed by keyword, fulltext, limited to a certain author’s contributions, or to questions submitted by authors using a certain textbook (Figure 2). Then a set of search results is brought up (Figure 3), listing the first line of each question, and giving the user the option to view the text of an item (Figure 4), edit it (this option is only available to the question’s author), or duplicate it, perhaps in preparation for creating a different version of the question. The revised question may then be saved back to the database.

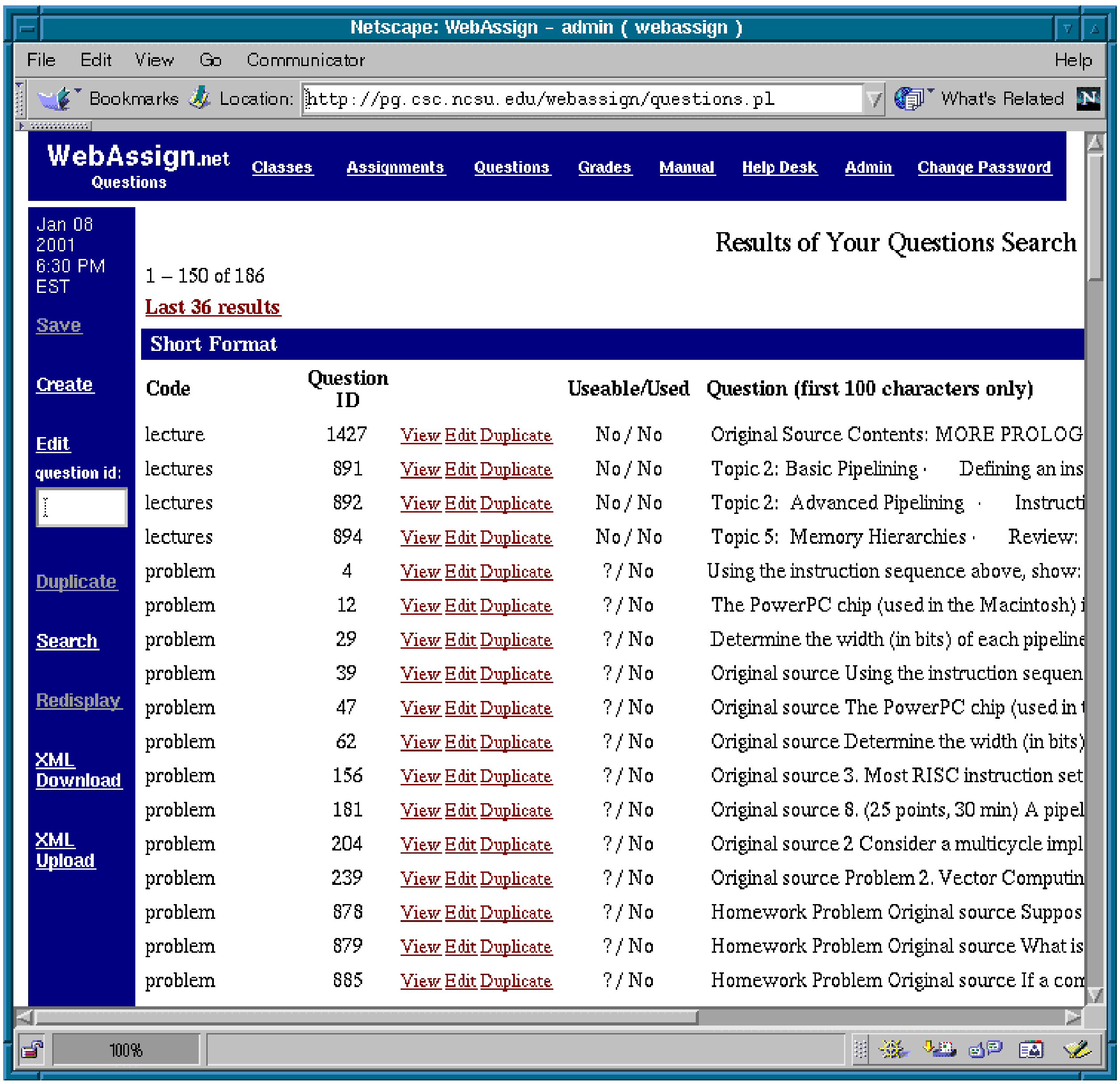

Figure 3. Search results

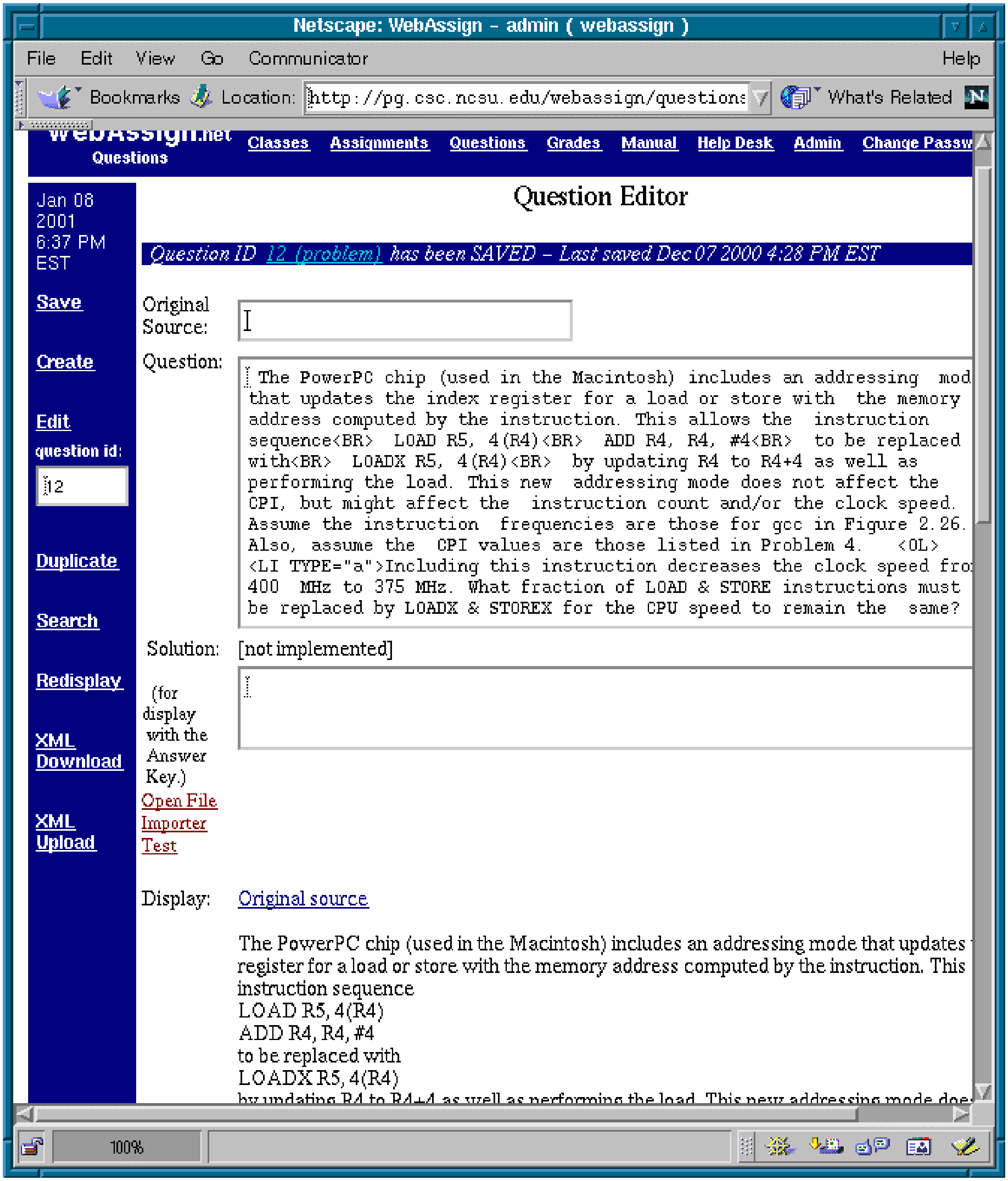

Figure 4. A sample question from the database

4. Populating the database

Our current repertoire of hundreds of questions is useful to many instructors—a survey of about 70 users is reported in [5]. However, coverage of some topics is very thin, and material specifically geared to any but the most popular textbooks is hard to come by. Clearly, we need to find more instructors willing to contribute their online material under a royalty-free agreement. We have used the following approaches to contact instructors:

- We sent e-mail to people with academic affiliations on conference attendance lists.
- We used Web-based listings of faculty in particular disciplines. For the computer architecture course database, we contacted people listed on the WWW Computer Architecture Website [6]. For the object technology database, we used Steve Beaty's list [7] of computer-science course Websites. We also used the list of course Websites that the primary author generated for last year's OOPSLA Educators' Symposium [8].
- We used attendee lists from several specialized education workshops.
- We have done presentations and demos at conferences like the American Society for Engineering Education 1999 annual conference, SIGCSE 2000, Frontiers in Education 2000, and the 2000 OOPSLA Educators' Symposium.
- We have used Web search engines to find course Websites on related material and then contact the instructors.

In addition, existing users of the database referred many users to us. Dave Patterson of UC-Berkeley deserves special mention for referring potential users.

These approaches yielded a total of 73 course Websites in computer architecture and 40 in object technology. However, only 31 of the architecture and 16 of the OT instructors agreed to let us load their material into the database. Seven of the architecture and 5 of the OT instructors refused, and the remainder (35 and 19, respectively) did not reply.

Those who gave us a reason for declining cited content or copyright reasons about equally often. The major content reason (cited four times) was that the materials were still under development and not polished enough to release to the world. A fifth instructor said that his materials are always under development, and loading a snapshot into the database would only contribute to his version-control problem. Another said that the lecture notes on his site were taken by students, and he couldn't vouch for the accuracy.

A number of legitimate copyright concerns were cited as reasons for not participating. Two of the respondents said that some of the material on their site was from textbooks they had authored, and they no longer owned the copyright. Another said that many of his problems were taken from textbooks he used in classes. Two said that their lectures were developed in conjunction with former colleagues, so they didn't feel it would be right to release them. Two said they had a general policy against giving permission for others to copy material.

5. Using a search engine

A specialized search engine might appear to be an alternative to a database. It would certainly have access to more material than could be placed in the database. It is not clear, however, that this would necessarily be valuable, if it also returned large numbers of marginally useful pages on every search. It would also fail to find material that is not linked to from the outside world. A search engine would find up-to-date versions of pages, whereas the material in a database might get out of date (though our application is programmed to download updated versions of material whenever an update is noted to one of the pages it has loaded). It would also retrieve material from instructors who have not responded to our request, and those that have declined. Of course, users of the search engine should still get permission from those whose material they are going to reuse. But authors might be much more willing to grant permission on a one-time basis than they are to allow their material to be loaded into a database and reused throughout the community.

On the other hand, a database has several advantages over a search engine. Its contents remain available, even when a Web page is taken down (e.g., at the end of a semester). It allows instructors to reuse material immediately, without seeking permission from the author. This can be a real advantage when one is trying to write a lecture, homework assignment, or exam at the last minute. It stores material in a common format (HTML), facilitating its reuse in new documents. It allows a better focused search, free of the extraneous material that always appears among the results from a search engine.

Therefore, we have augmented our database with a specialized search engine. It builds an index, using our lists of course Websites as the root. Whenever a database search is performed, the user may click on “Extended search,” and retrieve the search results in a separate window (Figure 5).

It should also be mentioned that a problem in the database usually possesses a link to its “original source”—the Web page from which it was taken, and a link to the local copy. The “original source” link is used to view the problem in the context of the assignment from which it was taken. This is useful in case a common set of assumptions apply to all problems in a problem set, or in case one problem is to redo a previous problem with a different set of conditions. The “local copy” link serves as a backup for the original source in case the original source is moved or removed after the end of the term when the course is taught.

The “original source” link also provides a way to access the most recent version of a problem. For example, if an instructor assigns homework and then discovers an error, (s)he may update the handout on the Web. The “original source” link will access the current version. This gives our database some of the same ability a search engine has to access up-to-date material.

6. Status of the database

At the time of writing, the database contains 883 entries for computer architecture material. Of these, 849 are homework and test problems, and 34 are lectures. The object technology database, which we started to develop last fall, has 286 entries, most of them homework and programming assignments with 19 lectures.

7. Conclusion

Building a useful database of course material is a challenge. It is a worthy challenge, because a successful attempt can facilitate large increases in teaching productivity among its users. It is not an easy task: Specialized software needs to be written, and permission needs to be obtained from all copyright holders. Marrying the database to a search engine offers a way around the laborious process of obtaining copyright permission from dozens of sources. Not only does it allow us to offer more material to our users, but it will help to advertise the database, when our users seek permission to reuse the material of others whose sites are indexed by the search engine.

Figure 5. Results from our specialized search engine

For too long, faculty have used a pre-industrial model for developing courses, "handcrafting" each lecture, lab exercise, and exam. It is time we engineering instructors adopted a more engineering-like approach, developing software and databases that can allow courses to be assembled from interchangeable parts. We hope our project has made an important contribution to this process.

Bibliography

[1] Edward F. Gehringer, Ana E. Goulart, Xiaokang Sang, and Chenhao Geng, "Computer Architecture Course Database: Implementation and Status Report," Workshop on Computer Architecture Education, held in conjunction with the 25th *International Symposium on Computer Architecture in Barcelona*, Spain, June 1998 (http://www4.ncsu.edu/~efg/isca-caew98.ps).

[2] Aaron P. Titus, Larry W. Martin, and Robert J. Beichner, "Web-based testing in physics education: methods and opportunities," *Computers in Physics* 12:2, March/April 1998, pp. 117–123.

[3] The IMS meta-data project http://www.imsproject.org/metadata/

[4] Edward F. Gehringer & Tony M. Louca, "Using the Computer Architecture Course Database," Proc. Workshop on Computer Architecture Education, Vancouver, June 10, 2000, pp. 85-89, also in September 2000 IEEE Technical Committee on Computer Architecture Newsletter

[5] Edward F. Gehringer & Tony M. Louca, "A Web-Based Computer Architecture Database: Improving the Viability," Proc. Frontiers in Education 2000, Kansas City, Oct. 18-21, 2000

[6] The WWW computer architecture maintained by Milo Martin http://www.cs.wisc.edu/arch/www/

[7] Beaty, Steve, "Resources for university teaching," http://lamar.colostate.edu/~beaty/

[8] Edward F. Gehringer, "A tour of course Websites," OOPSLA '99: Object-Oriented Programming Languages, Systems, and Applications (Educators' Symposium), Denver, CO, Nov. 1-5, 1999

EDWARD F. GEHRINGER
Edward Gehringer is an associate professor in the Department of Electrical and Computer Engineering and the Department of Computer Science at North Carolina State University. He has been a frequent presenter at education-based workshops in the areas of computer architecture and object-oriented systems. His research interests include architectural support for persistence and large object systems, memory management and memory-management visualization, and garbage collection. He received a B.S. from the University of Detroit(-Mercy) in 1972, a B.A. from Wayne State University, also in 1972, and the Ph.D. from Purdue University in 1979.

TONY M. LOUCA
Tony Louca graduated from the American University of Beirut in 1997 with a B.E. in Computer and Communications Engineering. He finished his M.S. in Computer Engineering from North Carolina State University in 2000. Currently, he is working as an ASIC design engineer for NVIDIA corporation, a leading graphics chip company in Santa Clara, CA.