A Web-Based Computer Architecture Course Database

Edward F. Gehringer
North Carolina State University
efg@ncsu.edu

Abstract

A Web-based database of course materials in computer architecture is being developed. Its goal is to allow instructors at different institutions to share independently developed materials, and to collaborate in developing new materials. This database comprises problems downloaded from the Web sites of courses in computer architecture at universities around the world. The site is searchable by classification or fulltext string for problems on particular topics in computer architecture. The database currently contains homework and test questions. It is planned also to include lecture notes, laboratory exercises, and multimedia teaching materials developed at a number of universities. Materials are gathered for the database by obtaining instructors' permission to include materials from their course Web sites. Scripts have been developed to fetch their material over the Web, separate homework assignments and tests into individual problems, and store them in the database.

This project has been developed in conjunction with the WebAssign project for on-line homework submission and grading. Where the format permits, homework and test problems can be automatically graded. At this writing, the database contains 500 problems. Although this prototype is specific to the field of computer architecture, the software for building the database is usable for constructing databases in almost any academic field.

1. Introduction

The World-Wide Web owes its existence to the Internet, whose original constituents were mainly universities. So it is only natural that the Web should abound with educational materials. As universities scramble to put courses on line, they are in effect creating a large distributed database of course materials, organized in an ad hoc manner, with varying degrees of incompleteness. These materials can potentially be shared to provide an educational experience far richer than any instructor could fashion alone. The challenge is to organize them so that other instructors can locate, pick, and choose the materials most appropriate for their situation.

Though course material is on the Web, it is not easy to find. Consider how one would look for lectures in a particular academic field, say, general chemistry. Searching for a keyword like "valence" might retrieve some lectures, but they would also retrieve research papers, lectures from more specialized courses, pages from the Valence Software company, and pages about the French city of the same name. Following hyperlinks through university sites would not be much better, because most lectures are not on line, or, if they are, they are likely to be removed or moved at the end of the semester. In short, there is no good way to retrieve even a healthy chunk of the relevant lecture material by searching the Web. [a]

Thus, an instructor attempting to create or update a course by building on the efforts of others needs a tool to bring order out of chaos. This calls for a systematic, automated strategy to retrieve and catalog relevant material when it appears on the Web and preserve it in a database for future reference.

Academic course material lends itself well to storage in a database. It is usually in machine-readable format. As long ago as 1995, forty-seven out of sixty-seven computer architecture instructors responding to a survey reported that the majority of their instructional materials were on the computer.[1] With more than 1000 institutions offering distance-education courses [2] which are increasingly Web based, there is clearly enough raw material for course databases in many academic fields. Moreover, this material falls into well understood categories, such as lectures, problems, test questions, labs, software, and multimedia tools. Given such a database, instructors would have little difficulty exploiting it.

Improving the educational experience: In technological fields, education needs to be up to date. Not only must the instructor keep current with the state of the art; (s)he must develop new course materials. This involves course components in all the categories mentioned above. One can rely on problems from a textbook, but too many textbooks provide only hastily composed open-ended problems that are hard to grade and do not force students to work through details of a design. Although some textbooks do have good problems, there may not be enough of them. After one or two semesters, they are "used up." One can also lecture straight from a textbook, but this is not a good idea either. The instructor's knowledge of the material will not be very deep, and the textbook may reflect the idiosyncratic perspective of the author (e.g., perhaps focusing the author's research out of proportion to its importance in the field).

It is difficult for instructors to devise realistic and well thought-out problems over material that is new. Moreover, to grasp a concept, some students need more examples than others. Students frequently ask for extra solved problems to use as a study aid. For material that is state-of-the-art, it is hardly ever possible to find enough.

Education engineering: Centuries ago, all kinds of manufactured goods—shoes, furniture, carriages—were made to order for the local customer. Manufacturing is no longer done that way ... but education is. "Handcrafting of courses" is expensive, and it is a major reason why the cost of education has been increasing faster than inflation. It makes little sense to have scores of highly trained researchers spending their time devising lab exercises or test questions over the same material, semester after semester. Rather they should be spending their time solving open problems or advancing the frontiers of technology. And their teaching hours should be spent more with students and less on preparation.

[a] The IMS project (see Section 5) is developing standards to facilitate locating educational materials.

This suggests the notion of "education engineering" —developing methodologies and tools to create educational materials more quickly and in greater volume, and disseminate them without loss of quality to the increasing numbers of students seeking a technologically up-to-date education. Distance education[3] and Internet delivery[4] are successfully attacking the dissemination of course materials. But it is still necessary to devise more effective strategies for creating them.

Multicampus collaborative work: It is one thing to reuse a colleague's lectures and lab exercises; it is another thing to collaborate on teaching a course. Multicampus collaboration in teaching distance-education courses is much talked about but rarely practiced. Upon reflection, it is not difficult to see why. Instructors have different ways of covering material. Some emphasize the concepts and elide the details. Some stress problem solving; others stress design skills. Courses taught be multiple instructors have a tendency to settle into a series of modules, connected loosely at best.

One reason for this is that it is hard to know in advance about a colleague's teaching style. For all the talk of "teaching portfolios," few of us have taken the time to compile a set of course materials in a form that can be presented to an outsider. A database of course materials can change that. It takes on the responsibility of gathering course materials from instructors at many universities and making them searchable by author and topic. It makes it possible for an instructor to find a partner whose interest area *and* teaching style are a good match. Indeed, instructors who find themselves reusing each other's material may well be encouraged to share the responsibility for developing a specialized on-line course.

2. Progress to Date

Over the past year and a half, we have built a prototype Computer Architecture Course Database. It is appropriate to begin with a subfield of computing because a lot of material is already on the Web[1] and the instructors are computer experts who can cope with difficulties like material retrieved in the wrong source format (e.g., RTF instead of HTML). Computer architecture is an appropriate area of computing, because it has a well delineated corpus of basic knowledge, is rapidly evolving as the speed of computers increases, and admits of both objective questions and open-ended design problems.

3. WebAssign

The software for the Computer Architecture Course Database[5] is built on top of WebAssign,[6] a Web-based multimedia exam and homework-grading system developed in the NCSU Physics department using Sybase 11 and a Sun Ultra 2.1 server. This allows us to share the database format and Web accessibility of the physics database. Although it shares software with WebAssign, the Computer Architecture database is totally separate from the physics database.

To this software, the Computer Architecture database adds the ability to download problems from Web pages and insert them into the database. Problems are downloaded using Perl scripts that can take a wildcard URL and, given a set of editing instructions, automatically break the page apart into separate problems. Each problem is then written to a separate file, in a format that can be imported into the database using another script that we jointly developed with the WebAssign implementation team.

It has been very advantageous to team up with an existing on-line testing system. It has 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, although at present, few problems in the database have objective answers that permit automatic grading.

Neither WebAssign nor the course database are specific to any academic field. WebAssign began as a system for automatic grading of physics homework, and is now in use by 5,600 students in physics, mathematics, computer science, statistics, and business courses. The software for our course database is capable of being used to gather course material in any discipline over the Web.

4. The existing computer architecture database

To enter the database, go to `http://wwwassign.physics.ncsu.edu/comparch`, and log in.[b] You will be directed to the main faculty page. It contains a number of links that are applicable to the automated testing system. To use the Computer Architecture Course Database, however, choose "Questions" from the main-menu dropbox in the upper left-hand corner. This will take you to the "Search Questions" screen (see Figure 1).

Currently, all of the questions in the database have a code of "problem". You may retrieve them all by filling in the Code field at the top with the characters "problem". You may also search on a text string by filling in the "Question text" box. For example, a search on "delayed branch" retrieves the two questions shown in Figure 2. It is also possible to search by keyword, e.g., "cache".

We have developed scripts for importing problems that are on the Web in HTML and ASCII. We have imported problems in PDF format by using a PDF-to-HTML converter on a per-file basis, and then breaking them into separate files using the HTML script. Microsoft Word documents have been imported by using Word’s HTML converter to translate to HTML in bulk, and then use the HTML script.

At the time of writing, the database contains only problems and lab exercises. There are about 500 problems in the database, derived from ten courses by nine different instructors. Of these, the largest concentration are on caches (78 problems), followed by computer arithmetic (largely questions on floating-point formats and arithmetic from computer-organization courses), performance, and instruction sets. The largest number of problems, about 200, have been

[b] If you desire a guest account, please contact the author.

translated from Microsoft Word, followed by 114 translated from ASCII text. HTML is close behind, with 94 problems.

While the database is currently operational, it does have significant limitations. First, only about half the problems have solutions in the database. This is due to the understandable reluctance of instructors to place homework solutions unprotected on the Web for long periods of time. When the solutions do appear, they are frequently in different files from the problems, so we have written a script to associate a solution with the corresponding problem at the time it is placed in the database.

Second, better filtering needs to be employed to remove questions that are really only pointers to other questions, e.g., "Problem 4.2 from the text," or "5 and 6. Refer to homework question 7," as well as questions unrelated to computer architecture, e.g., bonus questions like, "Circle the names of NBA players who are *not* Georgia Tech alumni." Perfect filtering is probably impossible, but simple heuristics could remove most of the deadwood.

Third, better cross-referencing of questions to Web sites is desirable. All of the questions entered recently contain a pointer to the Web page from which they were obtained (called "Original source"), so that they can be seen in context. But it would be good also to have a link to the course homepage, so that one could determine the textbook used (since several questions refer to pages in the text).

Interest in the project within the computer architecture committee has been high. When the database was initially proposed three years ago, [1] four dozen instructors indicated a willingness to participate. But when we began to gather questions in July 1997, few instructors followed through. After we succeeded in obtaining contributions from several faculty, interest again began to grow. To date, 57 instructors at 44 different universities have requested access. Most have browsed it only casually, although a large majority of those who have actually attempted to use it have found it useful (see "Survey of current users" on the next page).

A note on intellectual property: The problems included in the database are the intellectual property of their authors. Thus, we must be careful not to include problems without permission, for example, problems taken from textbooks. All contributors to the database are required to sign a letter stating:

> This letter is to confirm that all questions, answers and other materials I have submitted to your Computer Architecture course database are original, and not drawn from any textbook or other copyrighted source.
>
> While you are welcome to use and distribute this information via your database, I reserve the right as author to reuse this material in the future.

In effect, this grants the database a nonexclusive right to use submissions on a royalty-free basis.

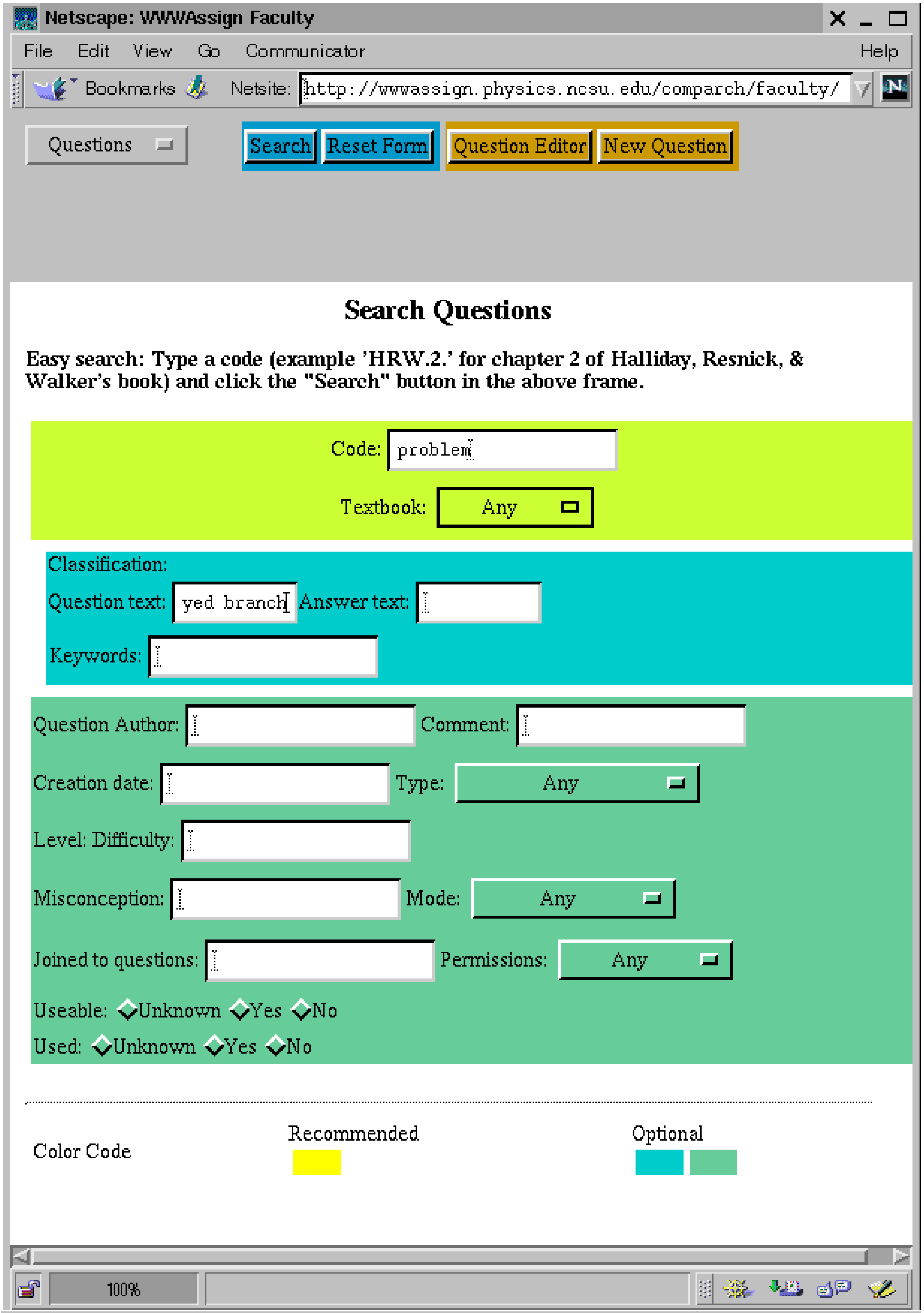

Figure 1: Search Questions screen

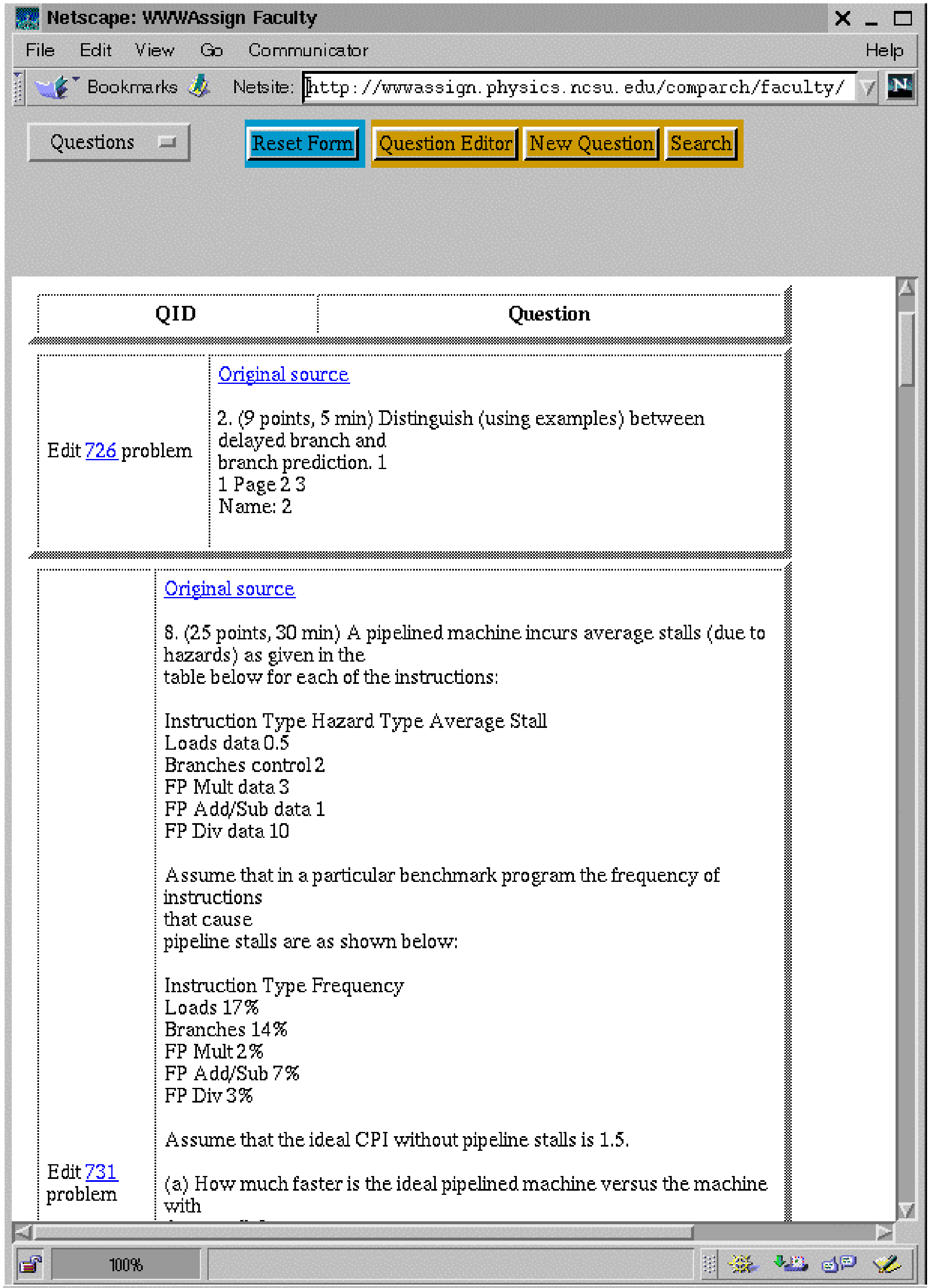

Figure 2: Retrieving "delayed branch" problems

5. Related work

The IMS project (http://www.imsproject.com) is addressing the problem of finding educational materials on the Web by devising standards for "facilitating online activities such as locating and using educational content." One of its goals is to allow content from multiple authors to be integrated into a coherent whole. The WebAssign project plans to incorporate IMS metatags when this becomes feasible. At that point, our project will be able to add metatags automatically to material that would not necessarily be tagged by its authors, and thus enhance the ability to locate specific kinds of content.

There have been at least two other efforts to promote sharing of course materials over the Web. Steve Beaty[7] at Colorado State compiled a list of pointers to Web sites of courses in all areas of computing at `http://lamar.colostate.edu/~beaty/`. This database currently contains about 65 courses. More recently, Beaty [8] has proposed a WWW domain devoted entirely to course material.

The author of this paper has compiled a list of courses in object technology in conjunction with education-oriented workshops at OOPSLA '97 and '98; this material can be found at `http://www2.ncsu.edu/eos/project/csc/reuse-workshop/database.html`[9] Currently, information on about 50 courses is included; of these, 25 have Web sites.

In addition to WebAssign, there are dozens of other Web-based assessment and testing systems.[c] Though such systems are not the focus of this paper, it bears noting that at least one of them, the CAPA system (http://www.pa.msu.edu:80/edu/ CAPA) has been collecting a repository of problems by CAPA users at different institutions. This approach differs from ours in two important ways: First, our software is designed to import problems from arbitrary Web pages rather than limiting its reach to problems designed for a particular Web-based testing system. Although a closed system might build up a formidable repository of problems for introductory courses, it is doubtful whether this approach could ever capture a sufficient number of problems for advanced courses on new areas of technology. Second, when CAPA imports external problems, they must be converted to the CAPA language. Our system is designed to translate material automatically—though at this stage of development, some hand-massaging of the generated HTML code is often needed.

6. Enhancing the database

The main thrust of this work is to build a prototype that will be truly useful with ordinary effort to a broad range of faculty in a particular field, computer architecture. This can only be done with extensive feedback from our users to determine what is most useful to them. In addition, we need to design appropriate experiments to find out what makes course materials easy to reuse, and whether students have benefited by hearing better lectures and having better problems to solve.

---

[c] Many of these are listed at http://www4.ncsu.edu/~swbonham/list-of-wats.html.

7. The survey of current users

To determine what enhancements are most desired by our present users, a Web-based survey was conducted. Eighteen of the 57 account-holders responded. This is a response rate of only 32% overall, but it probably represents a far higher percentage of the actual users, because seven others informed us by e-mail that they had not yet used the database.

(The most common reason for non-use was that the respondent had not taught a computer-architecture course since gaining access.) Results are given in Table 1.

Table 1: Results of the user survey

Question 1. How useful do you find the Computer Architecture Course Database?

Very useful	7
Useful	8
Neither useful nor useless	3
Not very useful	0
Not useful at all	0

Question 2. Which changes would make the database more useful to you?

Inclusion of lecture notes	11
More solutions to problems in the database	10
Better formatting of problems in the database	10
Making material available in other formats, e.g., RTF, HTML	9
More questions for the kind(s) of courses I teach	6
Automatically getting e-mail when problems/lectures that contain specific terms are posted	6
Greater focus on lab exercises rather than problems	5
Greater focus on test questions rather than problems	4

Question 3. This database is *more* useful to me than a list of Websites for coursese similar to the one(s) I teach.

Strongly agree	7
Agree	6
Neither agree nor disagree	5
Disagree	0
Strongly disagree	0

The users who had used the database considered it quite useful, on average. On a scale of 1 to 5, with 5 being "very useful" and 1 being "not useful at all," the average response was 4.22. Another question asked respondents to compare the usefulness of the database with a list of Web sites for courses similar to the ones they taught. When asked whether the database was *more* useful than a list of Web sites, the average response was 4.00 out of 5.00.

Further corroboration for the usefulness of the database approach comes from another survey the author conducted. Users of the list of object-oriented courses [9] were asked if those links to course Web sites were more useful than a searchable database of programming assignments and test questions would be. Their average response was only 2.93, with 5 being "strongly agree" and 1 "strongly disagree." Thus, those who have been presented with a list of Web sites think a database might be more useful.

Four enhancements were desired by at least half the respondents to our survey:

- More solutions to problems in the database. As previously noted, currently only about half the problems have solutions.

- Inclusion of lecture notes. Respondents felt that it would be good to see how other instructors approached particular material. It was a surprise that lecture notes were more frequently desired than test questions or lab exercises.

- Better formatting for problems in the database. Fonts should be readable; tables should be formatted correctly.

- More material in other formats (RTF, HTML, etc.). HTML was far and away the most-desired format. It would be especially easy for students to practice on automatically graded problems in HTML format.

All of these enhancements are pedestrian in the sense that they require no innovative techniques, just time and attention. To the extent that they ask us to obtain more material from instructors, they create an incentive for us to keep in frequent contact with our users. This should enhance the quantity (if not the quality) of feedback.

8. Conclusion

A Web-based course database is an idea whose time has come. First, rising student enrollments in computing[d] and other areas of technology are increasing the demands on faculty time. Second, the increasing pace of technological change is creating a need for more up-to-date courses. Third, instructors are placing more and more of their materials on the World-Wide Web. Fourth, the trend toward distance education is increasing the demand for course materials

---

[d]NCSU's Computer Science enrollment is up 31% from 1994-98, and among entering undergraduates, Computer Engineering majors now outnumber Electrical Engineering majors.

to be available in a form that can be served over the Web. Finally, the public is demanding that more attention be devoted to good teaching. Our approach is semiautomatically to collect course materials already on the Web into a single database. This will greatly enhance the availability of good, up-to-date materials to instructors around the world.

Benefits will accrue to both students and faculty. Students will be able to learn from better-thought-out problems over more recent material. Students will have more good problems to practice on, which will help many of them to improve their understanding. Faculty will be able to design better courses in less time. They will have a greater chance to identify colleagues at other institutions who share their interests and teaching style, and thus a greater opportunity to collaborate on course development and delivery. Finally, as the need diminishes to spend large amounts of time on course preparation, faculty and students will be able to spend more time working with each other, to the benefit of both.

Acknowledgments

The assistance of the following NCSU students with this project is gratefully acknowledged: Xiaokang Sang, Ana Goulart, Chenhao Geng, Zhiling Zheng, Sonali Aditya, and David Steffy.

Bibliography

1. Gehringer, Edward F., "A database of course materials in computer architecture," Proceedings of the First Workshop on Undergraduate Computer Architecture Education, Santa Margherita Ligure Italy, 19 June 1995. Available on the Web at `http://www4.ncsu.edu/eos/users/e/efg/archdb/public/full-paper.html`.

2. Lozada, Marlene, "Look out for distance learning," *Techniques: Making Education and Career Connections* 72:7, October 1997, pp. 24–26.

3. Mather, Mary Anne, "Virtual schooling," *Technology and Learning* 18:8, April 1998, pp. 30–37.

4. Dumont, Raymond A., "Teaching and learning in cyberspace," *IEEE Transactions on Professional Communication* 39:4, December 1996, pp. 192–204.

5. 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).

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.

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

8. Beaty, Steve, "University teaching Web resources," e-mail sent (privately) to mailing list uteach@emess.mscd.edu, Sept. 7, 1998.

9. Edward F. Gehringer, "Reuse in the classroom: Classifying and sharing o-o course materials," *OOPSLA '97: Object-Oriented Programming Systems, Languages, and Applications* (ACM/SIGPlan), Addendum to the Proceedings, pp. 88–92.

EDWARD F. GEHRINGER

Edwad 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.