A Database and Search Engine for Sharing Fine-Grained Course Materials over the Web

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

Abstract

A Web-based database of course materials was constructed to serve as a repository for lectures, homework problems, and other educational resources. What differentiates it from other educational databases such as Merlot is that it is targeted explicitly at fine-grained course materials—problems, lectures, or labs that can be “dropped into” existing courses. Instructors in two fields—computer architecture and object technology—were solicited to submit their work for inclusion in the database. An application was developed to automatically download material from the Web or from e-mail into the database. Accounts were offered to the contributors and to others that allowed them to do a fulltext search of the database for materials on desired topics. Although it has been a time-consuming task to induce instructors to donate their material, we have developed a community of several dozen contributors. We have designed the software to make it easy to develop future databases: A targeted Web search identifies likely contributors, and the system generates request e-mail. Currently, we have databases for two areas of computer science: The Computer Architecture Course Database contains 880 items, and the Object Technology Database contains about 450. Together the databases have about 170 users.

There will always be substantially more course material on the Web for any field than we are able to incorporate into our database. To provide access to a wider variety of material, we have extended our database with a search engine that can search the Web for items containing the same terms at the same time as it is retrieving problems from our 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. This was replaced later with a Java application that could take a Web page and break apart questions according to user-specified criteria, with an option for a user to make manual corrections when necessary. For a “back end” to our database, we initially used WebAssign, an Web-based testing and assessment system developed at NCSU. However, WebAssign had a lot of functionality that we were not using, and our users were frequently bewildered by all the choices presented through the user interface. This led us to substitute a MySQL database for the WebAssign database, and create our own UI using Java. Not only is this easier to navigate; it also returns search results much more quickly than WebAssign.

The following sections will guide the reader through the various functions provided by the Course Database. The first step in using the database is to obtain an account. This can be done by sending e-mail to the author of this paper. You should provide some evidence of faculty status, e.g., the URL of a list of faculty in your department, or of the on-line directory of your university. You will then be given an account that will permit you to view all of the questions and answers, modify questions that you have authored, and insert new questions into the database.

2. Logging in to the Course Database

Once you’ve obtained a user-ID and password, you can then proceed to access the database. Do so by following these steps:

- Browse to http://cd.csc.ncsu.edu. You will be presented with the screen below.
Figure 1. Login page

- Enter your username
- Enter your assigned password
- Click on “Login”.

Once logged in, a user selects a database from a list presented. In general, instructors are granted access to the database(s) in the areas in which they are teaching.

Figure 2. Selection page

The “administrative tools” option appears only if the user is also an administrator. An administrator has the right to add new users and configure accounts. Once inside a database, the user may search for questions, edit an existing question that (s)he authored, or create a new question.

3. Searching the Course Database

The database offers strong searching capabilities, which allow users to perform fulltext searches. For example, you can search for a string (say “processors”) in all the questions in the database, or you can search for a question written by a particular author or from a certain textbook.
You can search through the text of the questions in the database by entering a word (or words) in the question field (e.g., type "cache" for all questions that include the...
word "cache" in the question statement).

- You can also search the text of the answers in the database by entering a word (or words) in the solution field. Note that only about half the questions in the database have answers at present.

- You can specify additional search options like author name (to search for a question contributed by a certain instructor), or textbook (the textbook from which the question is taken,\(^1\) or the textbook used by the author of the question).

- Click on Search to run the search query.

- When the search results appear, you have the option to View the question or Edit the question (you can only edit questions that you have authored).

When viewing the list of search results, you can click on the link to a particular question in order to view the question, duplicate it (for later editing), or edit it (if you are the author of the question).

---

\(^1\) The publishers of two computer-architecture textbooks have given us permission to include the questions from these textbooks.
You have measured the "ideal" CPI of a program to be 0.75 (supercellular issue allows CPUs to be below 1). However, this doesn't include the memory system. Assume that cache hits cost nothing and that cache misses cost 50 cycles. If instructions miss 2% of the time and data references miss 5% of the time, what is the overall average CPI? Assume the instruction frequencies shown in Figure 1.17 in this text.

Solution:

Figure 6. Viewing a question

Figure 7. Editing a question
When viewing a question, you will see a link to the "Original source" of a problem. This link is provided so that you can access the problem as it appeared in the document from which it was downloaded. This is useful if there is a problem with the formatting of the question in the database, or if the question refers to a previous question in the problem set from which it was taken. At this point, not all problems have "Original source" links, and not all of the links work (because the Web page containing the problems has moved or been removed). We are working on a mechanism that will create a local copy of the original source document for use if the original becomes unavailable.

When editing a question, remember to save your changes so the question will be updated in the database. Do this by clicking on the Save button. If you have edited the question and want to create a new question rather than overwrite the original question, click on the Save Copy button.
4. Creating a new question

Although we expect that most authors will choose to give us access to their Web pages, or e-mail questions to us, it is also possible for authors to directly enter questions into the database. Any user can do this. It is a relatively simple matter if you have the question in a machine-readable format. Figure 8 shows the Create Question form, which is similar to the Edit Question form.

In order to create a question, do the following.

- Enter your question in the question field as ASCII text or HTML.
- If a solution is available, enter it in the answer field as normal text or HTML.
- If the question is taken from a textbook, select the textbook from the Textbook dropbox. You can contact the system administrator to add your textbook to the list.
- Click on Save to enter the question in the database. You will receive a confirmation when the question is saved.

5. The downloading application

To insert questions in the database rapidly, a downloading application was developed to take pages from the Web and split them apart into individual questions, then load them into the database. The main function page of the downloading application is shown below.

![Figure 9. Main function page](Image)

*Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*  8

Copyright © 2003, American Society for Engineering Education
Since each question must be associated with a particular author, a page is provided to enter information about the author. This screen is shown in Figure 10.

![Create-instructor page](image1)

**Figure 10.** Create-instructor page

![Selecting a database into which to download questions](image2)

**Figure 11.** Selecting a database into which to download questions

*Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*

*Copyright © 2003, American Society for Engineering Education*
Once an instructor has been created, pages authored by that instructor may be downloaded.

The downloading application will then proceed to separate the page into questions according to the rule selected. Rules are specified by regular expressions, which may, for example, have the effect of starting a new question each time a paragraph begins with the word “Problem”. If the separation is not correct, the user is allowed to merge two adjacent questions or enter manual separation characters into existing questions that will cause them to be divided into separate questions.

Figure 12. Questions automatically separated

6. How to find contributors

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 31 (16) of the instructors in CA (OT) agreed to let us download their material, 7 (5) refused, and 35 (19) did not reply. An important facet of the database is that contributors grant all registered database users an unlimited right to reuse their materials in the users’ own courses. Reuse rights do not extend to publication in any other form, however. Because of this, the database materials are an example of open courseware, and stand in contrast to the proprietary databases being developed by publishers. Because it is a database of small units of

Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition 10
Copyright © 2003, American Society for Engineering Education
materials (individual problems and programming exercises), it is quite different from the electronic library projects like Merlot [1] and (the larger part of) the Computer Science Teaching Center (CSTC) [2]. Our intent is to provide instructors with material they can drop into their existing courses and modules to enhance them, without the substantial effort that is required to integrate a new module or textbook.

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 [3, 4], we contacted people listed on the WWW Computer Architecture Web site [5]. For the object technology database [6], we used Steve Beaty’s list [7] of computer-science course Web sites. We also used the list of course Web sites that the primary author generated for the 1999 OOPSLA Educators’ Symposium [8].
- We used attendee lists from several specialized education workshops.
- We have used Web search engines to find course Web sites 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.

The reasons instructors gave for turning down our request were about equally divided between two factors.

- **Copyright concerns.** Some instructors wanted to use their material in books they planned to write, and were afraid that making it available in advance might compromise their ability to do this. Other instructors had taken material from existing texts, and were concerned that they would have to seek permission from the copyright holders.

- **Diffidence.** Several instructors felt that their material was not polished enough to be used in the database, either because they were teaching a course for the first time, or because they had not been able to devote adequate attention to it. This is a common concern expressed by instructors regarding their course material [9]. Unfortunately, acting on that concern tends to compound the problem, since if material is not made available, others will not be able to help improve it.

We addressed the copyright issue by posting the following disclaimer on the login page of the database.

“The authors of the material in this database grant authorized users of this database permission to reuse it for educational purposes in their own courses. Any further republication of the material requires the express consent of each author whose intellectual property is being reused.”
The author of the material is also required to certify that the work (s)he is submitting is original. This is an advantage of using a database instead of a specialized search engine, which would have no easy way to keep track of copyright.

The diffidence problem has been attacked by combining the database with a search engine, as described below.

Another problem we faced while downloading course questions is the painful task of associating solutions with homeworks. Not all instructors publish their solutions and if they do they may not put them on the Web (another disadvantage of relying on the search-engine approach). Right now, for example, out of about 1000 questions in computer architecture, 590 have solutions.

7. Search Engines

Our experience in seeking permission to download material to the database makes it clear that we will never succeed in loading all of the relevant material. Therefore, it seems appropriate to combine the database with a search engine. Users of the database do not, of course, have an automatic right to reuse and adapt material that is not in the database; however, they are able to ask the copyright holders for permission individually. Although we have no experience yet, we expect that an author might well grant a single instructor permission to reuse material, even if it is not deemed ready for wider dissemination.

The search engine and database each have advantages and disadvantages.

- A search engine is easy to program to find a large amount of material, while it is time consuming to build a database.
- A search engine is capable of providing access to material only as long as it stays on the Web (which is frequently only until the end of the current academic term). A database retains material permanently.
- A search engine is harder to use than a database. It will inevitably return many pages that are not reusable course material.
- If material is updated, e.g., to remove bugs, a search engine will always find the most up-to-date material. A database will find it only if the database has been updated since the material changed.

The approach we have used to building a search engine is as follows:

- Program an existing search engine to find course Web sites by selecting keywords from course syllabi in a particular area, and having the engine return links to pages containing those links. NorthernLight has been chosen, because it can be targeted to search only academic sites in a number of countries.
- Search those course Web sites by using terms that are typical of homework problems, lab exercises, etc.
- On each search, give the author the option of searching only the database, or the database and search-engine results.
We expect that authors who fear copyright violations would be more likely to give permission to a single user to use a single item than to grant blanket permission to all users of the database. Eventually this technology could provide a fast way to bring up a database of materials in a new academic field: The search engine bootstraps a new database full of links; this is advertised to instructors who find it useful and contribute their own materials. Shown below is the result of a Web search with our specialized engine for the same query that produced the database listing in Figure 5.

![Search results for 'cache' - Microsoft Internet Explorer](image)

**Figure 13.** Results of targeted Web search

8. Conclusion

Our course database engine puts an abundance of course material at the hands of any faculty member. Materials can easily be retrieved from the database in HTML or text format and inserted into exams or problem sets. We especially encourage the submission of new material. Simply e-mail the material, or its URL(s) to the authors of this paper. We hope to provide a very useful service to the computer-architecture and object-technology communities, and any other community that would like to develop a database of its own using our software.
ACKNOWLEDGMENTS

This work was supported by the National Science Foundation through grant number DUE-9950318. It was previously supported by the NCSU Provost’s office through an Instructional Grant. Additional support was received from the Electrical and Computer Engineering department in Fall 1997. Many students have contributed to it on independent-study projects and research assistantships, including Andy Goldstein, Tony Louca, Hassan Shehab, Xiaokang Sang, Ana Goulart, Chennao Geng, Leonard Kishore, and David Steffy.

BIBLIOGRAPHY


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.