Computer Architecture Course Database: Implementation and Status Report

Edward F. Gehringer[1, 2]
Ana E. Goulart[1]
Xiaokang Sang[2]
Chenhao Geng[1]
[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7911
{efg, aegoular, xsang, cgeng2}@eos.ncsu.edu

Abstract

A database of course materials in computer architecture is being developed on the World-Wide Web. Its goal is to allow instructors at different institutions to share materials and develop them jointly. 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. At this writing, the database contains 240 problems, and new problems are being added at a rapid pace. In the future, lecture notes and laboratory exercises will also be included. The software is adaptable to other academic fields as well.

Introduction

It is a challenge to teach in a field where technology is rapidly evolving. Not only must one keep current with the state of the art, but one must develop new course materials. This involves several components: lectures, problem sets, lab exercises and programming assignments, and exams. 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. Even taking lectures out of a textbook is fraught with hazards; the instructor's knowledge of the material will not be very deep, and it may reflect the idiosyncratic perspective of the textbook author (e.g., perhaps focusing the author's research out of proportion to its importance in the field).

Nor is an instructor's time profitably spent developing these materials solo. It is time consuming to do so, and it conflicts with the need for faculty to spend *less* time on course preparation, not more. Faculty are under pressure to be more productive in research and spend more "contact hours" with students. Course preparation takes time away from both of these activities.

Clearly, *cooperation* is a good way to attack the time pressure of teaching and research. If instructors reuse and adapt each other's course materials, students will be exposed to a

---

This work has been funded by the North Carolina State University Provost's office under an Instructional Grant for the 1997–98 academic year.

synergy of teaching styles and a diversity of homework assignments that better reflects the range of challenges they will face on the job. Instructors, too, will learn from each other, revising their presentations from more up-to-date material in the lecture notes of specialists in various subfields. The World-Wide Web is a convenient medium for such collaboration.

Previous work

We are aware of two previous efforts to promote sharing of course materials over the Web. Steve Beaty at Colorado State has 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, of which 10 are on computer architecture. The lead author of this paper has compiled a database 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` [Geh 98]. Currently, information on about 50 courses is included; of these, 25 have Web sites.

These efforts make a real contribution to promoting sharing and reuse in the curriculum, but they are not enough—any more than browsing the Web by following hyperlinks is a substitute for finding information with a search engine. What we really need is a database which can, given search terms, serve up homework problems, test questions, or lectures on a particular topic.

The database

The software for the Computer Architecture Course Database is built on top of WWWAssign [TMB 98], 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 WWWAssign, the Computer Architecture database is a totally separate database.

To this software, the Computer Architecture database adds the ability to download problems from Web pages and insert them into the database. (By contrast, most of the problems in the physics database have been taken[1] from textbooks.) 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 WWWAssign implementation team.

Currently we have 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. We have also sought a Postscript-to-HTML translator, but have not been able to find a reliable one.

Use of the database

To enter the database, go to `http://wwwassign.physics.ncsu.edu/compsci`, and log in. You will be directed to the main faculty page (see Figure 1). 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

[1]with permission!

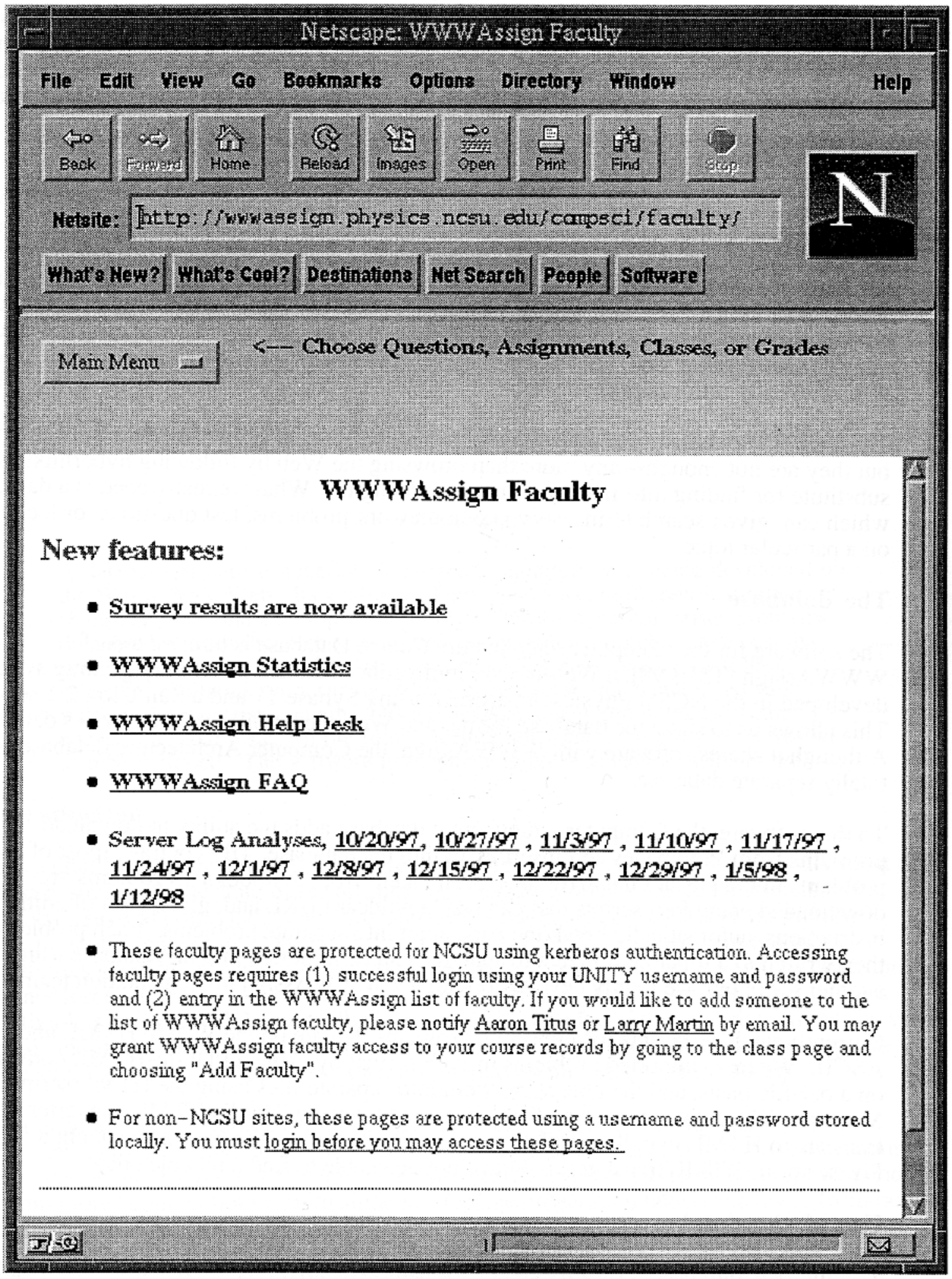


Figure 1: Main faculty screen

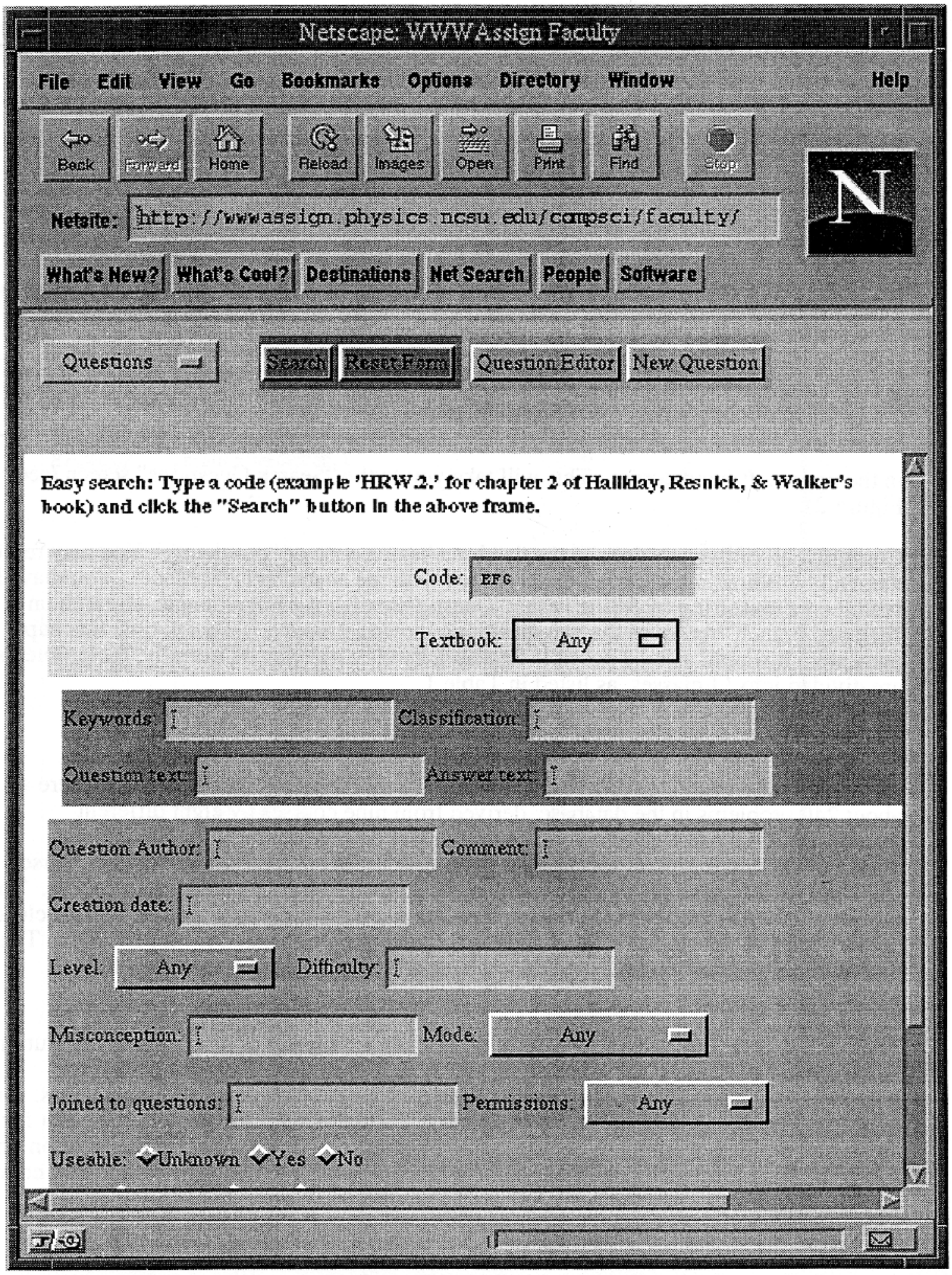


Figure 2: Search Questions screen

Table 1: Classification of Problems in the Database

1. Machine Arithmetic (ALUs, RCAs, FPUs, etc.)
2. Memory Hierarchies (Caches, VM, distributed memories, multilevel memories)
3. Performance Evaluation of computers/processors
4. Instruction Sets (formats, RISC/CISC)
5. Pipelining and ILP (dynamic scheduling, Tomasulo, data/control hazards, branch prediction, superscalar, VLIW)
6. Caches (types, optimizing, multilevel cache)
7. Vector Processors
8. Multiprocessors
9. IO Devices (bus issues, storage devices, RAID)
10. Timing Issues in Computer Design
11. Interconnection Networks (for processors or memory)

in the upper left-hand corner. This will take you to the "Search Questions" screen (see Figure 2).

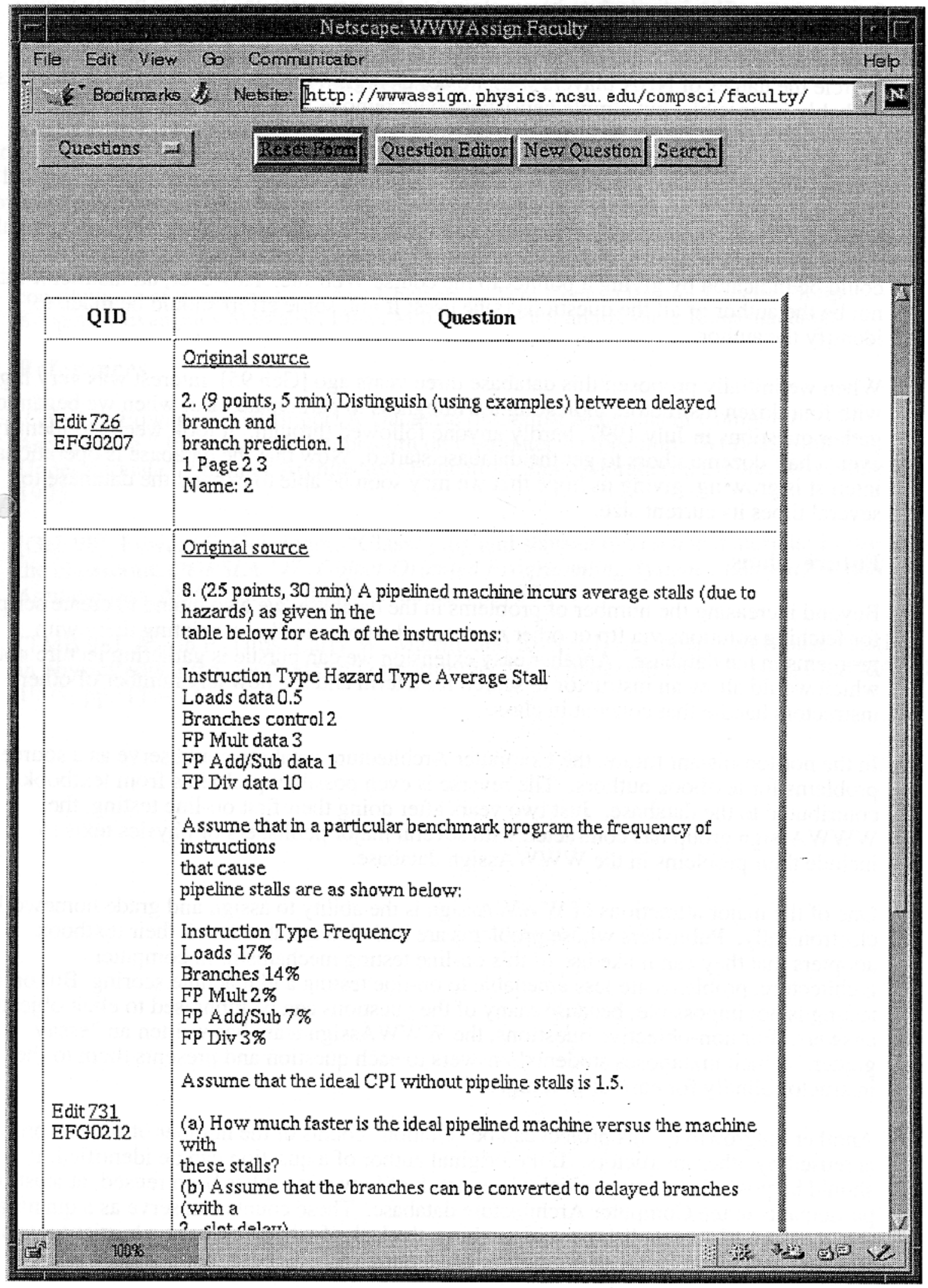
Currently, all of the questions in the database have a code of "problem".[2] You may retrieve them all by filling in the Code field at the top with the characters "problem". You may also search on a text string by filling in the "Question text" box. For example, a search on "delayed branch" retrieves the two questions shown in Figure 3. By the time this paper is presented, it should also be possible to search by classification. Currently, the problems are divided into 11 classes, as listed in Table 1.

Status

At the time of writing, the database contains only problems and lab exercises. There are about 240 problems in the database, derived from nine courses by eight different instructors. Of these, the largest concentration are on computer arithmetic (largely questions on floating-point formats and arithmetic from computer-organization courses), followed by 30 questions on performance, 22 on instruction sets, and 19 on caches. The largest number of problems, 114, have been translated from ASCII text. HTML is close behind, with 94 problems. The remaining 32 have been translated from PDF files. The total number of questions is expected to increase to at least 350 by the time of the workshop.

There are several limitations at present. First, only a minority of 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. Nonetheless, some instructors do post solutions, and the Hennessy-Patterson texts are so dominant that for any given problem from these books, it is likely that someone has placed the solution on the Web. This prompted one of our student helpers to declare that if he ever took a course out of a Hennessy-Patterson book, he would browse the Web instead of doing the homework.

[2] At the time of writing, the code was "EFG", so that is what appears in Figure 2.



QID	Question
Edit 726 EFG0207	Original source 2. (9 points, 5 min) Distinguish (using examples) between delayed branch and branch prediction. 1 1 Page 2 3 Name: 2
Edit 731 EFG0212	Original source 8. (25 points, 30 min) A pipelined machine incurs average stalls (due to hazards) as given in the table below for each of the instructions: Instruction Type Hazard Type Average Stall Loads data 0.5 Branches control 2 FP Mult data 3 FP Add/Sub data 1 FP Div data 10 Assume that in a particular benchmark program the frequency of instructions that cause pipeline stalls are as shown below: Instruction Type Frequency Loads 17% Branches 14% FP Mult 2% FP Add/Sub 7% FP Div 3% Assume that the ideal CPI without pipeline stalls is 1.5. (a) How much faster is the ideal pipelined machine versus the machine with these stalls? (b) Assume that the branches can be converted to delayed branches (with a 2 -slot delay)

Figure 3: Retrieving "delayed branch" problems

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., "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). A related question is identifying authorship. This could be indicated by giving a pointer to the course Web site. However, an instructor may not be the author of all the questions (s)he uses; if not, some effort should be made to identify the author.

When we initially proposed this database three years ago [Geh 95], interest was very high, with four dozen instructors indicating a willingness to participate. But when we began to gather questions in July 1997, hardly anyone followed through. It took weeks to identify even a half dozen authors to get the database started. Now that the database is operational, interest is growing, giving us hope that we may soon be able to expand the database to several times its current size.

Future plans

Beyond increasing the number of problems in the database, we also intend to create scripts for fetching solutions via ftp or other means, and automatically associating them with problems in the database. Another easy extension we can pursue is gathering lecture notes, which would allow an instructor to search for a term and see how any number of other instructors handle that concept in class.

In the not-too-distant future, the Computer Architecture database could serve as a source of problems for textbook authors. The reverse is even possible: problems from textbooks contributed to the database. Just two years after doing their first on-line testing, the WWWAssign group has contracted with several major publishers of physics texts to include their problems in the WWWAssign database.

One of the major attractions of WWWAssign is the ability to assign and grade homework electronically. Publishers whose problems are included are able to tell their textbook adopters that they can make use of this on-line testing mechanism. In computer architecture, problems are less amenable to on-line testing and machine scoring. But on-line testing is not impossible, because many of the questions can be rephrased to elicit objective answers. For non-objective questions, the WWWAssign team has written an "essay grader," which juxtaposes students' answers to each question and presents them to the instructor blindly for ease of grading.

Another outgrowth of this project can be "citation" counts of the number of times a problem is reused by other instructors. If the original author of a question can be identified, it should be possible to automatically keep track of how often it has been reused, at least by participants in the Computer Architecture database. These counts can serve as a quantitative measure of accomplishment in teaching, the lack of which has been a barrier to using teaching as a major factor in promotion and tenure.

With all of the studies that have been done of research productivity, it is surprising that almost no effort has been invested in teaching productivity.[3] We believe that our work is an important start in helping instructors to become more productive by offering more up-to-date courses with significantly less preparation time.

Acknowledgments

In addition to the co-authors, we would like to acknowledge the work of the other students who contributed to the database: David Steffy, Zhiling Zheng, and Sonali Aditya. Also, we are grateful for the support of the WWWAssign implementation team, Larry Martin and Aaron Titus, who have worked with us since our proposal and have always been responsive to our requests for new features. Finally, the database could not have come about without contributions of questions from Ann Chervenak, Tom Conte, Gershon Kedem, Philip Koopman, Evangelos Markatos, Ethan Miller, Yale Patt, and Kishore Ramachandran,

References

[Geh 95] Edward F. Gehringer, A Database of Course Materials in Computer Architecture," International Symposium on Computer Architecture, Workshop on Undergraduate Computer Architecture Education, Santa Margherita Ligure, Italy, June 19, 1995.

[Geh 98] Edward F. Gehringer, "Classifying and sharing o-o course materials: Reuse in the classroom, *OOPSLA '97: Object-Oriented Programming Systems, Languages, and Applications*, Addendum to the Proceedings, pp. 88–92.

[TMB 98] 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] A search of two online databases, Uncover and the Expanded Academic Index, revealed not a single article devoted to this topic.