

Oberlin College

Enhancement of the Introductory Computer Science Curriculum

Principal Investigators:

Rhys Price Jones
K. Fritz Ruehr
L. Richard M. Salter

1. Overview of Goals and Objectives

An emphasis on derivation, analysis, and correctness verification throughout the problem-solving and program-development processes is the main thrust of the project. Interactive web-based laboratories are the most obvious of the curricular materials developed to date. They support sophomore level courses covering the traditional content of courses in data structures, algorithms and programming language concepts. In order to develop the hundreds of web pages necessary for the laboratories, the project team found it necessary to create an HTML compiler called HtX. In the same way as LaTeX can abstract the style properties of static documents, HtX abstracts both static and dynamic features of web documents, enabling rapid development, multi-authoring and consistent appearance of large web structures. We are now in a position to disseminate not only the course materials, but also the powerful tools that will enable educators everywhere to produce effective classroom presentations and laboratory materials, concentrating only on the content of the documents and leaving style and format considerations for automatic treatment.

2. Current Status and Accomplishments

The project has run its three-year course, but more work, especially on evaluation and dissemination is continuing thanks to a supplement to the original grant. A large body of lab-based materials is available for access and/or download. The labs can be reached by following some fairly obvious pointers from <http://www.cs.oberlin.edu>. The HtX tool and supporting documentation and sample style files are also available for download at <http://www.cs.oberlin.edu/htx> in both Unix and Windows versions. The Macintosh version is experiencing some delays, but work is being done to get it up and available for distribution.

3. Plans for Remainder of Project

Rhys Price Jones is working on a book to support use of our curricular materials at other institutions. A sample chapter, table of contents and a preface indicating its relationship to existing books is almost ready to be circulated to potential publishers.

Work is in progress to create an "exportable set" of lab materials. Realizing that not every school will wish to teach the same courses as Oberlin, we are preparing a distribution version of our labs with smaller units of packaging so that other institutions will be able to more readily accept parts of our developments for their own uses.

4. Materials that have been developed

Our Web-based laboratories comprise a hypertext suite of documents. They take full advantage of the capabilities of hyperlinks to provide instructional sequences that go well beyond those that textbooks or laboratory manuals can provide. For example, the labs make heavy use of a "question -- hint -- answer" idiom. Students are prompted with questions. They may choose to follow a "hint" link or, if they prefer, they may transfer directly to the solution. From there, they can transfer to associated material for review or for further in-depth study. A Java-based multiple choice question structure allows the lab authors to

provide a kaleidoscope of multimedia responses to the various choices that students may elect. Other links can start up programs for data plotting or take students to editors or interactive program environments.

Titles of the labs include:

- Object-oriented Programming
- Imperative Programming
- Combinatorics
- Streams
- Recursion, Induction and Analysis
- Higher-order Functions
- Search and Lookup Algorithms: Trees, Balancing,
- Hashing, etc. (2 labs)
- Interpreters (2 labs)
- Dynamic Programming
- Heaps and Priority Queues
- Sorting Algorithms
- Graphs
- Combinators and Graph Reduction
- Parameter Passing
- Pattern Matching
- Coding (for compression and security)
- Analysis via recurrence relations
- More Graphs and Greedy Algorithms
- Abstracting Control and Continuations
- Coroutines

The creation of an authoring tool grew out of a need to effectively create and support a large repository of useful Web-based materials in the face of a rapidly changing network environment. Our model is based on LaTeX [2], a very powerful document preparation system whose strength lies in its ability to separate form and content. Extending this principle to include the dynamic structure of a set of highly interlinked pages along with conventional notions of structure that apply to individual pages, we devised an extensible environment capable of expressing both aspects as part of a uniform style. Devising a "laboratory style" independent of individual laboratory units has permitted authors the freedom to concentrate entirely on laboratory content while retaining access to a high level of hyperlink functionality. The main authoring tool, called HtX, can enforce a standard document style, so that students quickly become accustomed to operating in the lab environment. It is thus the case that both authors and students can focus on content rather than structure.

The actual collection of HTML documents which constitute a complete system are generated automatically by running the HtX processor on the source file. The power provided by this separation of style from content becomes apparent whenever changes are made to either aspect of the document: if content is changed, the processor need only be run again to produce a new and updated HTML document system, with all relevant details, such as re-numbering of document components, revision of file structure, etc., handled automatically.

More significantly, if changes in the style of presentation of the material are desired, for example to "freshen up" the look and feel, or to take advantage of new technological developments in the underlying HTML language, then only the style file needs to be modified. Merely re-running the HtX tool in a new environment can produce an upgraded result that takes advantage of new developments in Web technology. Alternatively, multiple style files may be used to provide different "views" of the same document contents.

The ready adaptability of our tools to the evolving web standards allow us to easily maintain and revise our work, thus greatly increasing its "shelf life". By continually extending the expressive power of our authoring environment to include models for managing new Internet technologies, we gain a platform where new idioms based on such techniques may be developed and analyzed for their effectiveness.

5. Dissemination Activities

All our materials are available on the Web. This has been advertised at several meetings and workshops, including

- the 1996 Principles of Programming Languages conference (and its associated Scheme workshop),
- the 1996 SIGCSE (and a related workshop and seminar presentation),
- the 1996 summer meeting of LACS (Liberal Arts Computer Consortium),
- the 1996 Scheme workshops held at Indiana University in conjunction with their E.I. grant.

Additionally, individuals have been directly contacted who are now using HtX at Indiana University and NorthEastern University.

6. Evaluation Activities

Rhys Price Jones has been working with Nancy Baxter Hastings at Dickinson College, adapting some of the evaluation tools she has developed in conjunction with Dickinson's NSF-funded project to teach calculus in laboratories. These tools are being applied at Oberlin to assess student attitude changes. Other, standard, evaluation tools have been applied and their results are being investigated.

7. Benefits Seen and Expected

There has been a positive impact on students: In the two years that these new laboratories have been in use, there has been a significant increase in students' conceptual abilities and programming skills. We attribute much of this to such features as

- Engaging presentation of materials
- Increased instructional and social support
- Greater exposure to sample programs
- Opportunities for peer interaction

Just over 40 students have taken the newly developed courses. Somewhat over 20 are currently enrolled. As a result, any observations we make at this point are largely anecdotal. This is especially true when comparing participation of under-represented groups, since they prima facie form a smaller group within an already small population. In addition, our curricular developments have occurred in Sophomore courses. We need to transpose our methods to truly introductory courses to assess how well our approach can produce a leveling effect on incoming students.

There has been an increase in enrollment by women. More significantly however, the retention rate of women in the sophomore courses has increased strikingly, and the success rate of women (measured by their relative class rankings) has increased even more dramatically. Indeed, in the Fall of 1995, four out of five of the top-scoring students were women. The top student in the class was an African American woman. It is reasonable to expect that as word of their success spreads, we will see an even greater increase in Sophomore course enrollment by women.

Students have served as laboratory assistants and as tutors, and have also been recruited to write laboratory materials. They can do this as soon as they have completed one of our courses. Their writing is often refreshing and attractive to new students. Students who have just completed the course have a good sense for where explanations are needed and where material can be trimmed. A testament to the ease of use of HtX, students are usually comfortably working with the tool and source files within a day. Recently, two of our student assistants traveled to ACM's Computing Week to demonstrate some of the materials they developed.

8. Major Benefit: A More Level Playing Field

One of our goals is to make the concepts presented relatively more accessible to traditionally under-represented students by using an approach that is less familiar to the students who have had greater prior exposure to traditional programming, a group typically white and male. "Hacking" skills are of minimal

utility in an environment which emphasizes scientific derivation of algorithms and skillful development of programs through verifiable sequences.

Our labs provide a supportive environment. On average, male students seem more willing to tolerate the isolation and consequent frustrations associated with more traditional approaches to the development of conceptual and programming skills. The socially engaging and supportive environment of structured laboratory and tutoring sessions provides all students with a better opportunity to learn new ideas and techniques.

We hope to see increased visibility of women as role models. Female student assistants and tutors act as role models and expand all students' expectations. We hope to see the effects of this in the coming year.