1 List of Publication References

- K. Maly, D. Ray, C. Wild, I. Levinstein, S. Olariu, C.M. Overstreet, N. Rao, T. Ireland, and G. Kantsios. Computer productivity initiative. *Proceedings of the Seventh SEI Conference on Software Engineering Education*, San Antonio, Texas: 441-459, January 1994.
- C Wild, K. Maly, C. Zhang, D. Ray, I. Levinstein, S. Olariu, C. M. Overstreet, N. Rao, D. Rosca, T. Taylor, T. Ireland, and D. Sibol. A decision-based hyper-multimedia case environment for the computer productivity initative, a large undergraduate project. *Proceedings of the World Conference on Educational Multimedia and Hypermedia*, Vancouver, Canada: 691, June 1994.
- K. Maly, C.M Overstreet, H. Abdel-Wahab, A Gupta. Melding television, networking, and computing for interactive remote instruction: exploiting potentials. *Proceedings of the World Conference on Educational Multimedia and Hypermedia*, Vancouver, Canada: 367-372, June 1994.
- K. Maly, D. Ray, C. Wild, I. Levinstein, S. Olariu, C.M. Overstreet, N. Rao, D. Sibol, and G. Panayides. Computer productivity initiative: past, present and future. *Proceedings of the Eighth SEI Conference on Software Engineering Education*, March 1995.
- K. Maly, D. Ray, C. Wild, I. Levinstein, S. Olariu, C.M. Overstreet, N. Rao, D. Sibol, and G. Panayides. CPI: CS undergraduate curriculum enhancement through multi-year coordinated project. Submitted for the *Twenty-Sixth Annual SIGCSE Technical Symposium*, March 1995.

2 Introduction

The current Computer Science curriculum must be modified to help students better understand how to apply their education to real world problems. The primary objective of the Computer Productivity Initiative (CPI) is to identify and develop these modifications, portable to any CS program. A multi-year, coordinated project is introduced into the curriculum. Students from several courses coordinate and share information concerning the broad, ill-structured project topic/problem. The courses involved in the CPI project are linked through the tasking of one class by another. The tasking class is always one of the senior CPI courses, while the tasked classes include a sophomore and a junior level course. The two new senior level courses focus on such topics as technical research, market research, presentation skills, group collaboration, interviews, budgeting, proposal writing, presentation tools, scheduling, hardware availability research, system architectural design, requirements specification, simulation, prototyping, and cost estimation. Initial feedback from our students, potential employers and an external board of advisors confirms both that the most successful graduates may not be those with the best technical education and that this effort can provide that additional dimension to the traditional CS curriculum which better prepares students to contribute to the solution of ill-structured but real problems.

3 Approach

The Computer Productivity Initiative (CPI) introduces a multi-year, coordinated project into the curriculum. Students from several courses coordinate and share information concerning the broad, ill-structured project topic/problem. The courses involved in the CPI project are linked through the tasking of one class by another. The tasking class is always one of the senior CPI courses, while the tasked classes include a sophomore and a junior level course.

The senior level laboratory courses are the vehicles for ensuring that the primary CPI objectives are achieved. The sequence of CPI senior level laboratories provides students with a clear view of applying the many theories and concepts illustrated and learned in previous courses. It is at this level that all of these knowledge-based skills are applied. Students are faced with new problems that did not appear in their previous courses. They are tasked to develop a solution to a typical problem of increasing the productivity of some system. The problem is the CPI project and is presented as an un-structured requirement. The students must develop the problem solution taking all product-related concerns into account in their decision-making process. In this context, the students participate as leaders of small groups, managers of a larger group, and support technicians and professionals. They function essentially as project managers, defining the problem, developing solutions, and evaluating alternatives for feasible solutions. They must present formal documents for proposing project continuation and produce specifications suitable to be used for production development.

The senior level laboratory students task lower level courses with software programming requirements, prototype experimentation, impact studies, problem solution plans and implementation algorithms. They evaluate the products from the lower level courses and use the information to support their ongoing efforts in developing problem solution specifications. The laboratory course material provides students with a guided set of methods to meet the course requirements. It is in the application of these methods that students develop the major productivity improvements. Their laboratory requirements are similar to the requirements of productive computer scientists in their first few years after graduation. Their laboratory experiences will virtually qualify them as assistant project managers who are knowledgeable in most of the facets of applying the technical computer science knowledge to actual problems either in industry or in advanced graduate studies. They will have explored the fields of planning and scheduling, resource allocation and budgeting, to the incorporation of computer hardware and software systems to produce the specifications for a tangible product that society needs and will buy.

The selected CPI project is carried out over the period of one year (or more years if the problem is sufficiently large) in various courses. Two main courses at senior level deal with the generation of a solution to the project; this step involves many issues not traditionally considered in software development for university course projects. The goal of this initiative is not to develop commercially acceptable systems but rather to develop system specifications that solve a problem. The specifications include a detailed design description, the expected functional performance, cost budget analysis, and a production plan based on analyses, simulations, and prototyping. Ideally the product should be such that it could be submitted as an SBIR grant proposal. The project selected as the CPI project will require a solution that consists of not just computer software or hardware, but requires the design of a system where a computer is only one solution factor albeit an important one. The CPI project of expanding the use of the family television as an interface for other family activities meets these requirements. The LIFE system as described by the initial 10 students of

CS 410 is a challenging project that defines the hardware and software modules that support many aspects of family life using the family television as a graphic display and communication device.

Since the CPI project theme will have been used in several supporting courses, computer science students will have the sense of participating in a computer productivity project for several years. By the time the student is presented with the entire task of the project, the majority of the basic computer science material has been seen in various courses. The senior students will then be able to readily integrate material of previous core courses into a single project solution at the senior level. It is the early piecemeal introduction of the CPI project that provides an increased productivity for the senior students, an efficient design and description of the problem solution that is the gathering of many of the pieces the students have seen in previous courses.

For most students, their project experiences will involve the same basic project. Developing an entirely new project each year imposes an excessive burden on the faculty. Therefore it is essential that projects be reusable. Reuse of projects can take two forms: the same project is repeated from year to year, as is often the case in engineering senior projects; or the original project is refined, revised, and improved from one year to the next. Our method combines the two approaches because we wanted every student to have the initial "blue sky" development experience when presented with an open ended problem area, but we wished to incorporate the best of the previous classes' work. The seniors will initially approach the problem area in a blue-sky fashion, but the instructor will introduce worthy materials to supplement their work after they have achieved a definition of their project goals.

We are convinced that the single theme project introduced and incorporated in courses throughout the curriculum will enhance the student's ability to rapidly grasp the concepts of the new course. At the senior level laboratories the students will see the total picture of the CPI project. Because of their previous knowledge of the project theme, they will not require the "learning curve" time to begin active project prosecution and specification development. They will also likely reflect on tasks they received at the lower level courses and use that knowledge in generating similar requirements to support their laboratory requirements. This integrated and coordinated project theme across the course boundaries is the CPI innovative concept that we believe will solve many of the problems with our curriculum of today.

The courses involved in the CPI project are linked through the tasking of one class by another. The tasking class is always one of the Senior CPI courses. The tasked classes are the Sophomore level course (Problem-solving and programming) and the Junior level course (Computers and Society). After some experience, we decided that tasking must be

- initiated early in the semester
- responded to well before the end of the same semester
- explained by the Seniors to the lower level course
- actively managed by the seniors
- be in line with the objectives of the lower level course

Our aims are to broaden the vision of the students in the lower level courses so that they understand that their activities support a larger enterprise and to give the seniors some experience of managing others.

The CPI modifications to the lower level courses (i.e. CS 250 and CS 300U) do not alter the course structures or the theory or skill development of the courses. The modifications affect the subjects and general theme of the student activities that are assigned during the normal concept-learning within each of the courses. This is viewed as a critical factor in that curriculum-evaluating bodies must not see substantive changes in these core courses that may affect certification decisions as result of CPI incorporation. Fundamentally, we believe that the installation of CPI must not adversely impact the scope of theory, concepts, and knowledge of existing courses.

3.1 Modifications to Problem Solving Course

Tasking of the Sophomores by the Seniors

The fundamental modifications of CS 250 include the specific use of the CPI project as the theme for programming assignments and course examples all at the sophomore level. Prototypes are given to the class to use in performing experiments and the results from the experiment are reported back to the senior courses. The critical difficulties expected with these modifications is the availability of problem definition from the senior level productivity laboratories early enough for use in the sophomore course. The CPI coordinator functions as the expediter to insure sufficient statement definitions are described in CS 410 and CS 411 suitable for CS 250 use. While the flow of information from CS 410 is critical for the achievement of CS 250 course requirements, the reverse flow of product definition is expected to provide direct support to the CS 410 laboratory efforts.

We emphasize that Computer Science in general and software engineering in particular are experimental disciplines. Consequently, our Problem Solving and Programming course contains a module on scientific method. The instructor introduces students to the the making and testing of hypotheses. Some kind of software gadget is provided to students who work in groups. They are required to formulate a hypothesis related to this gadget and design an experiment to test the hypothesis. Prior to running the experiment, their hypotheses and designs are critiqued and revised. After the experiments are run, the groups write them up and turn in their reports and data for grading.

This module is the point of connection with the seniors in the CPI course. The seniors decide, early in the semester, on an area in which the sophomores should conduct research. Two or three seniors introduce the CPI project to the sophomores and explain how the area of research fits into the whole project. The gadget which has been developed by the seniors and the TA for the CPI project is also explained. During the course of this assignment, the seniors are available to the sophomores for consultation on their research. Once the proposed hypotheses and experimental designs are submitted, the seniors (all of them, not merely the liaisons) are responsible for critiquing and evaluating them. After the experiments have been run, the reports are graded by the sophomores' instructor and made available to the seniors.

Our intention is that the gadget provided involve some aspect of the user interface. This is so for three reasons. In the first place, the user interface is often a useful standpoint from which to imagine the functioning of a system. Both when students are trying to develop a proposal (the first semester of CPI) and when students are fleshing out the first semester's design, user interface issues will arise. In the second place, the information to be gained from the sophomores will be useful but not essential to the project. The sophomores' research will involve ease of use, training time, arrangement of functions, interface metaphor and the like but is unlikely to affect the basic functionality of the product. Finally, the user interface is convenient from the point of view of providing the gadgets involved. There is a wide variety of interface objects already available for

Xwindows systems and those that are created for the tasking assignment are likely to be reused.

3.2 Modifications to Computers in Society Course

CS 300U (Computers in Society) currently requires two major forms of student class activities: a formal course paper and student presentations of topics in a forum environment. To utilize this course and the student presentations in the large CPI project, the topics of both the paper and the forum issues will be focused on such critical factors as:

Personal security	Energy sources	Expected life span	Occupations
Education	Personal Privacy	Food supply and type	Government
Transportation	Health and medicine	Economics	Labor supply
General tools	Communication	Business finances	

The subject requirements are products of the ongoing CS 410 and CS 411 courses. The CPI coordinator will function as the expediter to insure that the required topics and issues are identified and provided in a timely fashion suitable for use in CS 300U. The results of these taskings are critical components of the senior level courses. The flow of information with the junior level courses is critical in both directions for the success of the CPI project.

Senior students initially seem enthusiastic about the prospect of tasking CS 300U with research concerning their proposed topics. This enthusiasm wears thin as the necessity of precision of definition and the timing requirements begin to generate work for the seniors. When the results do come back from the tasked class, they are both later than desired and not at all what the seniors had expected. The CS 300U data tend to show a much broader perspective than the CS 410 and CS 411 students anticipated. While this broad perspective was planned in the overall design of the CPI concept, the presence of well supported negative viewpoints exceeded the original expectations of the CPI group.

In order to overcome some of the above mentioned problems, the current offerings CS 410 and CS 300U are more closely coupled. Students in CS 300U can elect to join the CS 410 project team. As a team member, they become aware of the nature of the project and the set of societal impact concerns regarding that project. Members of the CS 410 class act as mentors for the CS 300U students and act as questioners during the forum sessions.

3.3 Development of Computer-Based Productivity I

In order to achieve the broad set of goals established for CS 410 (Computer-Based Productivity I), the traditional class lecture has been augmented by guest speakers, class field trips, and individual student visits to corporations. Deciding on the tasking of the supporting classes added a new element to the requirements on the students. The infusion of project support equipment and computer application tools into the course expands student use of various systems including planning, development, presentation tools, and multi-media applications. Several formal presentations by each class member, with emphasis on professionalism, required students to become more involved in the direction of the class. The establishment of a self-organizing student structure with defined goals and responsibilities highlights the type activity that is expected of students after graduation.

The broad scope of topics and concepts presented in CS 410 critically emphasizes the fundamental instructor's need: know your subject. The broad perspective of topic material can easily consume the instructor's available time for course preparation. It is a challenge to budget the time to provide simple directions to students. The goal has been to lead students through the decision making and problem solving phases of the project, requiring the students to make the critical decisions essentially on their own. The requirement that the students succeed in developing a project solution requires a very close scrutiny on the part of the instructor. The approach of having students "learn by doing" remains as a superior method of providing the students with the necessary lifelong skills goal of CPI.

It has become clear that only with a relatively small class can the instructor fully comprehend how well the students are solving problems and employing the productivity tools made available to them. This teaching requisite cannot be delegated nor obtained without the close and conscientious vigilance which an instructor can provide.

Students entering the course generally have received no direct education in the methods and practices of preparing a technical presentation. The development of these lifelong skills is critical in the overall success of increased student productivity. Students must be provided a firm foundation in the effective methods of presenting. In addition to in-class presentations throughout the semester, the class ends with a major presentation to an external advisory board. The goal of CS 410 is to produce a proposal which could be submitted for a SBIR grant.

The introduction of multi-media and other leading-edge technologies has a mixed acceptance among students. Some see the technology applied to the solution of their problems, others view the technology as exciting, and still others appear somewhat ambivalent as to its use. By allowing much greater freedom of choice by students as to what set of technology they will take the time to learn and use effectively, a larger number of students benefit and much more productive work is accomplished by the class as a whole.

In almost every situation, the student learning curve greatly exceeded the expected value. Students have repeatedly overcome their lack of skills and met imposed course deadlines. In general, students have not just met the deadlines, but excelled in their performance. It would appear that instructors might routinely underestimate the students' capabilities.

3.4 Development of Computer-Based Productivity II

A key focus of CS 411 is to acquaint students with procedures and tools for evaluation and planning. This is done through their evaluation of the high level product specification completed by CS 410 students during the previous semester and preparing more detailed requirements and plans for further development of the product. Students also provide supporting materials to explain and justify all decisions which they make. As with CS 410, the class ends with a major presentation to an external advisory board, complete with demonstrations of prototypes and written back-up materials.

The major student activities planned for CS 411 included a review of the CS 410-prepared specifications, a risk analysis (with an emphasis in refining/improving the requirements to reduce risk of failure in the envisioned product), planning and scheduling (both short-term limited to the class activities for the semester, and long-term, for complete product development), product development cost estimation, preparation of a project management plan, and preparation of a verification and validation plan. While the results of students' risk analyses affect the choice of activities for the semester, the faculty have identified a specified set of activities which should be included in each offering of CS 411: we planned to introduce prototyping and simulation as basic tools to evaluate and refine specifications and to introduce students to existing document standards (such as those published by IEEE) for requirements or planning whenever appropriate. In addition, we intended to include a discussion on the design of computer-human interfaces since for many consumer-oriented products design of the user interface is critical to its success.

Not all of these topics were covered, primarily due to time constraints. Students did prepare individual risk analyses based on the CS 410-specification, prepared schedules for their semester-long activities, identified two components whose development seemed to entail the highest risk (or whose prototype might be impressive to the external board): a home grocery shopping system (which raises interesting user-interface issues) and an integrated telephone/television system which uses the TV to display videophone images. They used COCOMO to prepare cost estimates for the development of the software for these two components. This was based on a modified Delphi technique: students prepared individual high level designs for each component, agreed on a single design for each, and prepared individual size and factor estimates (as required by COCOMO) for each component, refined these into single consensus estimates, and used these as inputs to COCOMO. They also prepared an initial draft of the system specification using the format described in IEEE-830. Some of the draft came directly from the specification prepared the previous semester, particularly for hardware since this was not identified as a major risk area. Since IEEE-830 is a standard for software requirements rather than system requirements, additional sections were added to describe hardware requirements and the partition between hardware and software functionality. Students also prepared user's guides for each component as supplements to IEEE-830. Upon completion of experimentation with the prototypes, the draft systems specification was completed.

In order to get a quick initial evaluation of the basic functional and marketability of the home grocery shopping component, students conducted a focus group using adults who might be interested in such a system. The purpose was to get an initial reading on desirable functionality from people who might buy the product before time was invested in specifying and building a prototype. The purpose of the prototypes was to evaluate the proposed functionality for the two components; no effort was made to evaluate system performance. Thus the prototypes were used to evaluate the user interface and most key features of the functionality of the proposed system. Students specified the behavior of the prototypes before beginning their design and construction. These prototypes were later used by people assumed to have similar demographics to potential product customers, non-computer science students at Old Dominion University.

Several planned topics were not covered; no simulation due both to time constraints and the fact that the class identified no risks whose resolution was reasonable addressed by a simulation study. In addition no project management plan nor project validation and verification plan were developed.

Several lessons were learned through this initial offering of CS 411.

- The prototypes were well received by the board – better then "just a stack of reports." The user's manuals were also positively perceived by the board, apparently more than the IEEE-830 based specifications. But everyone, students and board members, like to see things that actually do something.
- Too many activities were scheduled. With more careful planning and scheduling, some more can be included, though scheduling will still be tight.
- Use of the focus group was effective, particularly for the general consumer-oriented product the students were working with. This gave the students early feedback about perceptions of

potential consumers. In addition, the depth of the study can be controlled based on available time.

- Our prototype evaluation was weak, but with forethought, can be easily improved. Each CS 411 student recruited three non-computer science students to evaluate each prototype. Evaluation included performing an assigned set of tasks while an observer (a class member) noted any difficulties and was available to answer questions if required. Each evaluator then completed a questionnaire. The problem with this approach was that the evaluators were often friends of the students and were usually not critical of what they had seen. More independent evaluators should be used.
- A strong point in the use of both focus groups and prototype evaluation is the direct interaction of computer science students with potential users.

4 Evaluation

We keep track of all problems encountered in the execution of CPI, their alternate solutions and the evaluation methods for the decisions in a database maintained by DHC. In this and the next section we shall address six major decisions and their justification. Our evaluation data is presented in Table 1.

(a) **Teaching CPI:** Based on the data collected using a control group our main conclusion is that there is a hardware component necessary to teach the senior courses but most schools should be able to handle it within their regular equipment budgets (the non-traditional equipment - video cards, TV, camcorders - is on the order of 10K). The software we use for presentation - Islands, Office - may also require additional funds if no equivalent is available in the department. All other software used in the course is available from us over the net without any cost. Based on the data collected from students, we were pleasantly surprised that students do not spend more time on the senior courses than in a normal project oriented course. We note that students indeed learned what we set out to teach them and that they liked the courses. That in itself does not mean that students actually benefit from CPI. We performed a survey of all students who completed the courses of CPI and asked them again if they benefited explicitly from and attributably to CPI. A large majority answered affirmatively. Secondly, we collected anedotal evidence of satisfaction by asking students, board members, and employers of our students. These are listed in Table 2.

(b) **Curriculum Requirements:** The three main alternatives we considered were to make the senior courses required, have them honor sections, or have students make proposals to enter the courses. After the first year offering we received a strong support from our industrial members on the board as to the marketability of our students. As a result, we felt compelled to make every effort make the courses required to ensure that all our students have the experience. The biggest obstacle was a concurrent effort by our University to reduce the number of credits necessary for professional degrees to 124 credits (with 120 credits preferred). Since CS250 and CS300 are already required we needed to add two courses without increasing our number of credits. We were able to do so by making one of our electives into the required CS410 and replaced a general education requirement with CS411.

	CPI				Comparison	
	CS250	CS300	CS410	CS411	CS390 (theory)	CS476 (project)
Enrollment/sem	45	60	8	8	50	8
#offered/year	2	2	1	1	1	1
Contact time (hrs/wk)	3 (class) 1 (lab)	3 (class)	3 (class) 1 (lab)	3 (class) 1 (lab)	3 (class)	3 (class)
Student effort (hrs/wk)	N/A	N/A	N/A	14	6	14
#TAs (20hrs/wk)	.5	.5	1	1	.4	.4
#speakers	1	3	3	3	0	0
Hardware	wkst	wkst PCs	4wkst 2PC TV+VCR 2camcds 2way video cards	4wkst 2PC TV+VCR 2camcds 2way video cards	wkst	wkst
Software	Unix/X Test environ.	Unix/X, Emacs WordPerfect, Word, etc.	Unix/X Island MS Office MS Project DHC Mosaic Emacs	Unix/X Island MS Office MS Project DHC Mosaic Emacs	Unix/X	Unix/X
Student satisf.[a]	4.8	4.2	5.8	4.8	2.8	5.7

[a] Student ratings range from Excellent (6) to Poor (1).

Table 1: Summary of Evaluation Data for CPI

Students' Comments

- This Course stands out in my college carrer as the most rewarding learning experience I have had. As you know, my interview was met with a wonderful (and lucrative) job offer. I feel that my success in that interview was a direct result of taking the CPI course.
  Kenneth J. Pratt '93
- Developed a student unity not present in any other CS course.
- Generated the pressures of real deadlines from a business sense and the knowledge of efforts required to meet deadlines.
- Great experience in group work. Very important in business use of CS knowledge.
- Significant increase of confidence in making presentations on technical subjects.
- Enhanced awareness of communication skills with other CS professionals.
- Great exercise in group interactions of technical personnel.
- Added immense personal value to my computer science knowledge.
- Provided valuable lessons in human interaction, creativity, and leadership.
- One of the best courses taken at the university.
- Learned valuable lessons in bringing order from chaos with people.

Employers' and Board Members' Comments

- In my opinion, all CS students should be prepared with practical skills as outlined in the CPI project. If it is implemented, CS Graduates at ODU would be six months to one year ahead of CS graduates from other institutions.
  Richard T. Cheng, Ph.D., Chairman and CEO ECI Systems and Engineering
- Very impressive professional looking presentation.
- Employers felt that this experience made students much more employable than CS students without the skills demonstrated and advised students to use the reports and slide show as their main selling point.

Table 2: List of comments by students, board members, and employers

(c) **Tasking:** This problem concerns the coordination necessary to integrate CS 250 and CS 300 with CS 410 and CS 411. The key issue was not how to task students in the lower level courses but on a timely delivery of the results of these tasks back to the seniors so that they can be incorporated into their decision process. In the first year, no effort was made to couple these courses. In the second year, the seniors used the material produced by students in CS 300 and CS 250 a semester earlier. In steady state, Fall 94, we have achieved a schedule which includes presentations of the tasks by seniors to students in CS 300 and CS 250 by the third week and results back from CS 250 by the twelfth week and from CS 300 results will come in a continuous stream starting in the sixth week. During that period individual seniors serve as expert help to these students and also they will attend foray held in CS 300 and provide a summary of the presentations there to the rest of the seniors.

(d) **Advanced Tools:** We decided to use the departmental network of workstations and a locally developed multimedia, hyperlinked system development environment (DHC) to maintain our project database. We were very much aware of the problem of the heavy learning curve of using sophisticated tools in a course. We made a special effort to tailor subsets of the tool environment for the course needs and provide self-guiding, multimedia tutorials on DHC, Island (a presentation tool) and the use of multimedia through TV and workstations interchangeably. The biggest addition in terms of advanced technology we made this year is to provide an interface to allow students to use PCs and their presentation software (we supply Microsoft Office). This decision is driven by the high cost of presentation tools specifically and multimedia software in general for Unix compared to Windows (or similar OS).

(e) **Project Selection:** Initially, we considered to be this a major problem but, as it turns out, it has become a non-issue. In the original proposal to NSF we had envisioned a rather complicated sequencing of the four courses. It was thought that we would actually have more than one project being worked on by different student groups at the same time. However, it soon became clear that there is an enormous investment in preparing the senior courses for a particular project.

5 Dissemination

In our various efforts to disseminate our curriculum we have made the, now obvious, discovery that in general other schools are not eager to accept major curriculum changes from without. In that we have found we are not alone. The major initiatives funded by NSF under this program have similarly failed in convincing a large number of schools not involved in the original proposal to accept whatever was developed. Clearly we have done what everybody else is doing: write papers, make presentation at appropriate conferences, put the material under Mosaic. Two efforts we consider worthwhile imitating are: one, workshops as held by Clemson University for their NSF project and two, inviting faculty for a sabbatical as is being done by UVA. A third method we are attempting is to make dissemination a part of another proposal we have made to NSF as part of the NIE initiative.

For the first two methods to work we would need some additional funding because we had not included them in our original proposal. The idea is to invite a number of faculty from four year colleges (the aversion to accept outside curricula is directly proportional to the reputation of the school) to a workshop on our Computer Productivity Initiative. At that workshop we would demonstrate our concept and show that we have a complete set of software, instructor manuals, course notes all in easily transported electronic form. A second step is then to invite faculty who are really committed to transport the curriculum for a one semester sabbatical to participate in the actual teaching of the courses. This way we would hope to create an initial set of four colleges over two years. This might be enough critical mass to convince others to join up simply by attending a yearly workshop where we would exchange experiences and improvements.

In the second approach, the curriculum is implicitly transferred to a set of thirty three de-facto institutions - the courses we plan to offer as part of the testbed include the Computer Productivity Initiative; to quote from our proposal:

> Building on its expertise in networking, computer supported collaborative work, and remote learning, Old Dominion University will build a statewide highly Interactive Remote Instruction (IRI) testbed based on networked workstations. Technology and pedagogical methods developed in the testbed will be integrated into TELETECHNET, a state funded program in which Old Dominion University and community colleges will provide four-year degree programs. The first two years will be taught at the community colleges with the last two being provided by Old Dominion University to remote community college campuses via networked and televised course instructions. Significant contributions of the proposed project include: 1) refining an existing IRI prototype to provide an open-ended teaching/learning environment with a set of basic features easy to use by students and faculty of all disciplines; 2) developing new instructional methods to exploit the interactivity of the system in support of higher domains of learning; 3) bringing together a consortium of government, industry and academic individuals who are dedicated to innovative approaches to education for the 21st Century; and 4) providing Virginia with a cost-effective method of delivering high quality education remotely in support of some of the 50,000 additional students expected to enroll in Virginia's public colleges and universities by the year 2001; and 5) serve as a model for other states faced with similar problems.

Appendix A: Supporting Technology

One of the objectives of the CPI effort was the utilization of productivity enhancing technology during the development of the CPI project. The software environment consists of off-the shelf packages as well as systems developed in house. For the UNIX environment, we purchased Island-Presents, a modern slide presentation package. For the PC-Windows environment we purchased Microsoft Office (includes a slide presentation package, word processor, spreadsheet and data base) and Microsoft Project. In addition we used DHC (Decision-Based Hypermedia Case) and the Software Experimenters LabBench developed in house. For dissemination purposes, we have placed a description of the CPI project on the internet, accessible through MOSAIC.