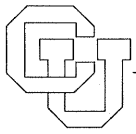
A Practicum – Seminar in Structured Programming

Clarence A. Ellis*

CU-CS-061-75 January 1975



University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

*This work was supported by the National Science Foundation under grant number GJ-660.

A PRACTICUM - SEMINAR IN STRUCTURED PROGRAMMING

ABSTRACT

This paper describes a practicum-seminar course which the author taught at the University of Colorado during the spring and summer of 1974. This course contained a number of salient features which made it an interesting experiment and a unique learning experience for all involved. The broad goals of the seminar were:

a) to introduce participants to system design and software engineering techniques and to provide practical experience in the use of these techniques;

b) to design and implement a flexible, human-oriented student registration and scheduling system.

This paper presents background and discussion of the project and the seminar. A class syllabus is given showing topics which were covered, and advantages and disadvantages of this type of educational experience are discussed.

1. General Course Goals

It has been argued in the literature [6] and claimed by students [10] that a University education in Computer Science is too theoretical, ineffective, non-applicable, and boring. It is not at all clear whether this argument is valid. However, the usual teaching techniques of lectures, exams, and homework may, in fact, be guilty at times of the above faults. Ideas have been tried which give the student a more individualized education [3,9]; and there have been attempts at giving the student an active rather than passive role in the educational process [2,5]. This paper describes a practicum-seminar course which was offered at the University of Colorado during the spring semester of 1974 which presents an alternative to exams and homework, and to a large extent replaces lectures by discussions. The purpose of the course was to teach principles of systems design, structured programming, debugging, documentation, and software engineering, while applying these principles to the design and implementation of a realistic software system. The course offers students an individualized educational experience in which students play an active role. This paper argues that the complaints listed above frequently are not present in a course of this type. However, this type of course requires a realistic project of large but not overwhelming size which is actually going to be used by a customer. Furthermore, this type of course demands a low student/teacher ratio and a large amount of planning and directing on the part of the instructor. Indeed, the commitment to a working final product by the teacher may be risky because success or failure may depend upon the caliber and nature of the students enrolled in the seminar. It is at least an order of magnitude more difficult to meet program completion deadlines with a team of students than with a team of paid professionals.

The software product which was produced by this class was a computerized registration system which attempts to assign classes and class times in such a way as to maximize student preferences. It prints out student schedules and class lists, and has a module which produces final grade reports. It was emphasized to our "customer" at the outset that the finished software product was a secondary goal subsidiary to the goal of a practical education in software engineering. The project was quite successful due to the perseverence of the class members and the flexibility and willingness to innovate on the part of the customer. The package was completed in time for its scheduled use (in summer school registration), and was highly and successfully utilized by the customer. The next paragraphs describe the project, and the environment in which it was developed.

2. The Project

The automated registration system was designed and implemented to be used by the Economics Institute, which is an intensive summer school program for foreign students who are just starting graduate study in the United States. After one initial summer of study and acclimatization in this program, the students disperse to various schools throughout the U.S. The Institute, which was begun by the American Economic Association in 1958 at the University of Colorado, has had a steadily growing enrollment of students. As the number of students has grown, so has the amount of work involved in assigning classes to students in appropriate subjects at appropriate levels. Thus it was decided that a computer science seminar would be instigated to implement a computer-assisted registration system. The size of the project was deemed feasible for a one semester project, with the expectation that some students would be able to gain experience and satisfaction by working as computer consultants to the Economics Institute during the summer after the seminar. Thus students could be involved in the project from conception through actual utilization of the system. It turned out that the system was well accepted to the extent that students were employed during the summer to add a final grade reporting module to the system.

One reason for the success of this project was the flexibility and innovation-oriented attitude which prevailed within the Institute. Thus it was possible to discuss and integrate the many ideas of the students with the goals and needs of the Institute in a fashion which would yield a workable sized system. One of the innovative ideas thus incorporated into the system was to accept an input from each enrolling student of his course preferences. Then the system would attempt to maximize student

preferences when it assigned students to classes. Another reason for the success of the project was the interest and enthusiasm shown by the participants (there were eight of us). Although students participating in the seminar ranged from freshmen to graduate students, all had gone through the registration process and could therefore relate to registration problems and offer suggestions and objections. Indeed, it was sometimes necessary to reject suggestions on the grounds that it would complicate our system to such an extent that our deadline could not possibly be attained. For example, it was decided that the administering of pre-enrollment placement tests, the grading of those tests, and the evaluation of a student's level of competence from tests and background information would all remain manual tasks. Our goals for the system included flexibility, good human-engineering, simplicity, reliability, and extensibility. Thus the actual seminar meetings, held once a week for two hours, covered the topics discussed in the next section.

3. Course Description

The seminar was intended to convey some of the latest theory and techniques to students, (e.g. structured programming), and to simultaneously provide a realistic opportunity to apply these techniques. Thus, the class was organized as a programming team, with the professor as project manager, two senior systems analysts, two systems programmers, two programmer-coders, and a documentation specialist. We further had a customer (the director of the Economics Institute attended many of our sessions) and occasionally brought in consultants. The first meetings were lectures presenting basic concepts which we would utilize: structured programming, top-down design, documentation and debugging aids; also presented at these lectures was a user's view of the desired registration system. At an early stage, learning responsibility and project design responsibility were transferred to members of the group: Each person was required to read about one or more potentially useful tools or techniques (e.g. decision tables, PERT charts, etc.) and give an oral report to the class. An extremely gross and simplified flowchart of the system was presented by the project manager and discussed. This was level 0 of our systems description. All team members were requested to produce a version of the level 1 and/or level 2 systems descriptions. This forced every student to think about how the system should be modularized and to make some design decisions about basic data structures and control flow. The systems analysts at an early stage set deadline dates and proposed the following schedule:

TENTATIVE SCHEDULE
STRUCTURES PROGRAMMING SEMINAR

WEEK	DATE	GOALS AND ACTIVITIES
1	Jan 23	Intro to Registration Process and to Course Objectives
2	Jan 30	Lecture and Discussion--Structured Programming, Programmer Teams Development Support Library, Module Proving, Existing Data Base, Job Assignments
3	Feb 6	Software Engineering--Design and Debugging Techniques
4	Feb 13	Select Documentation Technique
5	Feb 20	Level 1 Processes Documented
6	Feb 27	Level 2 Processes Documented; Level 1 Programs Proved with Stubs
7	Mar 6	Level 3 Documentation; Level 2 Proving
8	Mar 13	Level 3 Documentation; Level 2 Proving
9	Mar 20	Document Registration Process; Level 3 Proving
	Mar 27	Vacation
10	Apr 3	First Trial Run
11	Apr 10	CODE, DEBUG, and DISCUSS Programs
12	Apr 17	CODE, DEBUG, and DISCUSS Programs
13	Apr 24	Final Program Run (95 Simulated Students, 5 Real Students)
14	May 1	Overall Program and Procedures Firm and Documented for Use and Costed
	May 7	Start of Final Week
	Jun 21 22 23	Economics Institute Registration

This was the schedule as recorded by our documenter; however, alterations and slippages to this schedule occurred.

All programs were tested by our documenter before being accepted, and in some cases, there was time for all to read and discuss in class some

pieces of code and how they could be modified for clarity, modularity and extensibility. We were also concerned with the efficiency, but not at the expense of the above three criteria. There was one section of code which had to be rewritten after May 7 because of the above criteria. It was at times heartening to see the amount of class participation and suggestions on many issues such as programming languages to be used, manual key punching and verifying ideas, additions to the system for efficiency and extensibility, and ways to make the registration process easier on the students. It was at times disheartening, but realistic, to find that ideas carefully worked out (or even implemented in code) by students had to be revised or completely abandoned because of the nature of the system inputs. For example, some students may have blanks in certain placement scores, others may be required by their sponsoring agency to take a certain subject. Thus a large amount of time was spent interfacing with the Institute to find out exactly what was available, and exactly what outputs were required. Finally, a number of pragmatic tasks added immensely to the students' education. One example was the costing function. It was necessary to find out keypunch rates, CPU costs, and other overhead costs, and consider which of a number of alternatives was most cost effective. Students learned how to use sorting machines and interpreters as an integral part of the total registration process.

The topics and techniques covered in the course spanned a wide range; from analytic design ideas to pragmatic card sorting techniques; from arguing in class about programming style to giving polished presentations of the system to the Economics Institute staff; from the excitement of designing heuristic class schedule optimizers to the boredom of punching test data for 100 of last year's Institute students. Topics to which all

members of the class were exposed are listed in the outline of Figure 1. In addition, there were numerous tasks which were parcelled out to individuals or groups. It was necessary to carefully consider which tasks should be assigned to which individuals, so that all students would work on tasks which were significant learning experiences and which were not beyond the student's level of expertise. There were large variations in the amount of guidance needed by various participants in the seminar. In total, the question should be asked as to whether the mundane tasks outweigh the benefit of the educational experiences. We address this question in the final section of the paper. First, a brief explanation will be presented of the programs and data which make up the registration system. More detailed documentation can be obtained by contacting the author.

Figure 1

Structured Programming Seminar Contents

1. Techniques of Structured Programming
2. Modularization and Clarity of Code
3. Design Strategies
4. Documentation and Standardization
5. Testing and Debugging
6. Certification and Validation
7. Measurement and Evaluation
8. Customer Interfacing
9. Maintenance and Modification

4. Program Modules of the Registration System

The basic objective of the project was the design and implementation of a system to register students. We decided to do this in a manner which would allow students to specify their preferences as an input to the computer. The functions which are performed by computer program modules are nicely described in the level 0 documentation which reads:

1. Module A -- Produce Student Placements and Questionnaires
2. Module B -- Produce Preliminary Class Schedules
3. Module C -- Produce Final Student Schedules and Final Class Lists

The Level 0 documentation does not indicate algorithms or file structures employed to perform these three tasks. Also there is no indication of manual processing which forms an important part of the total system. This manual processing is done before and after the various program modules are run on the computer. Level 1 documentation does describe the manual steps needed, but in a non-detailed form. It is presented here to help the reader's understanding, and to show what the structured programming technique of top down design implies as it was employed in the seminar.

Step 0.1 -- Punch student data and test placement scores.

This step actually involves gathering and quantifying results from a battery of pre-registration tests which are administered to all of the foreign students upon their arrival at the University of Colorado. Tests are administered in the areas of English, Micro Economics, Macro Economics, Mathematics, and Statistics, which form the basis of input data to Module A.

Step 1.0 -- Run Module A

Besides accepting test scores, Module A also accepts general instructions and comments to students as input which are output along with a student's placement level in each subject area. A second sheet of output consists of one preference questionnaire per student. Examples of typical output sheets received by a student are shown in Figures 2a and 2b. This program module, which consists of 72 SNOBOL statements, ran at a cost of $9.45 for 220 students on a CDC 6400 computer.

Step 1.1 -- Punch preference data

The students fill out and return their preference questionnaires. This information is punched onto cards to form one of the input items to Module B.

Step 1.2 -- Punch course constraint cards

For each course it is necessary to specify various constraints. These include maximum number of sections offered, minimum and maximum number of students in a section, and the placement levels required for students to enroll in the course. These constraints form the second input item to Module B.

Step 2.0 -- Run Module B

This program module uses the input data described in the previous two steps to place students into sections of courses and to assign times when sections shall meet. It attempts to do this in such a way as to maximize student preferences. The optimization technique used in this program is an iterative approximation method which can accept assistance at certain states of the iteration. The output of this program consists of a class

Figure 2a

ID: 0026

ECONOMICS INSTITUTE 1974
COURSE PLACEMENT FORM
MAIN SESSION, SECOND TERM

SECTION I GENERAL INSTRUCTIONS:

THE INFORMATION CONTAINED ON THIS SHEET HAS
EEN PREVIOUSLY DESCRIBED TO YOU. WE WOULD LIKE YOU NOW TO COMPLETE THE ATTACH-
D PREFERENCE FORM. PLEASE RETURN IT TO ONE OF THE SCHEDULE ADVISORS LOCATED IN
CON 119 BETWEEN 8:00-10:00AM, FRIDAY, JULY 19, 1974. IF YOU HAVE ANY QUESTIONS
ONCERNING THE LISTING OF PREFERENCES, THE SCHEDULE ADVISORS WILL BE AVAILABLE
O HELP YOU. PROVIDED ALL PREFERENCE FORMS ARE COMPLETED BY 10:00AM FRIDAY, A
LASS SCHEDULE WILL BE AVAILABLE TO BE PICKED UP AFTER INTERNATIONAL
IGHT FESTIVITIES (9:00) SUNDAY IN YOUR MAILBOX IN ECON 5.

ECTION II PLACEMENTS:

BASED ON TEST SCORES AND REGISTRATION DATA, WE HAVE
LACED YOU BY LEVEL IN THE FOLLOWING SUBJECTS:

SUBJECTS	PLACEMENT
********	*********
ENGLISH	3A
MICROECONOMICS	2A
MACROECONOMICS	1B
MATHEMATICS	2A
STATISTICS	1A

ECTION III ORGANIZATIONAL MEETINGS FOR FRIDAY, JUNE 21, 1974

A. ENGLISH MEETINGS:

1. FOR ENGLISH 1A AND 1B STUDENTS:
   A. MEET IN HALE 104, 10:35AM, FRIDAY, FOR AN INTRODUCTION TO THE READING LAB PROGRAM (50 MINUTES).
   B. MEET IN OLD MAIN 8 AT 11:30AM, FRIDAY, FOR AN INTRODUCTION TO THE LANGUAGE LAB (40 MINUTES).
2. FOR ENGLISH 2A STUDENTS:
   A. MEET IN HALE 104, 10:35AM, FRIDAY, FOR AN INTRODUCTION TO THE READING LAB PROGRAM (50 MINUTES).
   B. MEET IN OLD MAIN 8 AT 12:15PM, FRIDAY, FOR AN INTRODUCTION TO THE LANGUAGE LAB (40 MINUTES).
3. FOR ENGLISH 2B STUDENTS:
   A. MEET IN OLD MAIN 8, 10:35AM, FRIDAY, FOR AN INTRODUCTION TO THE LANGUAGE LAB (40 MINUTES).
   B. MEET IN HALE 104, 11:45AM, FRIDAY, FOR AN INTRODUCTION TO THE READING LAB PROGRAM (50 MINUTES).
4. FOR ENGLISH 3A STUDENTS:
   A. MEET IN HELLEMS 81, 10:35AM, FRIDAY, FOR AN INTRODUCTION TO THE #3# LEVEL ENGLISH PROGRAM (60 MINUTES).
   B. MEET IN HALE 104, 11:45AM, FRIDAY, FOR AN INTRODUCTION TO THE READING LAB PROGRAM (50 MINUTES).
5. FOR ENGLISH 3B STUDENTS:
   A. MEET IN HELLEMS 141, 10:35AM, FRIDAY, FOR AN INTRODUCTION TO THE #3# LEVEL ENGLISH PROGRAM (60 MINUTES).

B. ECONOMICS LECTURE: THERE WILL BE AN INTRODUCTORY LECTURE ON THE TOPIC, ETHODOLOGY IN ECONOMICS, BY PROESSSOR RICHARD LEIGHTON AT 2:00 PM, JULY 19, IN UGG 201. ALL STUDENTS ARE INVITED.

Figure 2b

ID: 0024

PREFERENCE REGISTRATION FORM
MAIN SESSION, SECOND HALF
ECONOMICS INSTITUTE 1974.

PLEASE INDICATE YOUR COURSE PREFERENCE IN EACH SUBJECT AREA BY CIRCLING A NUMBER FROM 1 TO 5 REFLECTING THE STRENGTH OF YOUR PREFERENCE FOR EACH SUBJECT AREA.

SUBJECT AREA	STRENGTH OF PREFERENCE				
	STRONGEST				WEAKEST
MICROECONOMICS	1	2	3	4	5
MACROECONOMICS	1	2	3	4	5
MATHEMATICS	1	2	3	4	5
STATISTICS	1	2	3	4	5
BUSINESS	1	2	3	4	5

YOU MAY SPECIFY ONE SUBJECT AREA, OTHER THAN ENGLISH, TO WHICH YOU DO NOT WISH TO BE ASSIGNED IN THE NEXT TERM.

DO NOT ASSIGN ME TO THE FOLLOWING SUBJECT AREA IN THE MAIN SESSION, SECOND TERM

PLEASE COMPLETE AND RETURN THIS FORM TO ONE OF THE ADVISORS IN ROOM 119 OF THE ECONOMICS BUILDING BETWEEN 08-00 AND 10-00 AM, FRIDAY, JULY 19.

list for each section of each course, along with a card for each student indicating which courses he was placed in. This program, when run on a 220 student data set, cost $7.47. The module consists of 808 Fortran statements.

Step 2.1 -- Punch final class information

Special and optional courses such as English labs and the Economics Lecture Series must be inserted into the schedules of qualified students. Also, a number of scheduling tasks which, after some debate, were not automated, must be carried out such as the assignment of class rooms and instructors to courses. Finally, the results of step 2.0 and of these manual tasks must be approved (or altered) by the director of the Economics Institute.

Step 3.0 -- Run Module C

This last module accepts the cards punched by Module B plus the data generated by step 2.1, and from these prints a schedule of courses for each student along with comments and footnotes. This program module also prints out class lists to be distributed to the class instructors. See Figures 3a and 3b for an example of the output generated by this program. Module C was coded in SNOBOL, contains 210 statements and costs $78.00 when run on a 220 student data set.

The appendix of this paper contains a documents which was abstracted from level 2 documentation of the system. It introduces some of the data structures and card format types. The largest and most sophisticated module of the system is Module B (808 statements). This module is the only one using heuristic techniques, and is designed to allow Economic Institute staff members to interact with the program. The basic scheduling

Figure 3a

E: ABDOELKADIR K. K.

ECONOMICS INSTITUTE 1974

CLASS SCHEDULE
MAIN SESSION, SECOND HALF
(SECOND TWO WEEKS)(1)

TIME	CLASS	SECT	BUILDING	DAYS	INSTRUCTOR/S
0900-1000AM	ECON LECT SERIES	(3)	GEOLOGY 121	MTWTHF	LECTURERS
1035-1150AM	MACROECONOMICS	1AB	HALE 104	MTWTHF	FALERO-LANEY
0220-0320PM	MACROECNOMICS	1AB	GUGG 107	M,TH	LANEY-FALERO
0220-0320PM	MICROECONOMICS	1AB	GUGG 28	T,F	FREIDLANDER-WALTON
0330-0445PM	MICROECONOMICS	1AB	GUGG 107	MTWTHF	WALTON-FREIDLANDER

FOOTNOTES FOR SCHEDULE ABOVE

(1) CLASS CHANGES EFFECTIVE MONDAY, AUGUST 5, 1974, CLASSES MEETING M,W,F CHANGED TO T,TH--CLASSES MEETING T,TH CHANGED TO M,W,F. THESE CHANGES ARE REFLECTED IN THE ABOVE ACHEDULE

(2) A COMBINATION OF STRUCTURE COURSE-RELATED LAB PROGRAMS, SUBJECT COMPREHENSION LAB PROGRAMS AND READING LAB PROGRAMS AS APPROPRIATE.

(3) FOR ALL STUDENTS OTHER THAN THOSE IN ENGLISH 2A1 WHO HAVE A CHOICE BETWEEN ATTENDING THE LECTURE SERIES AND A SPECIAL ENGLISH LAB PROGRAM. A LIST OF LECTURERS AND LECTURE TOPICS WILL BE DISTRIBUTED SEPARATELY ON A WEEKLY BASIS.

(4) THIS TIME AND THE MEETING SPACE IS AVAILABLE FOR ANY ADDITIONAL OR SUBSTITUTE LECTURES, TUTORIALS, OFFICE HOURS, ETC.,THE PROFESSOR TO ANNOUNCE.

NOTE-IF THERE ARE ANY ERRORS IN THE SCHEDULE ABOVE SEE GEORGE ANTOINE IN ECON 1

SEE OPTIONAL COURSES AND FOOTNOTES BELOW

OPTIONAL COURSES AND FOOTNOTES(MAY BE CANCELLED IF DEMAND IS INSUFFICIENT)

IME	CLASS	SECT	BUILDING	DAYS	INSTRUCTOR/S	
750-0850AM	SUBJ COMP LAB(OPT)	2B3	ECON 117	T,TH(1)	PETERSON	7
750-0850AM	SUBJ COMP LAB(OPT)	2B4	ECON 117	T,TH(1)	PETERSON	7
750-0850AM	SUBJ COMP LAB(OPT)	2BV3	ECON 117	M,W,F(1)	PETERSON	7
750-0850AM	MATH VIDEO (2)	2A	ECON 119	MTWTHF	PECK	7
750-0850AM	MATH VIDEO (2)	2B	HELLEMS 199	MTWTHF	RESEK-ULVELING	7
750-0850AM	STAT VIDEO (2)	1A	HUNTER 208	MTWTHF	DOWLING	7
200-0100PM	SUBJ COMP LAB(OPT)	2B1	ECON 117	T,TH(1)	JOHNSON-ROWDEN	7
200-0100PM	SUBJ COMP LAB(OPT)	2B2	ECON 117	T,TH(1)	JOHNSON-ROWDEN	7
200-0100PM	SUBJ COMP LAB(OPT)	2BV1	ECON 117	M,W,F(1)	JOHNSON-ROWDEN	7
200-0100PM	SUBJ COMP LAB(OPT)	2BV2	ECON 117	M,W,F(1)	JOHNSON-ROWDEN	7
200-0100PM	MICRO VIDEO (2)	2A	HUNTER 208	MTWTHF	WESTFIELD	7
200-0100PM	MICRO VIDEO (2)	2B	HLMS ANX136	MTWTHF	WESTFIELD	7
200-0100PM	MACRO VIDEO (2)	2B	ECON 119	MTWTHF	NICHOLLS	7
200-0100PM	INTRO AMER BUS(1)(4)		ENG CTR 0-12	T,TH	WILSTEAD-BEATTY	7
200-0100PM	SPEC SEMINRS(OPT)	(3)	ECON 205	(3)	LECTURER	7
220-0320PM	MICRO VIDEO (2)	2A	HUNTER208	M,T,TH,F	WESTFIELD	7
220-0320PM	MICRO VIDEO (2)	2B	ECON119	T,F	WESTFIELD	7
220-0320PM	MATH VIDEO (2)	2B	HELLEMS 181	M,T,TH,F	RESEK-ULVELING	7
220-0320PM	STAT VIDEO (2)	1B	HELLEMS 199	M,T,TH,F	DOWLING	7
220-0320PM	AS ANNOUNCED	(5)	AS ANNOUNCED	W	AS ANNOUNCED	7
700-0800PM	MICRO VIDEO (2)	2A	HLMS ANX 134	MTWTH	WESTFIELD	7
700-0800PM	MICRO VIDEO (2)	2B	HLMS ANX 136	MTWTH	WESTFIELD	7
700-0800PM	MATH VIDEO (2)	1AB	HLMS ANX 138	MTWTH	KEYSER	7
700-0800PM	STAT VIDEO (2)	1A	HLMS ANX 140	MTWTH	YANCEY	7
800-0900PM	MATH VIDEO (2)	2A	HLMS ANX 134	MTWTH	PECK	7
800-0900PM	MATH VIDEO (2)	2B	HLMS ANX 136	MTWTH	RESEK-ULVELING	7
800-0900PM	STAT VIDEO (2)	1B	HLMS ANX 138	MTWTH	DOWLING	7
800-0900PM	MACRO VIDEO (2)	2B	HLMS ANX 140	MTWTH	NICHOLLS	7
300-0500PM	ENGLISH LAB (OPT)(6)		OLD MAIN 8	SU	PROCTOR	7
700-0930PM	ENGLISH LAB (OPT)(6)		OLD MAIN 8	MTWTH	PROCTOR	7
700-0930PM	READING LAB(OPT)(6)		ECON 205	MTWTH	FICKETT-ROWDEN	7

FOOTNOTES FOR OPTIONAL COURSES

1) CLASS CHANGES EFFECTIVE MONDAY, AUGUST 5, 1974, CLASSES MEETING M,W,F HANGED TO T,TH--CLASSES MEETING T,TH CHANGED TO M,W,F. THESE CHANGES ARE EFLECTED IN THE ABOVE ACHEDULE

2) FOR STUDENTS NOT ASSIGNED TO OTHER CLASSES DURING THIS PERIOD. DAILY VIDEO ROGRAMS WILL BE POSTED IN RECEPTION ROOM. IN ADDITION , MATH 1AB AND STAT 1A IDEOTAPES ARE AVAILABLE IN NORLIN 135 DURING LIBRARY HOURS.

3) TO BE ANNOUNCED ON A WEEKLY BASIS. ALL STUDENTS NOT ASSIGNED TO OTHER CLASS- S ARE INVITED TO ATTEND THESE SEMINARS CONDUCTED BY THE 0900AM LECTURER.

4) INTERESTED STUDENTS WHO HAVE NO CLASS CONFLICT ARE INVITED TO ATTEND.

5) OCCASIONAL SPECIAL LECTURES OR ASSEMBLIES MAY BE HELD DURING THIS PERIOD.

6) A MINIMUM OF TWO HOURS PER WEEK OF ADDITIONAL LAB WORK IS RECOMMENDED FOR TUDENTS IN ENGLISH 2A. READING LAB IS AVAILABLE IN OLD MAIN 8 AND ECON 205. HESE LABS ARE A COMBINATION OF STRUCTURE COURSE-RELATED LAB PROGRAMS, SUBJECT OMPREHENSION LAB PROGRAMS AND READING LAB PROGRAMS AS APPROPRIATE.

technique employed by this module was formulated and discussed at some length in the seminar. It reflects to some extent the philosophy and biases of the Economics Institute. The program first attempts to place every student in one of his first preference choices with bias against those students who either have more than one first preference, or have a placement level in the subject which is somewhat borderline. The justification for this strategy is that if a student with several first preferences gets closed out of a class, he can still try for another first preference. Also, if there are more students than can fit in some class, it makes a more homogeneous (and therefore better?) class if borderline students are closed out. Next, the program places students into a second course, etc., until students' schedules are complete (Pass I). If several courses are given equal preference by the student, the order of selection of courses is based upon Institute preferences which can be specified as input data. At this stage, it is necessary to check for under-enrolled courses, cancel those courses, and go through the whole program again (Pass II), trying to place those students who were members of the cancelled courses. Print-outs are produced after both Passes I and II. After a run, Institute staff members use the print-outs to juggle course sizes and sections to optimize. Further, the program has a very flexible facility called the course map matrix which defines a function mapping placement levels into courses. Single entry changes to this matrix, which is input data, can cause all borderline high (or borderline low) students to be placed in a higher (or lower) level course. Similarly changing this matrix can split a course in two or combine two levels of courses. The Institute staff usually runs the program first with very high maximum class sizes to see the student

demand. Then by using the features described above, they can zero in on an optimal schedule in two to six more computer runs. This interactive capability has proved quite useful and flexible. Finally, there may be students whose placement levels are such that they cannot fit into any of the offered courses. In this case, the Institute staff places him manually, or assigns him to independent study. By manually punching a card for this student and inserting it in the output card deck of Module B, this student will be given a satisfactory schedule by Module C.

These programs were used during registration for two terms during the summer of 1974. In the first term, 79 per cent of the students received their first choice, and 41 students had to be manually placed by altering their class cards punched by Module B. In the second term, 92 per cent of the students received their first choice and only 6 students had to be placed manually. This excellent performance was attributed to a single alteration to Module B: If a student's first (or k-th) preference is a class which meets at several times, and the times cause no conflicts for this student, then his second (k+1-th) preference was satisfied first, allowing other students who cannot take the course at several times to get placed in whichever section they can before the original, more flexible student. Now, this crucial module seems to be well-tuned and giving satisfactory performance.

5. Final Outcome and Evaluation

The project described had a successful outcome. The Economics Institute has decided to adopt this automated registration system as its standard procedure, and in fact, students from the project were hired during the summer to produce a fourth module which outputs student grade reports. Good points which were apparent during the seminar include its realistic nature, the exposure of all students to rigorous documentation and debugging standards, and the informal nature of the class sessions which encouraged participation by all. There are a number of considerations within the seminar which I consider to be bad points. First, more time must be spent by the instructor before and during a successful seminar of this type than in an ordinary lecture. Furthermore, this type of format can only work with a very small number of students. There were irritations caused by customer changes in their desired product specifications, and when the registration process actually took place, the Institute continually wanted more and more frills and additions (and always immediately). This type of patchwork tended to mess up an otherwise beautifully structured system. Finally, it was obvious that the quality of students and therefore the quality of code varied enormously. It was previously mentioned that one subroutine had to be completely rewritten and the rather high cost of Module C could be reduced by some rewriting of its code.

In summary, members of the seminar seem to have benefited from having concepts enforced by their practical application to a realistic project. I feel that the documenting, debugging, and other similar tasks were a realistic and necessary part of the students' education, and were not

wasted efforts. Unfortunately, this type of project and environment is not always available. An exercise without a real application may not catch as much enthusiasm and may not provide a very realistic experience. On the other hand, a customer who is not flexible may be quite dissatisfied with his final product.

I would like, in this closing paragraph, to acknowledge the hard work of my seminar participants and to express my thanks to members of the Economics Institute for their help, and their extremely open-minded and congenial treatment. It was enjoyable being a member of the E.I. family. Special acknowledgments for contributions above and beyond class requirements to: George Antoine, John Bidwell, Greg Slansky, and Professor Wynn Owen.

APPENDIX

Attempts at computerization of the task of scheduling students for classes have had some sparkling successes and some tragic failures (see news clipping below). The purpose of these computerized systems has always been to increase efficiency and speed while decreasing the amount of manpower needed to carry out the task. This report describes a computerized registration system with a different goal. It attempts to assign classes and class times in such a way as to maximize student preferences. This system was designed and implemented for the Economics Institute at the University of Colorado as a seminar on structured programming during the spring and summer of 1974.

'Bad' Package Cited in Delay Of School Grades

By Toni Wiseman
Of the CW Staff

TOLEDO, Ohio — School districts in this area have experienced delays of up to five weeks in getting out grades, as well as spending thousands of dollars in overtime, because of problems with an IBM software program.

Roger Schantz, director of computer services at Owens Technical College, is one of the users of the Epic program who has experienced difficulties. He said there are not many Epic users yet, only about 27, but most have had problems, he claimed, citing the Youngstown (Ohio) Board of Education as a prime example.

Youngstown, which handles 15 schools and has not been able to get the package working correctly, now has a backlog from the first marking period and is into the second.

The Villain

Epic consists of four parts: Fast, for test scoring; Socrates, for student scheduling and data file maintenance; Student, for grade reporting; and a Budget-Finance package.

Schantz had problems with Student and Socrates, the only two programs he has.

"The two packages just didn't interface," he said. "You were supposedly able to go from the scheduling into the student grade reporting, but they didn't interface as their documentation said.

The student enrollment and registration procedure described in this document is most appropriately termed a computer assisted registration procedure because it consists of a number of manual steps as well as some computerized steps. The idea is to intermingle the flexibility and sensitivity of human judgment with the speed and precision of automated data processing. Conceptually, the steps necessary in the procedure are:

1. For each student, punch a header card containing his name and identification number (ID). This type of card always has a 1 punched in column 80.
2. For each student, punch a subject card for each subject in which the student might possible enroll containing the student ID along with the name and number of the subject. This type of card is identified by a 2 punched in column 80. All of a student's subject cards immediately follow his header card in a deck of punched cards called the student data deck.
3. The subject cards may be optionally follwed by one or more comment cards which are punched for those students to which an individualized message must be transmitted. Comment cards must have a 3 punched in column 80. As the registration proceeds, the student data deck is modified and augmented as described in the following steps.
4. From placement tests administered to the students, data is obtained concerning their level of competence in various subject areas. This data must be punched on the students' subject cards.
5. A computer program next produces student preference sheets which are questionnaires with instructions and advice to students on them. After students indicate their preference on these sheets, the information obtained must be punched onto students' subject cards.

6. Using all of the data obtained in the preceding steps plus a deck of tentative course offerings, a second computer program tries to optimally schedule courses into time slots and place students into courses. The optimization technique used within this program is an iterative approximation method which can accept human assistance at certain stages of the iteration. The output of this program consists of a list of students in a class (for each class), along with a card for each student indicating which classes he was placed in. This card contains a 4 in column 80 and must be merged into the student data deck.
7. The next step consists of any manual changes and adjustments found necessary. Also, classrooms, instructors, seminar times, etc., can be specified on a deck of class assignment cards.
8. The student data deck plus the class assignment cards are input to a third computer program which prints out a final schedule of courses for each student.

It should be noted that some of the more mundane tasks (such as step 2) can be and have been automated. It should also be noted that some of the most difficult and time consuming tasks must be done manually. Notably, the administering of placement tests, the grading of those tests, and the evaluation of a student's level of competence are all done manually. These tasks and others present interesting and challenging topics for study and possible automation. It is hoped that this experimental system will continue to evolve into an efficient system while remaining human engineered and sensitive to the needs of students and faculty.

BIBLIOGRAPHY

1. Amarel, Saul, "Computer Science: A Conceptual Framework for Curriculum Planning," Comm. ACM 14, 6, June 1971.

2. Gary, John, "A Proposal for an Undergraduate Computer Laboratory," CS Report, Jan. 1974, University of Colorado Department of Computer Science, Boulder, CO.

3. Hoberack, et.al., "Theory of PSI Evaluated for Engineering Education," IEEE Trans. Ed., Vol. E-15, 1972.

4. Hosch, F. A., "Some Comments on the Role of Computer Science Education," SIGCSE Bulletin, Vol. 5, No. 3, Sept. 1973.

5. Kalmey, Donald L., "Profile of a Computer & Information Science B. S. Graduate," SIGCSE Bulletin, Vol. 6, No. 1, Feb. 1974.

6. Keller, F., "Good-bye Teacher ... ", Jour. Appl. Behavior Anal., No. 1, 1968.

7. Pitteway, M., "Goals of an Undergraduate Program in Computer Science," Computer Science Departmental Report, Brunel University, England.

8. Sackman, H., "Man-Computer Problem Solving," Auerback Pub. Co., N.Y., 1970.

9. Salton, Gerard, "What is Computer Science?", J. ACM 19, 1 (Jan. 1972).

10. Sistaire, John, & Sondak, N., "Introduction to Digital Computer Programming," SIGCSE Bulletin, Vol. 6, No. 1, Feb. 1974.

11. Wegner, Peter, "A View of Computer Science Education," Mathematical Education, Feb. 1972.