

Brooklyn College of the City University of New York

Distributed Processing
Across the Undergraduate Computer Science Curriculum

Principal Investigators:

David Arnow Dept. of Computer and Information Science
Carol Tretkoff Dept. of Computer and Information Science
Paula Whitlock Dept. of Computer and Information Science

1. Overview of Goals and Objectives

The objective of this project is to develop material (syllabi, writeups, labs, processing at every level of the undergraduate CS curriculum. The goal is to this in a way that minimizes displacement of other vital topics but at the same time guarantees a thorough preparation of CS undergraduates in

The strategy that we have chosen is to construct, in a number of fundamental and universally or at least widely taught CS courses, modules that advance the student's understanding of distributed processing and that are, at the same time, not only appropriate for but beneficial to the "host" course. That is, when a module is added to a course, it must not, if possible, displace an existing topic, but rather provide a distributed programming context in which to address a particular topic.

In addition, we planned to design and implement a cluster of advanced courses that make extensive use of distributed programming. One of these would serve as a "capstone" course, appropriate for seniors, that would build on and advance the distributed programming curriculum encountered in the rest of our courses. The capstone course also serves as a measure of the effectiveness of the module approach--presumably students entering that course will be so well prepared in this area that the course can approach the topic with some depth.

Our hope is to develop a body of material that not only transforms our curriculum but that is also easily transferable to other institutions.

2. Current Status and Accomplishments: The Original Plan

The table 1 summarizes the current status and accomplishments of the project, at the end of its first year.

The vagaries of departmental teaching schedules contributed to the discrepancy between planned completion dates and actual ones. Note that the cohorts of students taking the advanced courses that were introduced in the spring of 1996 did so without the benefit of prior modules. These courses will have to ramped up when cohorts with experience from earlier courses enter them.

The greatest omission from the plan for the first year was the absence of local faculty development workshops. Although through informal means, several other faculty have been brought into the effort, approximately 1/2 of the dept. is not yet involved. This will have to be corrected in the second year (the current year).

Table 1. Summary of the Current Status

| Course and/or Module students) | Planned (Year of Completion) | Designed (material available) | Approved (by dept. curr. comm.) | Implemented (taught to |
|--------------------------------------|---------------------------------|----------------------------------|------------------------------------|---------------------------|
| CS1: Clients & Servers | 1 | Yes | Approved | S97? |
| CS2: Clients & Servers | 1 | Yes | N.A. | |
| F95,S96,F96 | | | | |

| | | | | |
|---|---|---------|----------|---------|
| CS2: Parallel/Distribut. | 1 | Yes | N.A. | F96 |
| DM: Problem Sets | 1 | No | N.A. | ?? |
| Shell Programming: Problem Sets | 1 | Yes | N.A. | S96,F96 |
| Programming Languages: Distributed Languages | 1 | in prep | N.A. | S97 |
| Operating Systems: Network Support | 1 | in prep | N.A. | S97 |
| Workstation Programming | 2 | Yes | N.A. | S96 |
| Simulation: Distributed Simulation | 2 | Yes | N.A. | S96 |
| AI: Distributed- Parallel Searches | 2 | Yes | N.A. | S96 |
| Decision Support Software | 2 | No | N.A. | S97 |
| Capstone Course | 2 | Yes | Approved | S97 |

3. Current Status and Accomplishments: The Java Wild Card

One unanticipated development for us was the introduction of Java into the CS community. (At the time of writing our proposal, early 1995, Java was in early Alpha form.) Now it seems that, for all its faults, it is likely to become a widely-used CS instruction language. Since it directly supports many aspects of network and distributed computing, departments that are eager to make distributed programming a priority are especially likely to use Java as a teaching language.

In recognition of this, we have opted to introduce a Java-based CS1 course in the spring of 1997 on an experimental basis. Although it is possible to teach Java and ignore network and distributed programming issues altogether, obviously we will not take that route.

The course has been approved by our department and substantial amounts of course material are currently being prepared. We hope that this material too can be exported to other departments. We should note that we recognize that many departments will never use Java as a teaching language and that the need for constructing a set of modules, as we had planned, remains and we will continue with that approach as well.

4. Plans for Remainder of Project

This year we aim to fill in the gaps of the above table, place our module materials on the web, and develop a hardcopy distributed programming instruction manual. We also will hold a local faculty development seminar for Brooklyn College and CUNY faculty.

In the following year (year 3), we will organize a national workshop, perhaps in collaboration with other NSF curriculum developers in closely related areas (such as parallel computing). We will of course submit a summary of the experiences and outcome of this project to the 1998 SIGCSE technical symposium as a paper-- perhaps we will also be able to offer a tutorial.

5. Materials That Have Been Developed

Special software has been developed for use in several courses. For the lower courses, the purpose of this software is to provide easier access to network services. This has been done on the PC for CS1 and on the SunOS and Solaris environments for CS2. We have also developed support software for the Shell course, since the shell does not directly support network computing.

Instructor notes, exercises and handouts for students have been developed for all of the courses that have already been offered and that are mentioned on the table above.

6. Dissemination Activities

We have presented a short paper at the 1996 Northeastern Small College Computing Conference, describing our project and its progress.

One of us, in collaboration with another colleague, presented a short paper at the SIGCSE/SIGCUE meeting last June in, describing the software that we have developed to aid in the evaluation and support of student projects. This software is quite important for running OS labs in which students write simple OS network software. It is also a fundamental tool in our programming courses.

7. Evaluation Activities

We have not yet undertaken any evaluation. We don't anticipate doing so until the third year, when we will begin to see students who have passed through this entire curriculum.

8. Benefits Seen and Expected

Thus, the benefits we have seen so far are strictly anecdotal. We sense a great eagerness on the part of the student body-- this is undoubtedly due to the excitement that anything associated with the web seems to generate these days. We also are witnessing what seems to be an increase in the number of students who seek to meet their department "project course" requirement but undertaking one related to network or distributed computing.

Even without any formal development seminars, several of our colleagues have gone out of their way to contribute to the aims of this project. An teacher of the operating systems course for example went to great lengths to incorporate a discussion of threads in his course, in anticipation of its relevance to our capstone course.