TUTCHE - A PROGRAM PACKAGE FOR TUTORING CHEMICAL ENGINEERS*

F. P. STAINTHORP, D. LOMAS, AND
J. ALONSO
U.M.I.S.T.
Manchester M6O 1QD, England

ALL ENGINEERS, AND particularly chemical engineers, have needed to bring to their professional activities a blend of inspiration, information and perspiration. A common complaint about the academic training of engineers is cogently expressed in the Finneston Report where it is concluded that not enough attention has been paid, in the recent past, to developing the inspirational aspects of design. This report recognizes the need for more time and effort to be devoted to training in design methodologies but, apart from one brief reference and recommendation, it fails to anticipate the impact that developments in computing techniques would have in the areas of training and design methodology.

As computer technology has evolved in the directions of large efficient time-shared machines on the one hand and more powerful desk-top mini-computers or micro-processors on the other we are witnessing significant changes in the ways in which we tackle the design both of individual units and of integrated systems. It is particularly in the opportunities that we now have for examining alternatives in large scale "integrated" designs that we can see the greatest impact.

All the major chemical manufacturers, oil refiners, plant contractors and design offices have built up their computer-aided engineering activities over the past fifteen years so that they now possess significant libraries of well proven programs for use by their staff. The smaller firms may not have been able to spare the effort to develop the soft-ware themselves but in many instances make use of bureaus which have established themselves as reliable sources of design capacity on large, shared, computers.

The smaller firms may not have been able to spare the effort to develop the soft-ware themselves but in many instances make use of bureaus which have established themselves as reliable sources of design capacity on large, shared, computers.

Clearly, the universities have a responsibility to ensure that when their graduates emerge they are prepared to work in this new manner and many academic institutes have not been slow to recognise the potential for using computers to improve their teaching and training activities. The original CACHE committee which published a well-known set of some 112 programs in 1972 is now a Corporation which is fostering these activities in the U.S.A. Of these the most ambitious is the CHEMI project which aims to produce some 500 tried and tested instructional modules. CACHE also supports the use of FLOWTRAN as a networked flow-sheeting package and has now initiated a project to develop and exchange Micro-computer Aids for Chemical Engineering.

CHICHEE and EURECHA have encouraged parallel but smaller scale activities in the United Kingdom and Europe.

Most of the activities referred to above were directed to exploiting batch computing facilities but it became apparent in the late '70's that developments in low-cost interactive computing were going to have the most significant impact on Computer-Aided Design (in all its shades of meaning) and that, potentially, this mode of computation could be of equal importance for the development of Computer-Aided learning. It was these considerations which led the authors to examine the problems of constructing a package which would be designed to take advantage of the vibrant characteristics of this mode of working.

After some initial trials using an existing Cyber 72, the hard-ware described in the next section was acquired to evaluate the package under conditions which would provide some 250 chemical

*A revised version of a paper originally presented to the 1979 Montreux Conference—"Computer Applications in Chemical Engineering."

engineering students (3 years of B.Sc. and 1 of M.Sc.) with a novel experience and also permit colleagues in other departments to take advantage of the facility in order to develop teaching packages appropriate to their own disciplines.

HARDWARE

The hard-ware to be described is as it will be at the completion of the first phase of the project and is the best that we could afford and most appropriate for the computers available to us. It should be stressed that the TUTCHE package does not depend on this hardware and could be run on any interactive system ranging from a single-user mini with a 10 char/sec teletype to a multi-user main frame provided with a large number of (colour) graphics terminals. We had outgrown the former but could not afford the latter.

The compromise was to set up three clusters of five terminals arranged as shown in Fig. 1. In any one cluster all five terminals have an upper and lower case alphanumeric capability, three have a graphics capability which is required to give a full Tektronix 4010 emulation and one of these has a micro-processor capacity (twin 5½″ floppies) so that it can work in "stand alone" mode or in transparent mode when it behaves as a α/graphics terminal. All five may be connected by a simple multi-pole switch to an α/graphics printer with extended buffer storage that enables any one user to dump a "page" as displayed at the terminal onto the printer. (We should remark,

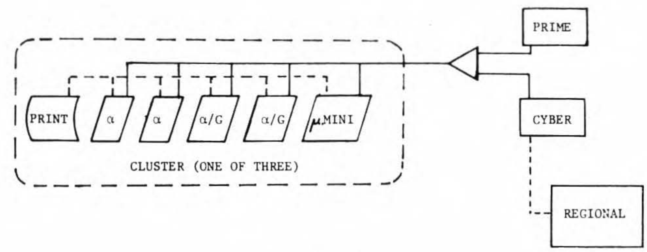
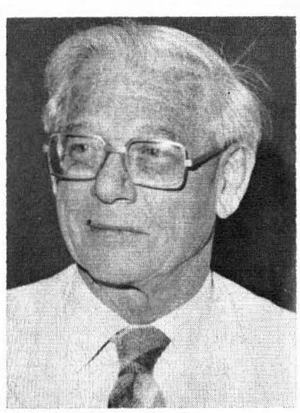


FIGURE 1

here, that we have had no problems with α-numeric displays and printing, but we are still not satisfied with the copying of graphics display and are currently examining alternatives.) The ambition is to progressively upgrade the alphanumeric terminals as funds and graphics requirements increase.

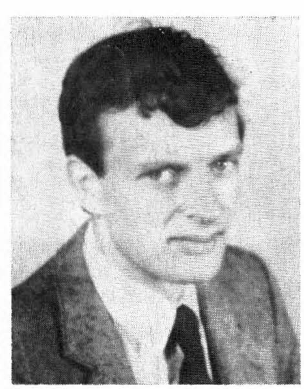
Any of the fifteen terminals in the suite will be connected via a "Gandalf Switch" to various machines and currently we may access either a Prime 750 or a Cyber 172 both of which provide time-shared interactive operation at linespeeds of 2.4kB which is just about the lowest acceptable rate for graphics work and is a very comfortable rate for conventional dialogue.

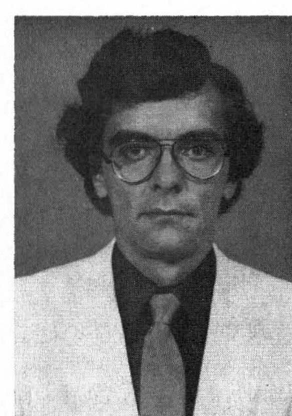
The Prime has proved to be a very effective machine since, up to the limits of the locally set connection limit, it gives a very fast response to "trivial" dialogue—i.e. during data input sessions. An important feature that we have been able to exploit is a "shared code" facility which means that when all fifteen terminals are used by students on the same tutorial exercise we can arrange



F. P. Stainthorp graduated in chemical engineering from Manchester in 1941. After eight years in industry he was transferred to University College, London to work for his doctorate and then returned to chemical industry for a further six years. He was recruited to Manchester, UMIST, at a time of expansion with a special remit to cultivate the growing fields of process control and digital computation. He is chairman of CICHEE and directs the chemical engineering computing service at UMIST. (L)

David Lomas is a member of the British Computer Society and has spent most of his working life in University engineering departments first as a programmer but lately as the supervisor of the chemical engineering computing service at UMIST. He has made a special study of the problems of transferring nominally 'standard' programs between different computing systems. (C)





J. Alonso received his Chemical Engineering and MSc. degrees from the Universidad Central Venezuela in 1972 and IVIC 1976, and his Ph.D. degree from the University of Manchester England in 1979. He currently is lecturer of process simulation at the Universidad Central Venezuela and his interests have generally been in software developments. (R) for only one copy of the program to be resident in the machine. This has the big advantage of enabling the normal load of other users to be satisfied without them seeing a significant impact from the student users.

The Cyber 170, having nearly 200 terminals connected across two campuses, tends to get overloaded at peak times, but a scheduling facility in the switch should ease this problem in the future. The big advantage of this machine is that it is linked to a CDC 7600 which is used to process the largest design packages. In this mode of working the intention is to use the interactive machine for validation and preprocessing activities that generate a suitable data file for subsequent batch processing in the larger machines. The interactive system is also used for post-processing the output files resulting from the batch machine so that only selected material required for permanent record need be printed.

SOFTWARE

Fig. 2 is a diagrammatic representation of the overall structure of the package which is shown in this manner for ease of discussion of the various components. There is some correspondence with the actual system since many of the programs need to use common data files and in general only one program will be in use by a student at any

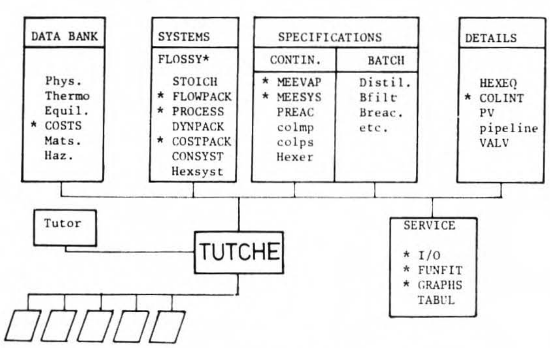


FIGURE 2

one time. We think of four banks which contain, respectively, common data, overall systems related programs, unit design specification programs, and detail design programs. Another bank contains a library of service programs or sub-routines which can be used to deal with input/output problems, graphical interactions, function fitting etc. Each component of the total package communicates with the user via a dialogue that is appropriate to the application. On the first occasion that a program is used the requests are full but not verbose. On subsequent occasions abbreviated prompts are used but the student may always ask for "Help" whereupon the abbreviations are explained and amplified if further requests for help are made. We also find it convenient to include a switch that enables the frequent user of a particular program to suppress the dialogue.

TUTCHE/TUTOR

Students gain access to the suite via a program named TUTCHE. This program is a purpose written data base management system which serves to control and record the use of the various programs by the students. It also tidies up the student's own data and results in files on exit so that they may be retrieved reliably on re-entry.

TUTOR is a separate interactive program which enables a tutor to examine and modify the record held in TUTCHE. The use made by all students of any one program may be displayed or the record of programs used by any one student may be examined. In some cases a permit has to be set and continued use of a program may be contingent upon monitored successes at selected points. The tutor may insert, examine and modify these controls through TUTOR.

This control is not a case of "Big Brother watching" since, in the main, the programs are directed to use by responsible and mature students who each work at their own pace but within constraints imposed by the capacity of the resources. Examples of the regulation of use will be given later but in general they either ensure that the student has understood the fundamentals of the tutorial exercise and is not using the program as a "black-box" or they serve to curb extravagance on open-ended exercises. This discipline is intended to correspond to that which they will find necessary in industrial environments when the use and time has to be paid for.

The titles shown in Fig. 2 are those that form the current basis for the package. Those marked with an asterisk have reached the stage of regular student use, those in capitals but not starred are undergoing trials and those in lower case are still

On subsequent occasions abbreviated prompts are used but the student may always ask for "Help," whereupon the abbreviations are explained and amplified if further requests for help are made.

in preparation. More details of some of the programs and how they are used follow.

FLOWPACK/PROCESS

FLOWPACK and PROCESS are two major flow-sheeting packages that have been made available to U.K. students by the generosity of ICI Ltd and Simulation Sciences Inc. respectively. Both packages are maintained on the CDC 7600's at the University of Manchester Regional Computer Centre by UMIST department of Chemical Engineering for use by some fifteen departments at other, remote, universities. A growing number of students make extensive use of the packages for their design exercises and we find the nominal royalty figure calculated and displayed for each run of the PROCESS program a very convenient means of control. Each student may be given an allocation of "cost- units" within which the exercise should be completed and this gives a strong incentive to rational use of the package.

At the moment the flowsheeting packages are run as batch jobs from card input but we are developing an interactive preprocessor that will validate the data as they are introduced. Ultimately, users will start from a flow-diagram of the type shown in Fig. 3 which has been produced by using "FLOSSY" on a graphics terminal .

COSTPACK

This is a three part package which has provision to generate costing information in any currancy using a reference year and a selected cost index. After the initial dialogue the student may enter any of three sections :—

"COSTIT" for costing items (vessels, heat-exchangers, etc) using stored correlations relating F.O.B. costs to a capacity factor with adjustment for selected materials of construction. In a few cases the cost of the item can be estimated for detailed fabrication factors.
"COSTFCI" for estimating Fixed Capital Investment. Two methods are provided: (a)Zevnick-Buchanon giving a very approximate figure based on the principle units, and (b) a factorial method (Millar) applying factors for installation, civil, structural work, etc to all the items.
"COSTASS" for estimating product costs from FCI, depreciation policy, taxes, materials, utilities, labour rates, etc, etc.
Profitability analysis and comparisons based on eight different criteria can be made and this may be followed by a sensitivity analysis in which the effects of varying any of the principle costing components may be examined.

All of the above are closely tied to tutorial exercises aimed at reinforcing an appreciation of the economic factors that determine good design.

MEEVAP/MEESYS

The determination of economic designs is intended to be brought out in all the programs stored in the Specifications Bank and the approach adopted is illustrated by these two programs which deal, respectively, with the fundamentals

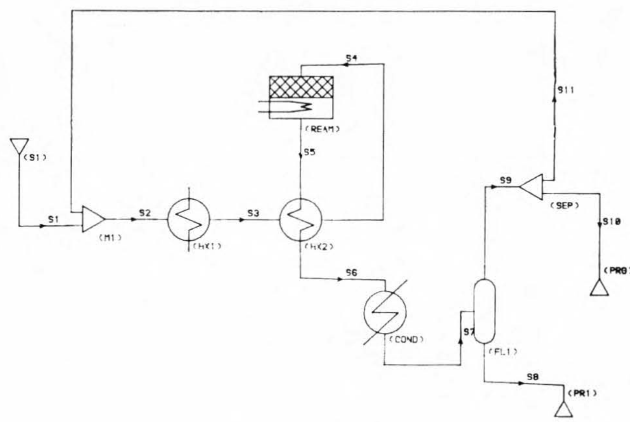


FIGURE 3

of multiple-effect evaporator design and the extension to more elaborate evaporator systems.

When using MEEVAP the first exercise is one in which the student initially determines by a hand method the area required for a three effect evaporator to carry out a particular duty when using steam at a specified pressure. (Each student has a different set of data). The student introduces the same data to the program using a dialogue which validates every item and at the end of the data input session is requested to provide the calculated answer. If this is within $\pm$ 5% of the (unrevealed) computed solution then the next phase of the exercise can be commenced. If the student's solution is outside the given tolerance the data is filed and he or she is invited to check their working and, if necessary, see a demonstrator for further advice. After three failures a lock is set to prevent access to the program until the tutor has discussed the problem again and released the lock.

The continuing exercise may take any direction desired by the tutor since the student can start from any one run and alter the evaporator arrangement, number of effects, callandria area or any other design or operating variables. From the results obtained the student examines the economics of various alternatives and ends up with a design which satisfies some agreed criterion.

With MEESYS, access to which may be contingent upon demonstrated successes with MEEVAP, the student can explore the economic benefits of introducing heat-exchangers to preheat the feed and/or the addition of condensate flash recovery.

Automatic optimisation is deliberately excluded from these packages but the student needs to be able to compare several designs so an essential component of the package is a facility to generate tables which collate nominated variables for a sequence of runs. Fig. 4 is an example of

NR	FS	NE	AE	PS	ULE
6 BAC	0.581	5.00	65.0	257.	2.50
5 BAC	0.660	5.00	65.1	273.	7.50
7 BAC	0.665	4.00	65.0	254.	2.50
4 BAC	0.702	5.00	72.9	240.	7.50
3 BAC	0.772	4.00	71.2	240.	7.50
2 BAC	0.910	3.00	69.6	240.	7.50
1 FOR	0.984	3.00	69.7	240.	7.50
NR	FS	NE	AE	PS	ULE

FIGURE 4

such a table in which the runs have been ranked in order of increasing flow of steam to the first effect.

EXPERIENCES AND OBSERVATIONS

After two years experience at using the package at UMIST and some attempts to transport parts of it to other departments with different systems we can comment on some aspects which seem to us important.

There is no doubt that the impact made by a well-prepared interactive exercise can be substantially greater than a conventional tutorial. The opportunity for the student to examine aspects of design which calculation time would hitherto have prevented is particularly welcome. We would certainly endorse the experiences of those such as our colleagues at Illinois, who find the PLATO system so effective. However, we also recognise that a considerable effort is required to generate the program, exercises and support documentation needed for each module in the packet. We believe that the only solution to this problem is to ensure that the packages receive wide distribution and use by generating modules to agreed standards and by enlisting the co-operation of several authors in many departments so that modules may be exchanged.

The original version of the package was written, as far as possible, in strict FORTRAN4 (ANSI 66) but we have now rewritten it to conform to strict FORTRAN77 (having looked at PASCAL and rejected it). There are still several machine dependant features, however, connected with file creation, opening and closing, I/O procedures, graphics or quasi-graphics. Our solution to the problems in these areas is to confine them to a few sub-routines which have been coded so that the local peculiarities may be incorporated during the initial setting up. Particular attention has been paid in writing the I/O procedures to making the dialogue "user friendly' and to trapping spurious or invalid responses.

The students need to be prepared for this type of tutorial work by providing them with concise documentation which sets out clearly the initial objective and encourages them to extend the exercises in an imaginative way. We arrange to schedule the terminals so that their use coincides with other practice and laboratory classes and find sessions of about 45 minutes to be about optimal. □

WILHELM'S INFLUENCE

Continued from page 15.

gave in Montreal in 1961 so much sums up his work that it is worth examining in a little more detail. The flavor of the argument and its presentation will have been gained from the lengthy extraction which was quoted above; here I would like to lay out its main lines. After an historical introduction Wilhelm divided the field of reactor studies into three parts. These were classified according to the difficulty of applying design objectives and the first was that part which seemed beyond immediate *a priori* design objectives because of incomplete theory. The second was the domain of moderately well-understood processes and the third was that part which was already developing rapidly. In 1962 he saw flame theory with its rapidity of reaction and complicated thermodynamics and other systems in which turbulent fluid dynamics plays a part as in the first category. Fortunately, the second category was larger and comprised most of the conventional reactors for which sensible models had been constructed. The incorporation of some of these reactor types into the third area of well-understood systems was the thrust of his own concern and, as he said, "I have chosen the packed bed reactor, a device which clearly falls in the third category, as the specific subject of my paper."

He commented that historically the packed