Computerised Choice and Negotiation in the National Certificate.

Phillip John Burton

Thesis submitted for the degree of Doctor of Philosophy, in the Department of Education, Faculty of Social Science, University of Glasgow.

© Phillip J. Burton October 1993

ProQuest Number: 13818877

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 13818877

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Thesis
9963
Copy 1

GLASGOW
UNIVERSITY
LIBRARY

Abstract

One of the original aims of the National Certificate was that it would encourage more students to participate in further education by offering flexibility and choice. Students of differing interests and needs would, through the process of negotiation and by the exercise of choice, achieve a greater sense of 'ownership' of their courses, thereby enhancing their motivation to succeed, and would develop a greater sense of responsibility for their own actions. Choice and negotiation were to be concerned with a number of aspects of the student's studies; this work focuses on the way in which the student should participate in the construction of his course by choosing the units of study, the modules, which would make up that course.

It is the contention of this work that choice and negotiation are largely denied to the majority of students. Studies of the extent of choice in the National Certificate have been summarised, and the reasons for the lack of choice identified. It is argued that some of the problems associated with choice can be overcome by the use of a properly designed computer system. The factors influencing the design of such a system are described. A prototype program simulating the operation of key parts of the system is described and evaluated.

This work contributes to the literature on the National Certificate, in respect of choice and negotiation, demonstrates the feasibility of the proposed system in terms of facilitating choice and encouraging negotiation, and details the design criteria on which the system should be built.

Contents

Appendices

List of Figures

List of Tables **Page**

Accompanying Material

51/2 " floppy disc, containing programs and files used in the prototype described in Chapter 7.

The programs run in a BBC Master 128 micro-computer. The disc is formatted to operate in ADFS (Advanced Disc Filing System). A double-sided, 80 track disc drive is required.

Acknowledgement

The author would like to thank Mr. W. Dunn and Mr. H. Paterson, of the Department of Education, Glasgow University, for their help, advice and encouragement during the course of this project, and Mr. K. Sangrar for his advice on the use of Prolog.

Chapter 1

The Introduction of the National Certificate

1.1 1979. The Consultative Paper

In 1979 the Scottish Education Department issued a Consultative Paper (SED 1979) which highlighted various concerns in connection with the provision of non-advanced further education. The phrase 'non-advanced further education' includes courses

> *"...in which the standard of instruction does not extend above that required for: (a) the Ordinary National Diploma (OND), Ordinary National Certificate (ONC), Scottish National Certificate (SNC),...or*
> *(b) the Scottish Certificate of Education (SCE Higher Grade and Certificate of Sixth Year Studies) ...."* (SED 1985a p5)

One concern centred on the small number of young adults in full and part-time education. This number is an important indicator of a nation's economic health. Industry needs a continuous supply of trained workers to allow it to expand and develop, and when the traditional heavy industries were being run down in favour of the more modern, more technologically oriented light industries, it was increasingly important that there should be a well-trained, well-educated, flexible workforce, able to adapt to new conditions and assimilate new skills.

While the importance of a well-trained workforce can be readily understood, it has proved more difficult to find the evidence which prompted the Government's apparent belief that the country was failing to provide this workforce. Some figures have been obtained to support the view that the United Kingdom as a whole was falling behind other countries in the provision of further education. However, as the justification of the Government's view is not within the scope of this work, and as the data are, to an extent, unsatisfactory, they are reproduced in Appendix 1, rather than here. The figures give some justification to the view that, while other countries were increasing the number of students in further education, the United Kingdom was not keeping up with them.

The Government felt that the existing arrangements in further education were not attractive to or suitable for many potential students, and that steps should be taken to remedy the situation. In the early seventies the minimum leaving age for compulsory schooling had been raised and, while some attempts had been made to provide satisfactory schooling for the pupils affected, by the mid-seventies it had been realised that more pupils were leaving school disenchanted with their educational experiences. The same young adults who had most to gain from involvement in further education were increasingly less inclined to volunteer to continue their studies. The problems in secondary education were being addressed (Dunning 1977, Munn 1977). The time was right for an equivalent review of further education provision.

At the time, further education courses were generally offered by further education colleges run by local authorities. The courses were certificated or approved by a number of organisations, for example, the City and Guilds of London Institute, the Association of Accounting Technicians, the Chartered Insurance Institute, the Institute of Bankers of Scotland, and many more - in 1985 the SED listed one hundred and thirty-one "professional institutions and examining bodies" (SED 1985a p96). This proliferation of certificating bodies led to duplication of teaching and confusion on the part of potential students.

Most of the courses offered were of a vocational nature, and, at a time of rising youth unemployment, it was felt that other courses of a less overtly vocational nature were required.
A young person, temporarily unemployed, could easily lose motivation, self-esteem and the habits associated with regular employment, e.g. punctuality. Attendance at a college and success in the courses attempted would help to keep the person 'employable' and prevent the slide into long term unemployment.
Continuing education in this way would also allow young people to re-train as required by the job market.

1.2 1983. The Action Plan

The Consultative Paper was followed by the "Action Plan" (SED 1983a) This reiterated some of the concerns of the earlier paper and developed a structure which would allow further education[1] to meet the expected needs of students in a more appropriate way. The Action Plan[2] laid down what had to be done, and discussed why it should be done; who would be responsible for doing it; and when the various tasks should be completed. It not only laid the foundation for what was to become known as the National Certificate[3]; it also erected the main structure within which the Certificate would be delivered.

Early in the document, the Action Plan set the tone for the discussion to follow;

"The paper is thus not prescriptive, but neither is it simply part of a further consultative process." (Ibid., p3 1.6)

Because of the importance of the Action Plan in determining the principles underlying the National Certificate, and because these principles are of fundamental importance to this work, it will be quoted liberally.

The discussions following the Consultative Paper had resulted in a general acceptance that change was needed.

"The paper was well received and the responses reflected a desire for reform. They did not, however, reveal any consensus of opinion on the changes which were necessary to make better and more co-ordinated provision for the age group. " (Ibid. p2, 1.3)

It was, perhaps, because of the lack of agreement on the direction of change that the Action Plan took the lead. At the risk of reading too much into it, the Plan appeared to be saying;

a) you all agree that something must be done;

b) but you can't agree on what to do;

c) so we'll tell you what you will do.

1.2.1 The need for reform

The Plan recognised two reasons for reform;

a) The existing arrangements were too complex.

"A coherent provision of education and training for 16-18 year olds, covering all levels of attainment and motivation and embracing schools and colleges, does not exist at present. At 16 years of age young people are faced with a bewildering choice from a large number of Higher and Ordinary grade courses, scores of certificates and diplomas awarded by different examining bodies, and other forms of provision in industry and the community." (Ibid. p26 para.3.26)

b) The existing arrangements were too vocationally specific.

"In further education, whether full-time or part-time, the courses have been devised in response to specific occupational requirements. At the present time of rapid technological development and occupational mobility, when the future is uncertain, a review is required."

(Ibid. p30 4.6)

1.2.2 The way ahead

The Plan explained in broad terms the direction in which change should proceed;

"The rate and extent of change which is currently being experienced in society will require a considerable measure of flexibility in developing new provision in order to avoid the dangers of obsolescence."

(Ibid. p5 2.3)

".....a more general approach, less specifically occupation-orientated and including both training and education, would offer considerable advantages." (Ibid. p16 3.11)

Thus the new structure was to offer flexibility and a wider range of activities than was previously the case.

1.2.3 Responsibility

In designing this new structure, there was a need to recognise the nature of the students who would be involved. Students had to be seen as young adults;

> *"By the end of this stage some will have taken decisions which will determine their careers and their life-styles, some will have developed their competencies to the point of earning a living, and some will have assumed full responsibility for their own well-being, possibly also for that of others."* (Ibid. p4 2.1)

and further education had a part to play in preparing them for adulthood.

> *"It follows that within the learning experience a deliberate effort must be made to foster self-confidence and self-reliance, to encourage autonomy, to exercise independent responsibility, and to ensure that young people understand how to gain access to advice or information which they require. "* (Ibid. p7 2.8)

1.2.4 Motivation

Apart from being desirable in its own right, encouraging students to exercise responsibility was seen as a motivating influence, giving the student a feeling of involvement in and 'ownership' of his course of study. The effect, on motivation, of involving the student in the design of his course had been noted in an earlier work;

> *"Motivation of the young person by giving him/her some real responsibility in the planning of his/her programme must be regarded as an essential element of his/her development. "* (FEU 1981 p12)

This principle was adopted by the Action Plan, which clearly linked it to the provision of choice and the process of negotiation.

> *"A young person's motivation and willingness are likely to be improved if he has some measure of choice in what he has to do, how it is to be done, within what time-scale and up to what standard. Motivation will depend on the extent to which learning experiences can be negotiated, and it will increase in proportion to achievement and visible results."*
> (SED 1983a p6 2.5)

Including a wider range of activities, already seen as desirable, could also play a part in encouraging more young people to participate in further education, and help to motivate those who thus became involved.

> *" Many young people ... have a range of personal interests and ambitions which are quite separate from academic and vocational aspirations, and, given opportunities to pursue these interests informally, they may find other aspects of their courses more acceptable. "* (Ibid. p10 2.12)

1.2.5 Adults

The same principles, of increasing scope and flexibility, would also be of benefit in allowing for 'continuing' education. Industry had changed, but it was recognised that it would continue to change, and further education had a role in helping the workforce to adapt to these changes by developing new skills in existing occupations, and by retraining as older industries contracted and newer types of work were introduced.

> *"Inevitably, further education will also be affected by the development, particularly through open learning methods, of opportunities for adults to undertake refresher, updating and occupational conversion courses and otherwise to engage in recurrent education. "* (Ibid. p30 4.6)

1.2.6 The modular system

Thus, the four characteristics which would distinguish the National Certificate from the previous system were to be choice, negotiation, flexibility, and a wider range of subjects. These would be made possible by adopting a 'modular' approach to course design, subject or skill areas being broken down into units, these units being assembled to give programmes of study to cater for the needs of the individual students.

> *" ..... it will be necessary to design curriculum components, or modules ....., which are sufficiently flexible to be built into individuals' programmes in a large variety of ways, and by a process of negotiation. "* (Ibid. p31 4.8)

1.2.7 Programme design

The main principle to be considered in the design of these modular programmes was summarised as follows;

> *"The education system should be designed, therefore, to enable all young people to undertake studies which are related in content and level of demand to their past achievements, present needs and prospective employment."* (Ibid. p8 2.9)

'Past achievements' were identified with the qualifications obtained by the student while at school;

> *"....it will also allow young people to negotiate individual programmes of study, the entry to which will depend on and develop from their school-based attainments, ..."* (Ibid. p46 4.33)

It can be assumed, given the need to cater for adults returning to further education, that 'past achievements' would also include attainments in earlier periods of further education, although the Action Plan does not specifically say this.

'Present needs' would include consideration of leisure or recreational activities;

> *"Each young person should be helped to construct his own curriculum through a process of negotiation, not only with the immediate session in mind but also planning carefully for the future, and taking account of activities and interests pursued outwith the educational system in leisure time."* (Ibid. p 45 4.32)

It is not clear, from this, just how account could be taken of something happening outside the system; the quotation given earlier, and reproduced here for clarity, is also slightly ambiguous.

> *"Many young people ... have a range of personal interests and ambitions which are quite separate from academic and vocational aspirations, and, given opportunities to pursue these interests informally, they may find other aspects of their courses more acceptable."* (Ibid. p10 2,12)

If the word 'informally' had been omitted there would have been no doubt as to the meaning; leisure and recreational activities were to be accepted as valid parts of a student's course. As written, these statements could be taken to mean, either, that students could include leisure pursuits as part of a course, but they would not be assessed; or they could mean that students were to be encouraged to organise and participate in sports and other clubs within the institution. This is the only area, of those relevant to this work, where any ambiguity existed.

'Future needs' would obviously include the need to find employment; but it was also recognised that, for some students, further education would be a stepping stone to more advanced studies, and this would have to be taken into account.

> *"Flexibility in provision and choice of course will also be an advantage so that young people, depending on their circumstances, may negotiate programmes which will allow them to aim for higher attainment levels, find alternatives to original choices of institution, or find faster or perhaps slower courses."* (Ibid. p10 2.12)

1.2.8 Summary of 1.2.2 to 1.2.7

Courses in further education would be constructed by assembling a number of modules, chosen to suit the needs of the student.
Account would be taken of the student's existing qualifications and future intentions, and would prepare him for employment or more advanced work.
Leisure and recreational modules might be included in the programme.
Students were to be actively involved in choosing modules, through a process of negotiation.
The modules offered would include subject areas previously outside the scope of further education, with provision being made to change them to suit the changing needs of industry.

1.2.9 The timetable

The Action Plan included details of the timing of the tasks to be undertaken for its implementation (Ibid. p60 Chapter 7).
The Action Plan was published in January 1983, and no time was to be lost; the timetable started with "January 1983". This in itself showed that the

Action Plan was not a series of recommendations; no time was to be available to discuss it. The first modules were to be made available "after July 1984". These would be the first modules as part of the National Certificate; modules were to be used earlier than this as an interim measure in certain courses which were due to start in August 1983. Before modules could be written, detailed guidelines on content, assessment procedures and teaching methods had to be agreed. Bearing in mind the need to prepare for the courses mentioned above, less than one year would be available in which details of the modules, and student material, could be written. It was not intended that all courses should change over to the new system immediately, but, given the condemnation of the existing provision, there would be pressure to adopt the new approach as quickly as possible.

1.2.10 Resources

There was some criticism of certain existing practices in further education, in particular class sizes and the duplication of courses or subjects both within and between colleges (Ibid. p47 Chapter 5). The education authorities were charged with the responsibility to reconsider the provision of courses within its area, and to ensure that class sizes were increased to make efficient use of staff and accommodation. The belief expressed in the Action Plan was that the modular approach to course construction would be a contributory factor in allowing better use to made of resources; hence, in discussing the resources required for the implementation of the Plan;

> *"The Government consider that it should, therefore, be possible to introduce the new framework in accordance with its programme and to meet the needs of staff preparing the new courses without additional provision. "* (Ibid. p 66 7.12)

1.2.11 The New Arrangements

The new structure was described in the following terms;

1) Certification would come under the control of one body, the Scottish Vocational Education Council (SCOTVEC)(4), which would develop, administer and moderate the assessment of a new 'National Certificate'.

2) Any programme of study would be constructed by selecting a number of modules from a large 'catalogue'(5) of modules.
3) The National Certificate would list all the modules completed by a student and allow a prospective employer to judge whether the programme met his requirements.
4) A 'standard' module would involve approximately forty hours of teaching. However, variations on this would be allowed, e.g. 'half modules', lasting 20 hours, or double modules lasting 80 hours.(6)
5) Modules would be offered at two levels, General and Specialist. General modules would be relevant to a number of courses of both a vocational and a more 'general education' type. Specialist modules would have more restricted applicability, probably to the course for which they were originally designed (but with no bar against them being used elsewhere where appropriate).
6) In some cases, a specific combination of modules would be recognised as equivalent to a course previously certificated by a different body. These equivalences were to be agreed between SCOTVEC and the previous certificating body.

1.3 The aim of this work

It is the intention of this work to;

a) show that the flexibility made possible by the modular system has not been fully realised;
b) show that choice and negotiation have not figured to any great extent in the modular programmes followed by most students;
c) suggest reasons for this failure to allow choice;
d) develop and describe a system, a 'tool' which would encourage the provision of more choice and allow more flexibility in the design of courses suited to the individual's needs.

1.4 The National Certificate in 1985/86

The details given below are intended to give the reader general information concerning the National Certificate in the year of its introduction into college courses. They are intended to show the scope and scale of the system, as an introduction to the discussions to follow. More data, particularly with reference to the changes in the catalogue following its introduction, will be given later at more appropriate points.

1.4.1 The Modules

When the first catalogue of module titles was published, in 1985(7), modules were listed in nine categories (Table 1.4.1.1, below).

Table 1.4.1.1
The Categories into which the Catalogue was divided, 1985
(SCOTVEC 1985)

01 Interdisciplinary Studies
02 Business and Administration
03 Distribution, Food Services and Personal Services
04 Engineering
05 Built Environment
06 Caring
07 Industrial Processing
08 Land and Sea-Based Industries
09 Pure and Applied Sciences

Within these categories modules were grouped under various headings, although the number of subdivisions varied; e.g. Category 01 was divided into twenty-three sections, while Category 04 was not divided at all.
Each module was given a five figure reference number consisting of the category number, e.g. 01, and a three digit identifier within the category. Modules were listed as G (General) or S (Specialist), as stated above.
Modules were listed in three groups;

a) those available as at May 1985
b) those that would be available by September 1985
c) those at an early stage of development

Groups (a) and (b) constituted the modules that were available for use in session 1985/86, and are used in the analysis of the catalogue, given below in Table 1.4.1.2.

Table 1.4.1.2
Sub-divisions of the SCOTVEC Catalogue for session 1985/86
(Data compiled using SCOTVEC 1985)

Category	(A)			(B)			(C)		
	G	S	G+S	G	S	G+S	G	S	G+S
01	176	142	318	10.6	8.5	19.1	6	49	55
02	17	67	84	1.0	4.0	5.0	0	0	0
03	17	62	79	1.0	3.7	4.7	0	0	0
04	119	279	398	7.1	16.7	23.9	3	20	23
05	40	184	224	2.4	11.0	13.4	19	37	56
06	6	45	51	0.4	2.7	3.1	0	2	2
07	42	114	156	2.5	6.8	9.4	1	10	11
08	58	220	278	3.5	13.2	16.7	0	19	16
09	35	45	80	2.1	2.7	4.8	0	9	9
Totals	510	1158	1668	30.6	69.4	100.1	29	146	175

Table (A) shows the number of modules in each category, divided into G (General) and S (Specialist) and G+S(Both).
Table (B) gives the same numbers expressed as percentages of the total number of modules in the catalogue.
Table (C) gives the number of modules being prepared for inclusion in the catalogue at a later date.

Thus 1,668 modules were available for use in session 1985/86.

175 modules were being prepared for later inclusion in the catalogue.

The specification for each module was given in a 'Module Descriptor' (see Appendix 2.1), which listed, among other details;

a) the Learning Outcomes (what the student would know or be able to do after successfully completing the module);
b) a more detailed description of what should be learned, and the context in which it should be taught;
c) suggestions and recommendations as to learning and teaching approaches;
d) the Assessment Procedures; the detailed format of the test or tests which are to be used to assess the student on the Learning Outcomes.

When the National Certificate was introduced it did not immediately replace all previous courses; it took time to replace these courses with modular equivalents. In addition, many students were already enrolled for the old style courses when the National Certificate started. However, as suitable modules became available, students enrolled on the old type of course frequently enrolled for a small number of modules as well. Some students were therefore counted twice when statistics on further education were being compiled. The Government Statistical Service, who provide these statistics, warned of this (SED 1989 p2). Coinciding with the introduction of the National Certificate the Service also changed the basis on which they counted students. It therefore warned that figures for student participation in further education before and after the introduction of the National Certificate are not generally comparable (Ibid.)

1.4.2 The Students

In 1985/86 177,466 students were registered as participating in non-advanced further education (Ibid. p7).

96,646 were enrolled for the National Certificate. Of these 46,920 (48.5%) were male, and 49726 (51.5%) were female. (SED 1990 p19).

Students enrolled for 687,941 modules, the success rate being 80.1% (Ibid.). The distribution of module enrolments among the categories is given below in Table 1.4.2.1.

Table 1.4.2.1
Module enrolment by Category, 1985/86
(SED 1990 p19)

	(a)	(b)
01	184,306	26.8
02	141,718	20.6
03	56,943	8.3
04	101,356	14.7
05	62,134	9.0
06	37,049	5.4
07	63,066	9.2
08	28,725	4.2
09	12,644	1.8
	687,941	100

For each of the nine categories, the table shows,
(a) the number of module enrolments
(b) the module enrolments as a percentage of the total module enrolment.

Students of a variety of ages enrolled for National Certificate modules. Details are given below in Table 1.4.2.2.

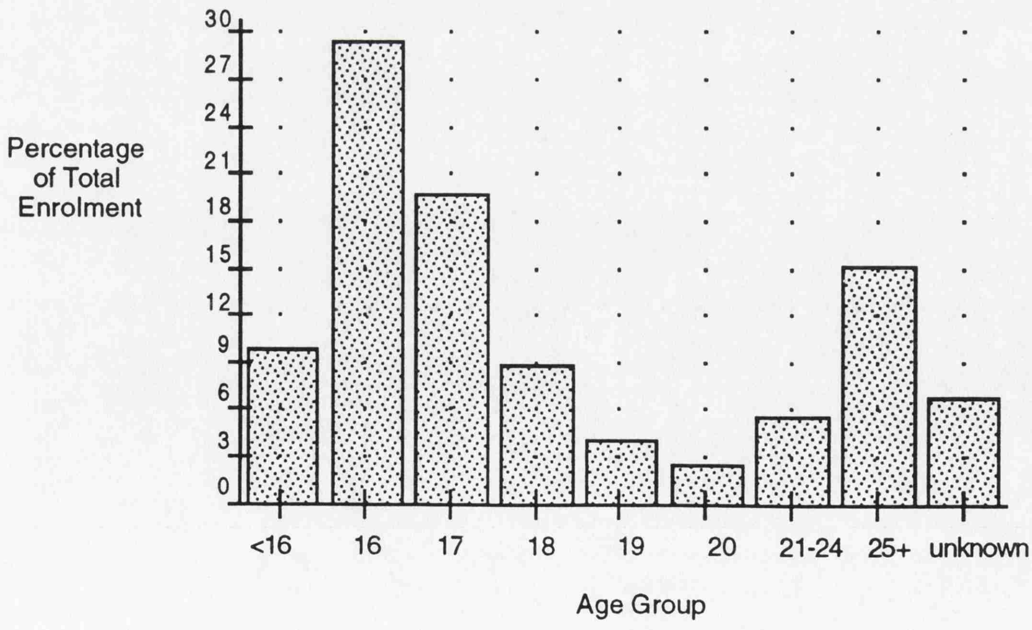
Table 1.4.2.2
Distribution of students by age, 1985/86
(SED 1990 p19)

	(a)	(b)
Under 16	9,246	9.6
16	28,087	29.1
17	18,883	19.5
18	8,450	8.7
19	3,648	3.8
20	2,268	2.4
21-24	5,260	5.4
Over 24	14,460	15.0
Not Known	6,344	6.6
	96,646	100.0

The table shows
(a) the number of students of a particular age or within an age range, and
(b) that number as a percentage of the total number of students.

These figures are shown diagrammatically in Figure 1.4.2.1.

Figure 1.4.2.1
Distribution of students by age
1985/86



1.5 Notes

1) The Action Plan also recommended changes in the later stages of secondary schooling. These changes are not relevant to this work, except to note that it was recommended that National Certificate modules should be made available in secondary schools.
2) For the sake of brevity, phrases such as 'the Action Plan laid down', or 'The Action Plan said' will be used in preference to the more accurate, but more cumbersome, 'the writers of the Action Plan laid down', or 'the authors of the Action Plan said'.
3) The National Certificate was (and still is) the name of a qualification; in this context, however, the name will also be used, somewhat loosely, to mean 'the system designed to lead to the award of the National Certificate'. This cannot be shortened, accurately, to 'further education' - see note (1) above.
4) The Scottish Vocational Education Council will be referred to throughout this work as SCOTVEC.
5) The 'catalogue' referred to here, and hereinafter referred to generally as 'the SCOTVEC catalogue', contained the reference number, title, length (expressed in terms of the standard module) and the type (General or Specialist) of every module available for use in the session following the date of publication.
6) Other variations were found to exist, e.g. $1^1/_2$ modules or 60 hours.
7) Although modules were offered in session 1984/85, 1985/86 is taken as the start of the National Certificate for this work, this being more representative of the system thereafter.

   The first programmes under the new arrangements were taught in FE colleges and (to a limited extent) in schools in 1984-85; the first full year of implementation so far as FE colleges were concerned was 1985-86 (SED 1988 p19 Note 1).

Chapter 2
Factors affecting Course Design

2.1 The Course Grid

Since the introduction of the National Certificate it has become common for further education colleges to divide each session into three thirteen week blocks. A 'normal' module, lasting nominally forty hours, is studied for three hours per week, giving a total of thirty-nine hours study during the block. Up to eight such modules can be studied during the block, giving twenty-four modules per session.

Modules can, however have other lengths - half modules, one-and-a-half modules, double modules, for example, are all possible (1).

The programme for a complete session can be represented diagrammatically as a grid comprising eight rows and three 'blocks' which can be further subdivided into two columns each. All the variations in module length can be accommodated in a grid as shown in Figure 2.1. This grid is referred to as the Course Grid in later discussions.

Figure 2.1
The Course Grid

	Block 1		Block 2		Block 3	
	1	2	3	4	5	6
1	A	A	B	B	C	C
2	D	E	F	G	G	G
3	H	H	H	H	H	H
4						
5						
6						
7						
8						

Each cell can contain the equivalent of a half-module, which is a module lasting half the recommended forty hours. In the context of course design it is more convenient to use this as the basic unit of length, rather than the forty hour standard module.

Row 1 shows three 40 hour modules, one taken in each thirteen week block. Each module appears twice, representing the two half-modules of which it may be considered to be composed.

Row 2 shows two twenty hour modules in the first block. This representation requires some explanation.

The figure is not a timetable in the conventional sense; module E does not have to follow module D. D and E could be taught at different times during the week, with only half the time being spent on each during the block. That arrangement would have the disadvantage that module D could not be an entry requirement to module E, as an entry requirement must be completed before the module to which it refers can begin.

A number of other possible module lengths are shown in the figure; for example, module H is a triple length module.

The model described above, consisting of three thirteen week blocks, is not the only way in which colleges organise the students' work. However, as it is a fairly common model it has been adopted as the standard throughout this project. Any column in the Course Grid contains the modules that will be studied at the 'same' time, for example, in the course of a week. The number of rows shows the number of modules that can be studied in this time. The total area of the Grid gives the maximum number of modules that can be studied in one session or its equivalent. Arrangements different from that adopted here will have their own equivalent Course Grid. The principles developed in this work should therefore be generally applicable.

Because of the variation in module lengths, programmes may differ considerably in the number of modules they contain. The two extreme cases are;

a) a programme consisting entirely of half-length modules, which would involve forty-eight separate modules;
b) a programme consisting entirely of triple length modules, which would involve eight modules.

It is unlikely that a real programme would fall at either of these extremes; the more typical programme would consist of approximately twenty-four single-length modules, with a few other modules of different lengths. Nevertheless, the variation in module length is a factor which must be taken into account when designing the programme.

2.2 Time available to the Student

A full-time student may take up to twenty-four forty hour modules in one session, but many students are part-time. This is reflected in the number of modules taken per student.

Table 2.2.1
Number of Modules taken per student, 1987/88
(SED 1989, p11)

Number of modules taken	FE College Students
1	23,558
2	13,077
3	7,790
4	5,201
5-9	27,103
10-14	11,927
15-19	6,422
20-24	6,271
25-29	3,975
30+	1,470

The average number of modules per FE College student was 7.7 (Ibid., p2)

These figures require some explanation.

a) Some modules last longer than the 'normal' forty hours, and even if the student is studying full-time, he would not able to take the full twenty-four in one session. The table below shows the distribution of modules lengths in the main categories, in the 1987-88 catalogue.

Table 2.2.2
Variation in Module Lengths
1987-88 Catalogue
(SCOTVEC 1987)

Category	Module length 0.5	1	1.5	2	2.5	3	4	Totals
1	42	314	0	78	5	5	0	444
2	14	70	0	9	0	0	1	94
3	24	50	0	4	0	0	0	78
4	147	305	0	21	0	0	0	473
5	119	267	2	22	0	0	0	410
6	14	29	0	15	0	6	2	66
7	49	131	2	6	0	0	0	188
8	111	174	15	8	0	0	1	309
9	30	49	0	5	0	4	0	88
Totals	550	1389	19	168	5	15	4	2150
Percentage of total	25.6	64.6	0.9	7.8	0.2	0.7	0.2	

The number of modules lasting more than the forty hours is 211, which is only 9.8 % of the total catalogue. By contrast, nearly 26 % are half the normal length, which would result in students taking *more* than the standard number of modules. Clearly, while the variation in module length does affect the number of modules taken by the students, it does not explain why the average number of module enrolments is so much smaller than the expected twenty-four.

b) A more likely explanation lies in the 'mode of attendance' of students, i.e. whether they study modules full-time or not.
The table below shows the number of students attending in the various modes (ibid., p14).

Table 2.2.3
Module Enrolments and Mode of Study
(SCOTVEC 1987)

Mode of study	Students	%	Total Module Enrolments	%	Modules per student
Full-time	65,999	41	478,255	50	7.3
Sandwich	1,518	1	21,197	2	14.0
Short full-time	6,023	4	22,252	2	3.7
Block Release	13,167	8	114,483	12	8.7
Part-time day	56,094	35	289,837	30	5.2
Evening	15,675	10	29,172	3	1.9
Directed Private Study	2,662	2	6,846	1	2.6
Totals	161,138	100	962,042	100	6.0

(Because the percentages have been rounded to integers, they do not sum to 100%)

c) Some students may be enrolled on the 'old style' courses, e.g. City and Guilds, and take a module or modules as additions, e.g. a sports module or a recreational module, rather than as an integral part of their courses. Similarly, students at college to sit O- and H-grade examinations may fill out their timetables with some modules. For example, see Appendix 3.1, Case 3.

The explanation for the relatively small number of modules taken by each student lies in a combination of (b) and (c) above.

2.3 Choice

In designing the programme, two key questions that immediately arise are

a) "How should modules be selected for inclusion in a programme for a specific student"?

b) "Who selects the modules."

(a) is discussed below. Given the importance attached to choice and negotiation by the Action Plan, the question of, "Who chooses", and, "What should be negotiated", are crucial. Before discussing these points it is necessary to know what has to be chosen, what factors influence choice and what difficulties may be encountered. The discussion of choice and negotiation have therefore been deferred to Chapter 5, by which time these questions will have been answered.

2.4 Course characteristics

Four key characteristics of a good programme have been identified (CAST 1987, pp20-23). These are;

a) Flexibility, allowing a programme to be modified according to the needs of the user;
b) Choice, involving the student in the selection of modules;
c) Breadth, allowing the student to develop broad transferable competencies relevant to a range of jobs;
d) Integration, helping the student to integrate learning from various parts of the programme.

However, of these, three are more descriptions of the modular approach rather than of a programme constructed from modules. Flexibility and integration follow from the decomposition of 'subjects' into modules; flexibility by allowing many different programmes to be constructed from the modules, and integration by designing, for example, a module in basic arithmetic which can then be used in any programme requiring that knowledge. Choice, in the given sense, is not a characteristic of the programme, it is a statement of 'who will be involved' in selecting the modules.

The only 'key characteristic' from the list that does indicate the components of a programme is breadth, implying that the programme should not consist entirely of modules of a specifically vocational nature.

2.4.1 Programme Breadth

The programme should allow some experience not directly related to vocational needs, perhaps, for example, by including the "Learning and Study Skills" mentioned earlier. It has also been suggested that all programmes should contain Communication, Personal and Social Development, Numeracy and Computer Literacy. (quoted in CAST 1987, p38)

"Specialisation within the vocational area should be deferred ."

(CAST 1987, p22).

Although a student may be working within a broad vocational area, he should be allowed to experience the wider field before being asked to specialise in a narrower, more specific area.

2.4.2 Articulation

In addition, the Action Plan used the word 'articulation' to mean the process by which syllabuses supply progression from basic to more advanced stages, and that this was desirable. This implies that the National Certificate should lead on to more advanced education, but also that, within a modular programme, there should be a progression from earlier to later modules. Modules may be arranged sequentially for one of three reasons;

a) Two modules, A and B, may be sequential because A is a necessary pre-requisite for B - referred to in the Module Descriptor as a 'preferred entry level'.
b) Modules might be studied consecutively purely for convenience; seventeenth century history need not always be taught before eighteenth century history, except if the latter demands knowledge of the former in order to understand it - in which case, see (a), immediately above.
c) Some modules, perhaps not a large part of the student's programme, have a logical position in the course of their content. For example, "Learning and Study Skills", is recommended to come early in a course because it is designed to help the student study, while others, perhaps designed to prepare the student to enter work, should come towards the end of the course. The module, "Life and Work", is an example.

A modular programme can therefore be considered as an array of modules in which some are in a fixed relationship to others, (a) above, some are only fixed in relation to the entire programme, (c) above, and others have no required position, because they neither have entry requirements nor are they specified as entry requirements to other modules, (b) above.

A number of factors influence the choice of modules in a programme :-

2.5 The Student's Previous Qualifications

2.5.1 The Student's Age

Students can arrive at the college from a variety of educational backgrounds. Many will come directly from school; some will have worked for a time before resuming their education; others will be sent by an employer on block release; etc.

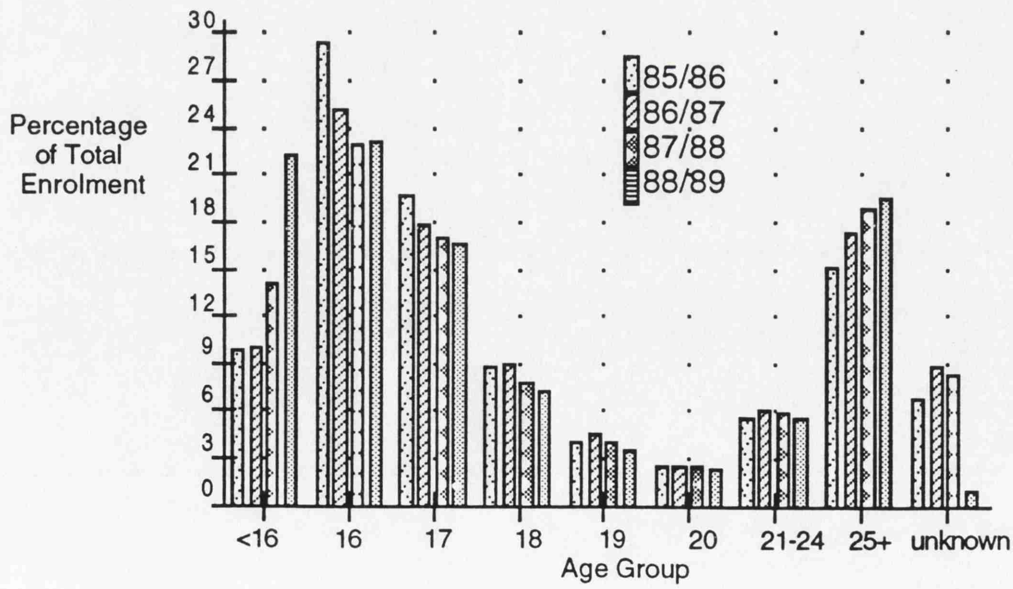
The distribution, by age, of students attending college is shown in Table 2.5.1.1 and Figure 2.5.1.1.

Table 2.5.1.1
Distribution of student enrolments by age, 1985-1988

	Age								
Session	Under 16	16	17	18	19	20	21-24	25+	Not known
85/86	9.6	29.1	19.5	8.7	3.8	2.4	5.4	15.0	6.6
86/87	9.9	25.0	17.7	8.9	4.3	2.3	6.0	17.2	8.7
87/88	14.0	22.8	16.9	7.7	3.8	2.4	5.7	18.7	8.1
88/89	22.2	23.0	16.5	7.2	3.4	2.2	5.3	19.4	0.9
Change	+12.6	-6.1	-3.0	-1.5	-0.4	-0.2	-0.1	+4.4	-5.7

The table shows, in each session, for each age or age range, the number of students in that age/range as a percentage of the total number of students enrolled.
The figures have been calculated from data in SED 1988 p7, SED 1989 p8 and SED 1990 p7.

Figure 2.5.1.1
Distribution of student enrolments by age, 1985-1988



From these data, the following observations may be made;

a) The biggest change was for students under the age of sixteen. As the vast majority of these students would be school-based, the implications of this will be discussed later in this chapter under the heading, "Schools Participation in the National Certificate".
b) The change in the 'not known' category presumably reflected an improvement in registration procedures.
c) The largest single proportion was that of the sixteen year-olds, who may be assumed to have left school with few, if any, formal qualifications.
d) The proportion of students in the '25+' category rose slowly but steadily over the years shown, and was approaching that of the largest group.
e) The sixteen and seventeen year-olds together constituted about forty percent of students enrolled for National Certificate modules. Some of these students would have been taking modules for the first time, whereas some of the seventeen year-olds would have been continuing into the second year of an F.E. course.

Students clearly cover a spectrum of ages, but they may be considered to form three main groups, categorised by their previous educational experience;

a) school-based pupils;
b) students straight from school, possibly after the minimum of secondary education;
c) adults returning to education after some time elsewhere.

It is likely that these groups have different reasons for enrolling on the National Certificate, and this has implications for the programmes offered. These will be discussed later in this chapter.

In terms of qualifications, pupils leaving school can have awards (at various levels) in SCE Ordinary grade, Standard grade, Higher grade and Certificate of Sixth Year Studies, in a variety of subjects. At the age of 16

years, pupils are likely to have only O- or S-grades.
Older students can have similar qualifications, with the possible addition of the 'old style' college certificates (City and Guilds etc.) and, as the National Certificate matures, may have completed some modules during a previous attendance at college.
Alternatively, students of any age may have no formal qualifications at all, having left school at the earliest possible time and gone into employment (or unemployment).

In devising a programme of modules for a student, these variations in 'previous' qualifications must be taken into account.

2.6 Schools Participation in the National Certificate

SCOTVEC modules are increasingly used in schools; this was one intention of the Action Plan, to break down the distinction between 'education' in schools, and 'training' in further education colleges(2). Tables 2.6.1 and 2.6.2 and Figure 2.6.1 contain information relevant to this point.

Table 2.6.1
Student enrolment,
comparing school with college, 1985-1988
(SED 1990 p19)

Part A	1985/86	1986/87	1987/88	1988/89
F.E. Colleges	77,800	96,241	106,794	113,072
Schools	15,848	23,017	46,785	74,519
All Centres	96,646	124,456	161,138	200,684
Part B				
F.E.Colleges	80.5	77.3	66.3	56.3
Schools	16.4	18.5	29.0	37.1
Colleges+Schools	96.9	95.8	95.3	93.4

Part A gives the number of student enrolments in school, college and in all centres, for each session.
Part B gives the number of student enrolments as a percentage of the total for all centres.

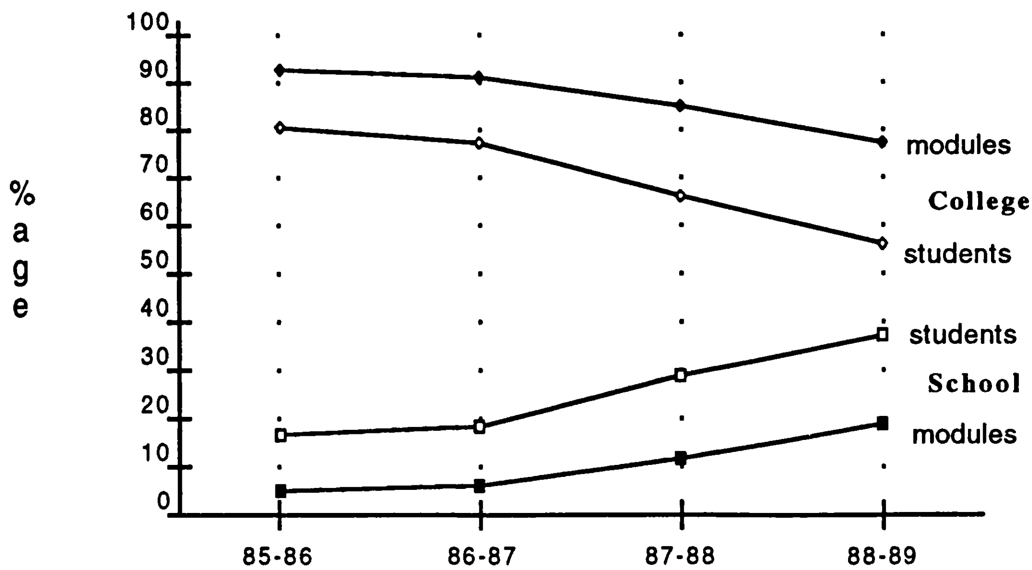
Table 2.6.2
Module enrolments, comparing school with college, 1985-1988
(SED 1990 p19)

Part A	1985/86	1986/87	1987/88	1988/89
F.E.Colleges	639,779	767,762	819,619	811,069
Schools	34,559	50,528	110,587	198,581
All Centres	687,941	840,807	962,042	1,052,525

Part B				
F.E.Colleges	93.0	91.3	85.2	77.1
Schools	5.0	6.0	11.5	18.9
Colleges+Schools	98.0	97.3	96.7	96.0

Part A gives the number of module enrolments in school, college and in all centres, for each session.
Part B gives the number of module enrolments as a percentage of the total for all centres.

Figure 2.6.1
Module enrolments, comparing school with college, 1985-1988
(SED 1990 p19)



The tables show considerable increases in the number of school students taking modules (Table 2.6.1), and in the number of modules taken by school students (Table 2.6.2).

This was because in the years following 1985 it was increasingly realised that the traditional certificated courses offered by secondary schools to senior pupils (S.C.E. Higher grade and Certificate of Sixth Year Studies), were not suitable for large numbers of pupils who, by that time, were staying on at school in larger numbers. National Certificate modules were seen as a suitable alternative to these traditional courses . It may be assumed therefore, that the number of students arriving at further education colleges with experience of modular courses will continue to increase, unless a more systematic revision of fifth and sixth courses in secondary schools removes this need.

Given that an increasing number of students have already successfully completed some modules at school, fixed programmes would be inappropriate as students would simply have to repeat modules - probably with adverse effects on motivation.
Students will also vary in the S.C.E. subjects and grades that they may have taken at school, and if the programme is not for their first session in college they will also have completed a number of modules in earlier sessions.

2.7 The Needs and Interests of the Student

It was shown, in Table and Figure 2.6.1, that the number of older students has been increasing steadily since the National Certificate started(3). In 1987 a survey was carried out (SCRE 1988), among adults, to find out their attitudes towards further education. It had been found (Ibid. p13) that the main reasons for participating in education and training were as given below in Table 2.7.1.

Table 2.7.1
Reasons for Participation in Further Education
(SCRE 1988 p13)

	Total N=765 %	Men N=382 %	Women N=383 %
Thought it would help in my job	19	**24**	14
Subject interested me	17	12	**23**
Thought it would be useful to me	15	14	17
Wanted to improve my qualifications	10	12	9
Employer suggested it	9	14	4

(Only the main reasons are quoted here; hence the percentages do not sum to 100.)

For men, participation was seen as vocationally useful, while women were more interested in the subject to be studied.

The main reasons for returning to further education are shown in Table 2.7.2

Table 2.7.2
Reasons for Returning to Further Education
(SCRE 1988 p13)

	Total N=765 %	Men N=382 %	Women N=383 %
The job you were doing then	41	**60**	23
A job you hoped to get	17	19	15
Increasing your qualifications	19	20	17
Personal interests/hobbies	41	22	**61**

(Respondents were able to give more than one answer.)

The most popular response for men was 'The job you were doing then', while for women it was 'Personal interests/hobbies'.

If women are not to be discriminated against, 'personal interests and hobbies' must be taken into account.

This contrasts with the responses when asked why they would return to further education in the future (ibid. p 35). See Table 2.7.3

Table 2.7.3
Reasons for Future Return to Further Education
(SCRE 1988 p35)

	Total N=642	Men N=291	Women N=350
The job you are doing now	19	31	9
A job you hope to get	26	20	31
Increasing your qualifications	22	24	21
Personal interests/hobbies	59	**56**	**61**

(Respondents were able to give more than one answer.)

Both men and women cited 'personal interest and hobbies' as the main reason for returning to further education in the future.

Given that there are gender differences in the reasons for attending college, it is useful to compare the numbers of male and female students enrolling for National Certificate modules - see Table 2.7.4.

Table 2.7.4
Student Enrolments by Sex 1985/86 - 1988/89
(SED 1990, p19)

	Session	1985/86	1986/87	1987/88	1988/89
	Males	46,920	61,786	78,108	97,256
	Females	49,726	62,670	83,030	103,428
Percentages	Male	48.6	49.6	48.5	48.5
	Female	51.5	50.4	51.5	51.5

Neither males nor females can be said to form a significant majority; to ignore gender differences would be to disadvantage about half of all students.

Taking the three earlier tables together, three 'needs' of the student may be identified :-

i) vocational needs
ii) interests/hobbies
iii) the subject is of interest.

If we assume that students returning to take National Certificate modules are in the older age groups, and given that slightly more females than males are involved, the assumption that students continue their education for vocational reasons will result in inappropriate provision for a large proportion of students. In broad terms, older students constitute between nineteen and twenty-four percent of student enrolments and the majority are not interested in purely vocational courses; and women, who make up slightly more than half of student enrolments, may not be as interested in vocational courses as men.

2.8 Employers

A number of students attend college on secondment from employers. As the employer is paying for the student's studies, and has specific reasons for sending him to college, the employer expects to have some say in the modules undertaken by the students. (For example, see Appendix 3.3, Case 2, paragraph 10). Where students attend college as part of a government training scheme, the college discusses the course content with the training officer of the company responsible for the student. (For example, see Appendix 3.3, Case 2, paragraph 4)

2.9 Equivalences

Certain groups of modules are recognised by professional bodies as meeting their requirements for entry to the profession. These groups are refered to as 'equivalences' because they are equivalent to the course that would have been offered prior to the start of the National Certificate.
If a student wants to enter one of these professions he has to take the required group of modules. Some of these equivalences offer very little choice, while others offer slightly more. As an example, shown below in Table 2.9.1, in a letter to schools and colleges in 1984 the Scottish Nursery Nurses' Examination Board listed the modules shown as being "deemed by the Board to be equivalent to the Board's existing Certificate course".

Table 2.9.1
A Modular Equivalent of a Qualification

01001 Communication 1
01002 Communication 2
01003 Communication 3
01179 Accident Prevention and First Aid (0.5)
01310 Music and the Young Child
06002 Basic Nutrition, Menu Planning and Costing
06050 Human Development: Conception - 8 Years
06052 Child Health
06053 Infection and Infectious Diseases
06054 Childhood Ailments
06055 Care of the Sick Child
06056 Stress in Young Children (0.5)
06060 Nutritional Needs of Young Children (0.5)
06061 Infant Feeding and Weaning (0.5)
06065 Language Development
06067 Storytelling, Poetry and Drama (0.5)
06068 Play and the Development of the Young Child (2)
06069 The Young Child and Discovery of the Environment
06070 Family, Community and the Young Child
06071 Children's Clothing and Nursery Equipment
06072 Basic Practical Fabric Work
06075 Nursery Organisation
06078 Child with Special Needs (0.5)
06081 Role of Movement in the Development of the Child 0-8 years
06087 An Introduction to Crafts (0.5)
06088 Use of Crafts in the Development of the Young Child
07500 Food and Food Preparation
or
06063 Nutrition and Dietary Needs (0.5)

Note: the length of each module is 1, unless otherwise stated (in brackets).

This contains the equivalent of twenty-four full-length modules, which does not allow for any others to be added. The only choice is between the last two modules; if 06063 is chosen, the student has a half-module space in his/her timetable.

(Module 06081 appears to be taught on placement, not in college. It is not included in the 'twenty-four' modules mentioned immediately above.)

In 1986 SCOTVEC published a list of approximately one hundred 'modular programmes', each of which was equivalent to a former SCOTEC Certificate (SCOTVEC 1986). The following is a summary of the conditions relating to one programme.

a) Compulsory core - 16 half-modules, 30 full modules = 38
b) plus 1 from a list of two = 1
c) plus two from a list of five = 2
d) plus five full modules or equivalent from a list of 37 = 5
"Including not than TWO of the modules marked * and not counting any module which already has been counted in (a) above."
(Five of the modules were so marked.)
Equivalent to the former SCOTEC Diploma in Agriculture.
(Ibid. pp29-31)

From all the modules listed, only two were not specifically concerned with agriculture; these were "Learning and Study Skills 1", and "Accident Prevention and First Aid". The total length of this course would be equivalent to forty-six full modules; over a two year course (forty-eight modules), this leaves space equivalent to two modules for the student's choice.

2.10 Preferred entry level

Many modules require that the student has already achieved certain qualifications before he starts the module.

Example 1

Office Organisation and Information Processing Ref 02105

Preferred entry level :-

a) 02100 Introduction to the Office and General Services
and
b) knowledge of business documents, e.g. 02101 Business Documents and Methods of Payment, or 02250 Financial Recordkeeping 1

(SCOTVEC 1985, Appendix 1, p39)

Example 2

Health and Safety in the Work Environment   Ref 02352

Preferred entry level :-

Standard Grade English at 3 or above.

(ibid. p42)

Example 3

In some cases the 'preferred entry level' includes modules that may be taken concurrently with the module being considered, for example,

Amplifier Operating Characteristics, Combinational Logic and Counters   Ref 04305

Preferred entry level :-

04302 Electronic Components and Combinational Logic

and after or concurrently with

04304 Amplifiers and Sequential Logic

(ibid. p44)

Most preferred entry levels are expressed in terms of other modules, but some include SCE Ordinary, Standard or Higher Grade subjects.
If a student has chosen to study a module which has entry requirements, he must also take those modules, or, where the entry requirement is an SCE subject, the modular equivalent of the subject. It is in taking these sequences of modules that the student achieves articulation within the course.

2.11 Summary of Chapter 2

a) The Course Grid is a diagram used to represent the arrangement of modules in the student's programme. Each cell represents a half-module, i.e. twenty hours. As this is the smallest unit used for module lengths, all modules can be shown on the grid. The Grid will be used extensively later in this work.

b) the factors affecting the choice of modules for a particular student are;

1) The time available to the student, usually determined by his mode of

study (e.g. full-time, part-time).

2) The need to give programmes breadth, to avoid over-specialisation at too early a stage in the programme.
3) Articulation; that within a course there should be a progression from general to more specialised studies, and that the course should allow for progression to more advanced study.
4) Previous qualifications, i.e. the qualifications the student has before starting the course.
5) Needs and interests. The student's reason(s) for studying modules will vary. Factors which appear to influence these reasons are age, recent educational background, employment aspirations and gender.
6) The need to satisfy employers' requirements (where applicable).
7) The need to provide courses equivalent to the entrance requirements of various professional bodies.
8) The preferred entry level, applicable to some modules. Where a module has such an entry requirement, this must be included in the programme unless the student has already succeeded in it, or in an equivalent.

A fixed programme of modules is unlikely to satisfy the demands of a large number of students.

2.12 Notes

1) There are some quadruple-length modules, which either cannot be completed in one session, or have to be allocated more than the normal time in each block.
2) The Action Plan had recommended a review of post-16 provision in secondary schools; one of the recommendations had been that National Certificate modules should be made available in schools.
3) This had been another objective of the Action Plan. See Chapter 1, 1.2.5.

Chapter 3
Implementing the Action Plan

3.1 Introduction

The structure of National Certificate courses offered flexibility and the possibility of giving students far more choice in the selection of study programmes that had been possible previously. However, the development of strategies to take advantage of these opportunities was hampered by various difficulties surrounding the implementation of the Action Plan. These other difficulties were;

a) The context in which skills in negotiation had to be developed
b) The attitude of staff
c) Changes in the SCOTVEC catalogue
d) Staff development
e) Administrative complication

These will now be considered in more detail.

3.2 The Context in which skills in negotiation were developed.

(This includes the development of choice, negotiation and flexibility.)
The development of choice and negotiation did not take place in isolation. The same staff who were responsible for these developments were involved in a number of other changes at the same time. These were

3.2.1 The writing of the module descriptors.

A Module Descriptor is a description of a module, written in a standard format which had been specified by SCOTVEC. An example is given in Appendix 2.1. In addition, Module Guides had to be written. These gave additional advice about teaching strategies, examples of situations and resources that could be usefully exploited by the teacher.

3.2.2 Turning module descriptors into teaching programmes.

Module Descriptors and Guides did not constitute teaching material; they were intended for lecturers. Notes had to be written for students and lessons developed to teach the specific learning outcomes detailed in the module descriptors.

In particular, the new assessment procedures, with their emphasis on continuous assessment, were new to many college lecturers, and appropriate assessment instruments had to be developed. Module Descriptors were highly prescriptive in respect of the material used for assessments.

These procedures had to be applied to each module to be taught.

3.2.3 The development of a guidance system

> *"Although guidance structures had been in place in the school sector for some time prior to Action Plan, no systematic or formal provision for guidance had existed in further education."* (SCRE 1987, p11)

Some further education colleges had considerable development to undertake, as is apparent from the following extracts;

Speaking of two case study colleges

> *"Although the facilities for careers help are explained in both colleges as part of the induction programme there was little evidence that students had taken notice of this information. In college 1 only eight full-time and six employer-based students out of the thirty-four interviewed knew of the existence of the Careers Officer. "* (Ibid. p44)
>
> *"In college 2 one third of the students interviewed had received some overall programme guidance and almost all the students were able to indicate how many modules they had completed. "* (Ibid. p48)

Two points can be deduced from this statement;

a) two thirds of the students had not received any overall programme guidance;

b) some students did not know how many modules they had completed.

This second point is clarified later;

> *"The implication here is that students are not specifically told at the end of each module whether they have 'passed' or 'failed'. "* (Ibid. p48)

This is probably linked to the difficulties of keeping and collating student records - this will be discussed later in this chapter. Anecdotal evidence also suggests that the SCOTVEC procedures for verifying success in modules, and issuing certificates, can be slow.
Because of the vocational influence on the selection of modules, the development of a guidance system had to precede any realistic attempt at introducing choice into the system. The intelligent choice of modules required careers advice.
Clearly, before negotiation of modular programmes could be considered, an efficient guidance system was necessary.

Faced with a number of important tasks people tend to

a) complete the simple tasks first and defer the difficult ones until later, when they can be given proper consideration;

or

b) defer the simple tasks until later (because they will not take long) and tackle the difficult tasks first.

In the situation facing the lecturers following the adoption of the Action Plan, the former would probably be the case. Negotiation could not take place until there was a choice of modules (without a choice to be made there is nothing to negotiate). This implies that the teaching material had to be developed first, so that the modules could be offered, and the guidance structure had to be established in order to co-ordinate the careers advice to be given. Careers advice would initially have to depend on the experience gained under the old system, and it was the modular equivalents of the old courses that were developed first.

3.3 Staff Attitudes

College staff saw both advantages and disadvantages in the principle of choice, with slight differences depending on whether they were discussing the vocational or non-vocational areas. (SCRE 1985)

3.3.1 The Vocational Area : Advantages

In the vocational area, the possible benefits were seen as;

1) It would increase the motivation of the student;
2) It would help to develop maturity and responsibility;
3) Students could opt out of subjects for which they had no aptitude;
4) It would give a personal emphasis to the work they undertook;
5) The students would be more satisfied, resulting in better staff-student relationships;
6) The students could broaden their range of vocational skills.

3.3.2 The Vocational Area : Disadvantages

However, staff saw difficulties in feasibility and appropriateness;

1) Choice must be limited by external factors e.g. the influence of employers and training boards, the need for equivalences, statutory obligations;
2) There would be organizational difficulties, e.g. timetabling, viable numbers, resources required, mode of attendance (full- or part-time, or block release etc.).
3) There is a conflict between student choice and the perceived professional role of the educator (staff generally believe in vocational coherence - progression to higher levels of study). Students should not be allowed to make choices that would reduce the possibility of this progression;
4) Staff would require time and resources to carry out guidance tasks associated with student choice.
5) Students lack maturity and experience, and are ignorant of what they know or want.

3.3.3 The Non-vocational Area : Advantages

In non-vocational choice, the advantages were seen as;

1) The student could be exposed to wider learning experiences;
2) Experience in other fields of study would help if the student had to retrain;

3) There could be more variety in work programmes;
4) Leisure-related, or personal and social development modules could be integrated into the student's programme;
5) Students who were unsure about which course or vocational area to follow could sample modules before making firm decisions;
6) Students would have contact with staff outside the student's main vocational area.

3.3.4 The Non-vocational Area : Disadvantages

Some of the same difficulties as those for the vocational area were raised Three were stressed;

1) The level of resources to support the system;
2) The extent to which students might see options as relevant;
3) The guidance arrangements required.

3.3.5 Contradictions

The discussion resulting from the issue of choice in the National Certificate raised a number of apparent contradictions.

a) If students are to take responsibility for their work, in the sense of choosing the modules, and if this is to result in increased motivation, the choices must be seen as important. If choice is restricted to a relatively small number of modules, and it does not matter what choices are made, then the students will not take the choices seriously and motivation will decrease in respect of these 'peripheral' modules.

But can students be allowed to make important choices - when a bad choice could have serious repercussions for the students career or future intentions?

b) Should choice be compulsory? Should students be compelled to choose modules from areas they would not have considered if choice were not imposed?

If choice is not compulsory and large numbers opt out the numbers opting for some modules would be too small to constitute viable classes (staffing is not unlimited). This may restrict choice for those who do want it.

In addition, some would choose not to broaden their education, concentrating on a narrow field - one of the objections to the pre-National Certificate courses.
The conclusion is that choice should be compulsory. Compulsory choice implies a contradiction - if a person is given freedom, he should be free to refuse to exercise that freedom.
If the intention of choice in the National Certificate is to increase motivation and student responsibility then students should be given the freedom to choose to broaden their experiences or not, as they decide.
Compulsory choice does not rest easily with the ideals of the Action Plan, because the conditions under which choice may be exercised are determined by college policy, and are not negotiable.

Some of the problems foreseen by staff stem from the idea of 'college-wide electives', where most of the student's programme is drawn from a subject area with a vocational bias, while some modules have to be chosen from other departments (e.g. sports and other leisure activities, personal development etc.).
If 'vocation' is the student's main criterion, how will forced choice into non-vocational areas affect motivation, performance and attendance?

> *"Is the ultimate aim to ensure that students have a broad programme of study, or is it to ensure that students are interested and enthusiastic about a genuinely broadening experience?*
> *If the latter then motivation and co-operation are of prime importance."*
>
> (SCRE 1985)

Choice per se is only valuable if the choice matters - but college-wide choice, in the terms described here, tends to be seen as peripheral.
If choice is to be used to affect motivation and responsibility it must be from within the area that the student sees as important - which for many (but not all) students is the vocational area. This implies that the students must be given enough information to make intelligent choices.

Staff were clearly ambivalent towards the idea of student choice, the main objections being seen as the danger of students choosing inappropriate modules, and the difficulty of resourcing large numbers of (possibly) small classes. They were not convinced that student choice was either desirable or feasible.

From the discussion above, it can be seen that staff had identified many problems connected with the concepts of choice and negotiation and had posed a number of questions, many of which had not been answered at the time that choice and negotiation should have been introduced. These and other questions will be discussed in Chapter 5.

3.4 Overall Growth of the SCOTVEC Catalogue

The table and graph, below, show the total number of modules in each catalogue between 1985 and 1992 inclusive (i.e. session 1985-86 to 1992-93).

Table 3.4.1

The Number of Modules in the Catalogue

Session	(a)	(b) Additions	(c) % change
1985-86	1268		
		485	38.3
1986-87	1753		
		399	22.8
1987-88	2152		
		287	13.3
1988-89	2439		
		99	4.1
1989-90	2538		
		128	5.0
1990-91	2666		
		47	1.8
1991-92	2713		
		256	9.4
1992-93	2969		

a) Number of modules in Catalogue

b) Net increase in number of modules

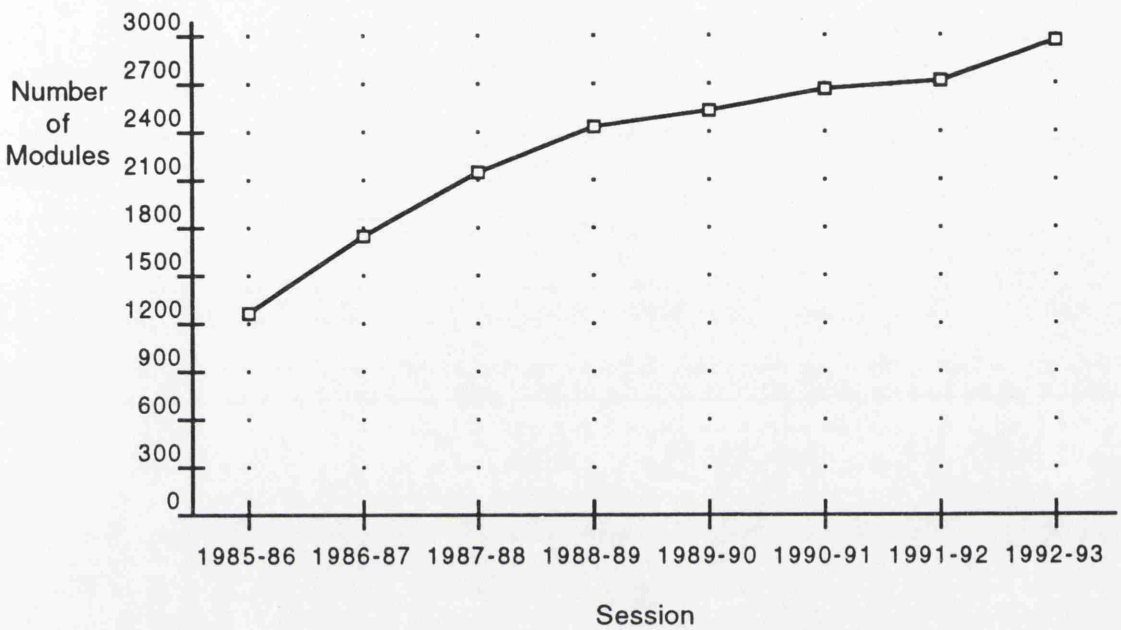
c) Increase as a percentage of the older catalogue

The figure for 1985-86 includes only those modules available as at May

1985, i.e. it excludes those that were to be available by September. This is to allow proper comparisons to be made with other catalogues, as they did not include the latter figure either.

Figure 3.4.1

Growth of the SCOTVEC Catalogue



Before the final catalogue for 1985/86 was published a number of interim catalogues were distributed. Each listed more modules than its predecessor, as we might have expected for a catalogue describing the initial 'setting up' phase of the operation. The table shows the catalogue initially growing as we might have expected; after a short settling period, 1986/87, omissions and deficiencies in module provision had been identified and the addition of the necessary modules accounts for the large increase between 1987 and 1989. The catalogue then settled down, with routine additions being made between 1989 and 1992. Indeed, by 1992, it was probably felt that the catalogue was virtually complete, with only special cases requiring the addition of more modules. However, for the 1992/93 a slightly larger number of modules was added. This may reflect a change in the organisations responsible for the writing and development of modules. These changes will be described later.

The figures in table 3.4.1 show the overall increase in the size of the catalogue, but the true scale of the alterations is not correctly shown.
From one catalogue to the next, modules can be added, revised or deleted. Some modules are deleted while others are added. If 10 modules are deleted and 15 added, the overall figures will indicate a change of 5. To the user, a more relevant figure is the number of alterations, which is 25 in this example.
An attempt was made to identify in more detail the changes to the modules. In some years this can be done by comparing the module reference numbers from one catalogue with those from the next.

a) If the reference number does not appear in the subsequent catalogue, the module has been deleted.
b) If a reference number appears in the later version but not in the previous catalogue, the module has been added.
c) If the reference number appears, but the first digit has been changed, the module has been revised. (The numbering system will be described shortly.)
d) If the reference number appears, unchanged, in both catalogues then the module is unchanged.

Placing the modules in one of the groups listed above requires that every module in one catalogue is compared with every module in the subsequent catalogue. To identify modules in group (b) the comparisons must then be repeated, comparing every module in the later catalogue with every one from the earlier catalogue. (It is possible to reduce the number of comparisons if certain assumptions are made - it is still a very long and tedious process.) A computer program was written to perform these comparisons, although all the module reference numbers still had to be entered, in itself a tedious process.
The comparisons described above are only meaningful if the reference numbers can be interpreted in a specific way - in some years this was not possible. (See below under "The Numbering System".)
The results given below, while incomplete, give an indication of the number

of alterations from one catalogue to the next.

Table 3.4.2

Comparing Catalogues 1985-1989

	1985-87	1987-88	1988-89
Revised	1558 *	164	201
Added	593	337	179
Deleted	110	66	80
Alterations	703	567	460
Old Total	1668	2152	2439
Alterations %	42	26	19

Alterations % = Alterations as a percentage of Old Total

* In 1985 all modules had a revision number of '0'. By 1987 all modules had been given revision numbers of '6' or '7'. As the majority of these were probably simple cases of renumbering, they have not been included as 'Alterations'.

The following information is also relevant;

a) In the 1985-86 and '86-87 catalogues, at the end of some categories there appeared a list of modules that were being prepared but not as yet ready for use.

	85/86	86/87
Number of modules in catalogue =	1268	1753
Number of modules in preparation =	366	323

If we assume that these modules would be ready for use in the following session, the next catalogue would increase by 29% in 85/86 and 18% in 86/87. It should be noted however, that as far as a member of staff is concerned, it is the **number** of alterations, not the proportion, that is of interest.

b) In the 1990-91 catalogue a new system of numbering was used, changing from a five digit to a seven digit reference. Only new or revised modules were given the seven digit reference; this allowed these modules to be identified easily.

Table 3.4.3

Comparing Catalogues 1990-92

	90-91	91-92	92-93
Five digit =	2433	2137	2205
Seven digit =	233	576	764
Total =	2666	2713	2969
New 7 digit =		343	188

'Five digit' means the number of modules with the old, five digit reference number. 'Seven digit' refers to the new seven digit system. 'New 7 digit' is the number of seven digit references that were not in the previous catalogue.

The table merely confirms the impression already formed from the previous figures; the catalogue is not static, but subject to constant revision and expansion.

It would be difficult for any lecturer to be confident of his ability to advise on the selection of modules in anything other than his own field, because it is difficult to become familiar with information that keeps changing.

3.5 The SCOTVEC Numbering System

3.5.1 The original system

The original numbering system, used from 1985 until April 1990, consisted of a five digit number, e.g. 12345.

The first digit, (1), was used to indicate the year in which the module was being offered.

The second digit, (2), indicated the category number. The catalogue was divided originally into nine categories, numbered 1 to 9.

The remaining three digits were the reference number of the module within the category.

The five digits together gave a unique reference number for each module.

Modules were designated 'General' or 'Specialist'.

Some remarks on this numbering system.

3.5.1.1 The first digit indicated the year in which the module was offered.

This would change every year, and is effectively the year in which the student took the module. This may have had some significance to the SCOTVEC administration, but what would be its value to an end-user, for example, a potential employer of a student? An employer may have to choose between two candidates for a post. They might both have taken the same modules, but in different years, in which case the module numbers will be different - but the difference would indicate nothing that the date of the certificate did not already show.

3.5.1.2 SCOTVEC insist that presenting centres are careful to use only the reference numbers valid for the year of presentation. It is therefore likely that module descriptors would have to be reprinted every year for no other reason than to show the change in the first digit. Any data-base containing the reference numbers would have to be updated every year to show the change in these first digits. This seems to be a lot of work for no real reason.

3.5.1.3 The first digit would recur every ten years. If, as was anticipated by the Action Plan, people were expected to return to college to learn new skills from time to time, they might have had modules spread over a number of years. Two modules with the same 'year' digit might have been taken ten years apart. While this may be a trivial point, it does seem to offer an unnecessary possible source of confusion.

3.5.1.4 In most numbering systems the first digit is assumed to be the most significant, and, when a data-base is being searched, can be used to reduce the search space. The subsequent digits, taken in order, narrow the search area until only one item is located. With the SCOTVEC system the first digit gives no indication at all of the location of the module within the catalogue - it is the second digit, the category number, that first reduces the search space. Any computer system working with these numbers would have to be programmed to ignore the first digit and use the remaining four. While, again, this is a trivial amount of work, it does seem unnecessary.

3.5.2 1985 (No edition number)

From memory it is thought that there was a 'very first' catalogue, a stapled set of duplicated sheets. No surviving copies of this have been found.

The earliest catalogue located was dated '1985', with no month or edition number. A note written on the cover at the time states that this catalogue was replaced by 31st June 1985.

A second version of the catalogue was dated 'May 1985', with no edition number. It is not now clear whether this is the replacement referred to above, or whether there was yet another version published in 1985. Given that the next edition was called the third edition, it seems likely that the first edition was the '1985' with no month, and the second edition was the 'May 1985' version referred to above. In both these editions all modules had '0' for the first digit.

Remarks

3.5.2.1 The first year digit used was '0', indicating 1985. If the year digit is important, (and that has been questioned above), all subsequent operations would have to take into account this 'translation of the origin'. For example, it is not immediately obvious that '8' means 1993.

3.5.3 1986-87 3rd Edition

The first digit for all modules was '6', indicating that they were available for use in session 1986-86.

Remarks

3.5.3.1 By using the last digit of the session, point 3.5.2.1, above, has been corrected.

3.5.4 1987-88 4th Edition.

The meaning of the first digit of the reference number was changed. It now meant the year in which the module was first offered, the number changing only if the module was revised. Unrevised modules carried the prefix'6', revised modules '7'.

Remarks

3.5.4.1 Point 3.5.1.2, above, was corrected by this change. Only revised modules would have to have their reference numbers changed.

However, a potential source of confusion was introduced. What is the difference between modules 61234 and 71234? The '1234' indicates that they are almost the same module, but 71234 is the revised version of 61234. What was the extent of the revision? Is the difference significant - it was significant enough for SCOTVEC to renumber the module, but is it significant to a potential employer?

SCE Ordinary and Higher grades are revised on occasions. For a short time the older version of the syllabus is usually referred to as the 'traditional' syllabus and the other is called the 'revised' or 'alternative' syllabus. Once the older version has been phased out the name of the syllabus reverts to the original name. So, for example, Higher grade Physics has been in use for many years. Despite a number of alterations in content and format the basic principles taught - and mastered by the student - remain unchanged. With short duration modules, teaching highly specific skills, a revision may change the nature of the module considerably, effectively making it a different 'subject'. Module teachers have said that a revision can imply anything from a minor rewording of the module descriptor to a major alteration to the content and methods of assessment. How much revision can be made before the module is designated a new module, with its own reference number (the older version probably being dropped from the catalogue)? No clear answer to this question has been found.

As a further complication, it seems to be standard practice for presenting centres to destroy obsolete module descriptors. This is not surprising - a file containing two thousand descriptors, each of four or five pages, is a fairly bulky item. There simply is not enough space to justify keeping old versions. But if the old versions are destroyed, how can anyone find out what skills were acquired in a module taken a few years earlier? A statement of the learning outcomes is sent to the student with the certificate stating the modules satisfactorily completed. The employer has to rely on the student still having both the certificate and the statement of learning outcomes.

CAST do not keep old descriptors; they do not even keep old copies of the catalogue. The older catalogues used in this study were acquired from a SCOTVEC field officer, who admitted that the only reason he still had them was that he had not tidied up for some time. No-one seems to see the need to archive this material. Whether this is significant or not remains to be seen.

3.5.5 1988-89 5th Edition.

The category '0' was introduced. This was called 'Interdisciplinary Studies', as was category '1'. It contained all the Physical Education and Recreation modules originally in category 1, but with a large number of additions.
The distinction between General and Specialist modules was discontinued, as "The distinction between the categories became blurred and was not considered to be helpful" (SCOTVEC 1988 p4).

Remarks

3.5.5.1 In the five digit system the second digit gives the category number. This implies that all knowledge (or at least all knowledge to be taught at non-advanced further education level) can be placed into ten categories. The original catalogue contained only nine categories. In 1988 a tenth category was deemed necessary. What will happen if another category is required at some future date?

3.5.6 1989-90 6th Edition

No change from the principles above.

3.5.7 1990-91 7th Edition.

In May 1990 a completely new system for numbering new and revised modules was announced.
A seven digit number was to be used, as follows;

Sector Board			Year/ Series
7	0 1	0 0 2	0
Cognate Group		Module Identifier	

The 'Sector Board' refers to a group of people with overall responsibility for each part of the catalogue. In practice, this divides the catalogue into categories similar to those used previously.
Table 3.5.7.1, below, lists the Sector Boards.

Table 3.5.7.1

List of Sector Boards

1 Land and Sea Based Industries
2 Engineering an Manufacture
3 Industrial Processing
4 Built Environment
5 Service Industries
6 Business Administration and Management
7 Community Services
8 Communication and Information Systems

The Sector Board number along with the next two digits form the 'cognate group', a group of modules related by subject matter. The next three digits form the Module Identifier, a number identifying the module within the cognate group.

The number formed by combining the cognate group with the module identifier gives a unique reference number for each module.

'Year/Series' indicates the year of introduction of the module (1990 = 0).

The old five digit system would be used 'in tandem' with the new system until (presumably) all modules were either new or revised. It was recommended that, for data-bases etc., five digit numbers could be converted to seven digit numbers by prefixing '00' to the five digit number.

Remarks

3.5.7.1 There would be no straight-forward relationship between a 'proper' seven digit reference and that formed by expanding the old five digit number, as the first non-zero digit in the latter would be the revision number, not the category (the nearest equivalent to the sector board). A data-base designed to work with both types of reference number would have to have two separate routines for interpreting the numbers.

Relegating the revision number (Year/Series) to the last digit answers the objection, 3.5.1.4, that the first digit should be the most significant.

Using three digits to indicate a cognate group effectively gives a huge number of possible 'categories' (solving problem 3.5.5.1).

When the 1990-91 catalogue was published it was found that, in addition to the change to the seven digit reference number, the modules were also arranged in a different way within the catalogue. The reference number was not used as the basis for ordering the modules. The SUPERCLASS classification system (DoE 1989) was used to divide the catalogue into seventeen categories, each category being further sub-divided.

> *"SUPERCLASS is a subject structure for organising and searching databases and other information on training and education opportunities."* (DoE 1989, p1)

The system will be discussed in more detail in the next chapter. The example from the SCOTVEC 1990-91 catalogue shows the new structure of the catalogue. (SCOTVEC 1990)

Figure 3.5.7.1

Catalogue Arrangement 1990/91

Section A	Business and Management	-	SUPERCLASS main class
A2	Small/New Businesses	-	SUPERCLASS sub-division
6120410	Considerations for Self-Employment	-	SCOTVEC reference no.
6120420	Researching and Business Opportunity		
6110110	Roles and Responsibilities within Group Business Activity		
A3	Management Skills Systems and Techniques		
6110310	Basic Applications of Behavioural Science		
82503	Committee Skills	-	SCOTVEC old style reference no.
6110410	Interviewer Skills		

Note that a cognate group (e.g. 611----), can be spread over more than one SUPERCLASS class.

In SUPERCLASS the sub-divisions of a main class are indicated by a decimal point. For example, 'A2' should be written as 'A.2' . SCOTVEC omit the decimal point. This difference is probably trivial.

The point of this lengthy description of the numbering system is that no-one can become skilful at a task if the data relating to that task keeps changing. Lecturers could only become confident in their ability to negotiate a suitable programme if they became familiar with the modules and the relationships between them. While this might be possible within the restricted set of modules taught by a department, extending the negotiation to modules from a wider range would be considered impossible.

It is paradoxical that the flexibility of the modular system, and the ability to add or revise modules to improve them or to fill gaps in provision, itself militates against the effective use of that flexibility.

3.6 Training and Help in Negotiation.

To aid the development of the National Certificate a number of 'Task Groups' were established. These were;

a) Curriculum Assessment and Certification

b) Teacher Education

c) Guidance (SED 1983b, p4)

There was no group specifically set up to advise on choice and negotiation - these were assumed by the Task Group on Guidance. Given the earlier comments on the state of the Guidance system in FE colleges, it is understandable that the main effort of the Group was concentrated on the consideration of various forms of guidance structure that might be considered suitable, the various people who might be made responsible for guidance, and the duties and responsibilities to be assumed by these people.

Two booklets which appeared to be relevant to choice and negotiation were published - these were "Selecting a Programme of modules" (SED 1985a) and "The Negotiation Process" (SED 1985b)

The first of these contained eight pages. Two pages described the SCOTVEC catalogue. Three pages contained a list of the publications associated with the Action Plan in general. The remaining three pages discussed the construction of a modular programme in very general terms - in many cases stating the obvious (or, at least, it would be obvious to anyone involved in the National Certificate). For example,

> *"The number of modules a student will be able to undertake in a programme will be determined by the length of attendance."*
>
> (SED 1985a, p2, para. 4.1)

The second booklet also contained eight pages. A number of statements were highlighted as being 'keys' to successful negotiation. These were;

> *"A key to successful negotiation is making it clear that there are limits to the matters which can be negotiated."* (SED 1985b, p2)
>
> *"A key to negotiation is in setting out the stages of the process at which negotiation is possible. "* (Ibid. p3)
>
> *"Another key step in successful negotiation is to encourage both parties to produce their own suggestions and then work to reconcile these."* (Ibid. p3)
>
> *"A further key to negotiation is that it is two way. It encourages a degree of freedom but not licence. "* (Ibid. p4)
>
> *"Another key to successful negotiation, therefore, is setting limited targets at the beginning of the process. "* (Ibid. p4)

This appears to be a rather negative approach - negotiation will occur when the college decides, it will be about matters decided by the college, it will be limited in aims, and it will be a compromise.

The clearest examples of negotiation given were;

- does who sits where, or how the seats are arranged, matter?
- what should students be called (surname, Christian name, "you") and what should they call the teacher or lecturer?
- will lecturing be the sole teaching approach, or will there be a variety of methods? (Ibid. p 4)

The writers of the booklet clearly took a wide view of 'what should be negotiated', but the only reference to the negotiation of the programme of modules itself lay in three fairly general sentences;

> *"i. to discuss with members of teaching staff the quality of the progress they have made in each module and what this suggests for the modules which should be tackled next;*
>
> *ii to spend some time looking at the descriptors for the next modules available and then to discucs these with the teaching staff involved;*
>
> *iii to review with guidance staff the most sensible choices of study."* (Ibid. p6)

In neither of these booklets was there any suggestion of a model that could be followed or even developed by the lecturers likely to be involved in negotiating programmes of study.
In 1987 the Curriculum Advice and Support Team, (CAST), published a curricular package for staff development modules. The package is in the form of module descriptors, with explanations as to how the modules can be delivered. Volume 2 is concerned with various aspects of guidance, and contains module 2.4 "Programme Design". (CAST 1987)
About two hundred copies of the series were issued to colleges and schools, for use in in-service training. CAST were not in a position to say whether the material had actually been used or not.
As the specification for a module, this does not actually give teaching material - more specifically, it does not tell the reader how to negotiate modular programmes - but suggests workshops, case studies and other activities through which the principles of programme design could be explored.
When the syllabus is published for a subject, for example, Higher grade Physics, it will consist of statements such as "the candidate will be able to solve problems on projectile motion". The syllabus does not say how these problems are solved - it is assumed that the teacher knows. In the case of programme design, however, where the 'teachers' do not know the answers, the value of such a syllabus is questionable. One can imagine groups of lecturers all over the country using the CAST modules and trying to work out a model for programme design. Some will be more successful than others - some may not arrive at a solution at all. It would have been more useful for a group of lecturers to be seconded to develop a rationale for programme design, incorporating negotiation, and then for CAST to disseminate this to everyone else. Against this it can be argued that every college will have different problems, and any such model would still have to be 'tuned' to the particular conditions in the college.

The CAST document does include definitions of what the authors saw as the four key characteristics of a good programme, although reservations have already been expressed about whether they are more correctly characteristics of the modular system. As the statements by CAST were particularly succinct and informative, they have been reproduced here in full.

Flexibility

Definition : The potential to meet the needs of users by modifying programmes of learning.

Rationale : Enables centres more easily to meet the requirements of a diverse range of students.

Examples : Modification of programmes in which some modules are i inappropriate to particular students' needs, e.g. some modules already achieved.
Construction of novel programmes to meet new needs.
Provision of accelerated learning.
Open and resource-based learning.

(Ibid. p20)

Choice

Definition : Involvement of the student in making decisions about the design of his/her modular programme.

Rationale : Better curricular decisions may result when the student makes a guided choice rather than being allocated on the basis of the views of staff.
The student is thereby taking responsibility for his/her own learning.
The exercise of choice may increase the motivation of students.

Examples : Opportunities to choose from a range of programmes.
College wide electives.
Departmental electives.

(Ibid. p21)

Breadth

Definition : The programme helps the student to develop broad transferable competencies relevant to a range of jobs and to other aspects of life, and in general promotes personal growth.

Rationale : From the point of view both of all-round personal development and of occupational competence, it is desirable that students develop broad competencies including communication skills, interpersonal skills, planning and problem-solving, and self-confidence and initiative.

Examples : Programmes in which there is a conscious attempt to incorporate the development of broad competencies, e.g. through the inclusion of modules with this particular purpose.
Programmes in which students have access to modules outside their specific vocational area.
Programmes which seek to defer specialisation and prepare

the student for a range of opportunities within a general vocational area.

(Ibid. p22)

Integration

Definition : An approach which assists the student to integrate learning from various elements in the programme.

Rationale : The transferability of skills and knowledge is increased if students are able to draw connections between different elements of their programme. Integration may support breadth.

Examples : Combining a number of modules into a single learning and teaching process.

Analysing learning outcomes so as to be able to explain to students their relationship, e.g. in terms of core skills.

(Ibid. p23)

3.7 Administrative difficulties

These difficulties occur in four main areas;

a) The negotiation process itself

b) Record keeping

c) Timetabling the chosen modules, taking into account the staff and resources available

d) Integration

3.7.1 The Negotiation Process

1) For the closest match with the intentions of the Action Plan, the student should be able to choose from a wide range of modules. Because of the annual changes in the catalogue it is difficult for the lecturer to be fully informed about modules outside his own speciality.

2) Negotiation is required in order to identify the student's needs, with reference to his existing qualifications, vocational intentions, and interests/aptitudes. Negotiation is required before the student starts his work at the college, and thereafter, at the end of each block of modules (in the typical arrangement, this is every thirteen weeks). Finding the time to interview each student, individually, can be difficult.

3) Students frequently cannot have a free choice of modules, even if the college were capable of accommodating this. For example :-

If a student has a specific vocational intention his choices will be constrained by the requirements of the occupation - i.e. the student is required to take the modules specified as being equivalent to the 'old-style' course which was originally designed to train entrants to the occupation.

> *"It must be said in reply, however, firstly that there is a large population of students who do not fall into this category and secondly that there may be greater room for flexibility and choice even in 'equivalent programmes'. "*

(Reproduced from BROADCAST 5 winter 1986, in CAST 1987, p110)

If the student is being sponsored by an employer (for example, on day or block release), the employer may insist that certain modules are taken. If the student hopes to go on to more advanced studies, the modules taken have to prepare him for these studies, possibly to gain entrance to the course.

4) In some colleges (see Appendix 3.3, Case Study 1, Note 5) students can choose to study modules which are already being provided for other courses. This 'in-filling' allows a wider choice than would otherwise be available, and helps to ensure that all classes are of a viable size. Negotiation concerning these modules can only take place after the 'fixed' modules have been timetabled.

5) Some students, particularly in the early stages of their college careers, may not be able to articulate their needs clearly. For the negotiation process to work they need guidance. However, for maximum effectiveness timetables must already have been constructed - it is frustrating to identify a need, suggest a module to satisfy this need, and then find that the module cannot run because of timetabling difficulties.

3.7.2 Record Keeping

1) Where students have taken modules at school, there is a considerable delay before the student receives his/her certificate stating the learning outcomes successfully completed. Colleges therefore have to make assumptions regarding these modules. If the assumptions prove to be

wrong, programmes have to be changed at short notice. In some cases students repeat modules because of confusion over what has been completed.

2) At the end of each block of modules the student's existing qualifications have changed (he will have succeeded in some modules). As in (1), above, it appears that it is time-consuming to find out which modules have been successfully completed. (See Appendix 3.2, Case Study 2, Note 5. In that case, the problems were to a large extent solved with the introduction of computer assistance.)

3) When courses consisted of a fixed set of subjects, taught to a fixed set of students, all records could be kept as a group. With the modular system groups are far less coherent - in theory, every student in the college could be taking a different set of modules. Information about the students is spread over a large number of registers and record sheets. Finding this information in order to be able to advise the student effectively is very time-consuming.

For a statement of the problem of record keeping, see Appendix 3.2, Case Study 2.

3.7.3 Timetabling

1) Resources, in terms of accommodation, equipment and staff, are limited. If a large number of students choose a module, they will have to be timetabled as two or more distinct classes, because the room cannot hold the number required, or equipment is not available in sufficient numbers. Expensive equipment cannot be 'duplicated' for additional students - the timetable must ensure that maximum use is made of the existing equipment.

2) If a small number of students choose a module the class may not be viable - using a member of staff to teach three or four students is not efficient when he could be teaching a class of twenty. Education authorities usually specify a minimum class size - if the number of students falls below this, the module must be cancelled, and the students have to re-choose modules. This not only means that the students'

programmes have be re-negotiated, but the students are likely to be frustrated.

3) The supply of rooms, equipment and staff cannot be adjusted quickly, therefore it is essential to forecast the likely demands accurately and in sufficient time that (a) an attempt can be made to satisfy the demand with the existing resources, or (b) additional resources can be obtained.

3.7.4 Integration

Integration comes from the Action Plan demand for rationalisation in colleges (SED 1983a, pp49,51). By integration is meant the merging of two or more modules which have a common component. For example, where a number of modules all include the teaching of basic mathematical skills, integration requires that the common skills are taught as one, possibly separate, module. The remainder of the modules can then make better use of the time available by simply assuming that the skills have already been taught. Integration also allows the most appropriate staff to be used for teaching the 'integrated' module. Integration should make better use of resources, accommodation and staff. By concentrating 'large' numbers of students in a large room, instead of spreading them out as a number of smaller groups, each with its own teacher, it should be possible to release rooms and staff to teach a bigger variety of modules - allowing more choice. The Action Plan specifically discussed these points (SED 1983a, p51), stating that in many cases class groups were formed according to the needs of practical classes, where safety and supervision demand small groups, but were not re-organised for 'theory' classes, where the same constraints did not apply. The result was that there was an unnecessary demand on staff and accommodation. The modular course structure was supposed to overcome this problem - practical modules could be limited to small numbers, while non-practical modules (involving students from a number of different practical modules) would involve larger numbers.
An example of integration appeaied in CAST 1987(p74). A number of modules in Electrical Engineering were integrated by identifying common elements. There are, however, dangers in integration. If a number of

modules assume that certain skills have been taught in a separate module, that module becomes an essential preliminary to studying the dependent modules. If groups of modules are integrated in this way, then the whole programme is determined in advance and we end up back at the old style courses.

> *"Where such schemes have been in existence for some time (in higher education in Britain and high schools in the USA), it has been universally found that structures harden over time, with well-marked pathways forming through options and crypto-cores emerging, such that a programme of modules comes to resemble something very like a 'course'."* (Jonathan 1987, p88)

It is this danger that prompts the statement that integration and flexibility are mutually exclusive (CAST 1987 (BROADCAST 5 Winter 86)). This is because integration is easiest to organise when a defined group of modules is taken by a stable class group, whereas flexibility implies free-standing modules, for students from different classes.

It is felt, however that the two do not need to be mutually exclusive. If the modules are integrated with flexibility in mind, and are then used sensibly, the advantages of integration can be gained without sacrificing too much flexibility.

3.8 Conclusion

Given the problems raised earlier in this chapter it would not be surprising to find that negotiation is not a prominent feature of college programmes, and this appears to be the case.

In 1985, immediately after the introduction of the National Certificate, it was found that choice was "very limited" in the twenty colleges contacted. Three colleges were studied in more detail - of the 19 departments in these colleges, less than 1/3 offered any kind of vocational choice. (SCRE 1985)

A later study showed that the situation had not changed very much;

> *"On the whole, however, we found little regional support regarding the introduction of choice into the colleges ."* (SCRE 1987, p68)
>
> and

"Some principals, however, maintained that choice was inappropriate for students: that they chose to come to college in the first place and preferred to leave the design of their programmes in the hands of the staff." (Ibid.)

"What we have found throughout Scotland is that choice for students is very limited and marginal to the mainstream provision of further education." (Ibid.)

The main features operating against choice were;

a) Staff were not convinced that it was desirable or possible.

b) Staff tended to see college courses as vocationally oriented, and that choice should extend only as far as choosing the college and the course, not the individual components within it.

c) The Action Plan had to be implemented quickly. A number of tasks had to be completed. Some, like the provision of a satisfactory guidance system, were seen as more important than others - some tasks were essential to the implementation of the modular system, while others, like negotiation, could be attended to after the system was operating.

d) The frequent changes to the SCOTVEC catalogue reduced the confidence of the lecturers in their ability to advise effectively on a wide range of modules, i.e. modules from outwith their own departments.

e) Allowing choice from a large variety of modules produced a situation where it was likely that there would be smaller classes. No model was provided to show how choice could be increased within existing resources.

f) Training for negotiation was minimal or non-existent.

g) Difficulties were foreseen in timetabling, and in keeping up-to-date records on student progress, making continuing guidance difficult.

Appendix 3 contains examples of case studies and interviews in which these difficulties can be seen.

It seems strange that colleges containing departments that were teaching students to use computers for the maintenance and interrogation of large data-bases were not themselves in the forefront of computerised record-keeping.

However, there are some examples showing the way forward, e.g. Appendix 3.2, Case Study 2. In this, the administrative problems, including timetabling, were eased by the use of computer programs.

This case highlights what might be seen as three desirable features;

a) the department concerned was large;
b) preliminary work, in preparation for the new session, started in March;
c) computers were used to ease the administrative burden.

3.9 Summary of Chapter 3

a) Staff concerned with the implementation of the Action Plan were required to perform a number of tasks, and develop new skills. Relatively little time was available for this. The development of skills related to choice and negotiation was low on the list of priorities.

b) The relevant training and advice were inadequate for the purposes of developing these skills, of clarifying the issues involved, and of generating confidence in the ability to cope with choice and negotiation.

c) As a result, staff were ambivalent towards these concepts, being able to see both advantages and disadvantages. They were not convinced that choice and negotiation were desirable, and, possibly with some justification, that they were possible.

d) Given the speed with which the National Certificate was developed it is understandable that changes would be required in the catalogue of modules offered. However, the number of changes involved, and the frequency with which they occurred, militated against any widespread involvement in negotiation, and this reduced the amount of choice offered.

e) The administrative difficulties associated with the large number of modules available, and with record-keeping and timetabling, were not given adequate attention.

f) As a result of all these factors, choice and negotiation did not feature prominently in the experience of most students.

g) Where progress was made it was generally associated with the use of computers to deal with the difficulties mentioned in (e), above.

3.10 Conclusion

If the provision of more choice, guided through negotiation, is to affect a larger number of students in further education, it will be necessary to address the administrative problems associated with record-keeping and timetabling. Timetabling, being concerned with staffing, accommodation and the availability of specialised equipment, is inextricably linked with the provision of resources. These are not within the scope of this work and will not be discussed, except for the following comments.

a) Record-keeping involves the storage of information in such that a way that it can be up-dated, retrieved and inspected, preferably easily and quickly. Computerised data-bases are now widely used for these purposes, and it is desirable (and expected) that they will become more common in further education.

b) The processes involved in timetabling can be described in mathematical terms. While anecdotal evidence might suggest that good timetabling is an art rather than a science, it can be greatly facilitated by the use of computers. Programs are available to assist timetablers, if not to replace them completely. Again it is expected that computers will increasingly be used for this purpose.

Chapter 4
Designing a Modular Course Manually

4.1 Introduction

The procedure for constructing any modular programme may be considered in three parts;

a) select modules relevant to the student's needs and interests;
b) check the various combinations of modules to ensure that they can be completed in the time available;
c) select the best programme from those identified.

Parts (a) and (b) are essential, (a) because the course must motivate the student to succeed, and (b) because the course must be capable of completion. Part (c) is not essential; any course satisfying (a) and (b) will have some value for the student. If, however, a number of combinations of modules is found after parts (a) and (b), it would be desirable to be able to identify the 'best'.

The aims of this chapter are

a) to list the tasks involved in selecting modules;
b) to give an introduction to the process of checking that the modules can be formed into a viable course;
c) to discuss a simple approach to course evaluation;
d) to show that these tasks are too time-consuming to be performed manually;
e) to show that there is every reason to believe that a computer could perform these tasks.

4.2 Selecting modules

One of the criteria used by students in choosing modules is 'relevance to a job'. A student may therefore ask the equivalent of "Which modules are relevant to this job?" In order to ask this question he has to be able to specify the job. Just as a library classification system can be used to guide the user to books on a particular topic, it was felt that a job classification system would be useful in helping the student identify the job in which he was interested, and also allow him to specify the job unambiguously. What was sought was a classification system that would allow the student to select from a list of recognised names, working from general areas of related work to more specific job names.

A number of classification systems for jobs were examined. These had all been designed for use in careers guidance and, to a greater or lesser extent, the principles behind their design had been determined by the purpose to which they were going to be put. As careers guidance was not one of the objectives of the system proposed here, none was immediately recognisable as the most appropriate for this purpose.
It will be mentioned later that it had been found that CAST were using the SUPERCLASS classification system in their computerised guidance system. Details were obtained and it was decided that this was clearly the most appropriate system for the purpose described here. This proved to a fortunate decision, although perhaps not coincidental, as the system was later incorporated into the SCOTVEC catalogue (see 3.5.7.1).

Another criterion used by students involves the identification of an interest, possibly a hobby. A brief search was made for a relevant way of categorising hobbies, but none suitable was found. Fortunately, it is fair to say that 'one man's hobby is another man's job'. Obvious examples are Car Maintenance, Interior Decoration, Cooking, Carpentry, Child Care. Virtually any hobby can appear as a job, and therefore a reasonably large list of jobs will automatically contain a large number of possible hobbies. There will almost certainly be exceptions; for example, butterfly collecting may not be a job in its own right. However, information relevant to this hobby may well be found in an appropriate zoology module. 'Train spotting' would probably not appear in a list of jobs. However, in such cases we must ask, "Is a student likely to want to study a module on this topic?". If the answer to this is "Yes", then we can follow with the question, "Is college the best place to study such a module?". Mechanisms exist, within the SCOTVEC structure, to ask and answer both of these questions. If sufficient demand exists for a module, on any topic, it is possible for one to be written.

A useful characteristic of the SUPERCLASS system, or at least, of the way it is operated, is that the system can expand to include new job names. If necessary therefore, a hobby for which modules are available but for which no appropriate job is specified could be added to the data-base. SUPERCLASS includes one category, Category C, which specifically includes hobbies, although, as indicated above, any area of the system can be searched for names of hobbies.

Finally, a student may want to select modules on sports or games. SUPERCLASS Category G contains names and codes for a large variety of sports, games, and recreational activities.

Consideration of these factors leads to the conclusion that there are three types of course.

a) Qualification-centred. In this type of course an employer has specified the requirements of a course, by listing the modules that must be studied (see 2.9, Equivalences). The course is designed to satisfy the needs of the employer - the student's needs are met by choosing the course that leads to the desired qualification.
Courses designed by professional bodies (groups of employers) are also of this type.
b) Higher education-centred. Having completed this type of course the student will have met the entrance requirements for a course (or courses) in Higher Education. It is the needs of this more advanced study that determine the content of the student's course. The student's needs are met by choosing the course that gives entry to the desired field of Higher Education.
c) Student-centred. In this type of course it is the needs and, particularly, the interests of the student that are given greatest priority. Given the descriptions above, 'needs', in the context of a student-centred course,

do not include the need for a specified qualification or for entry to Higher Education. As explained above, the student can specify his needs in terms of jobs, hobbies, sport or recreational activities.

It may be felt that the distinction between (a) and (b) is somewhat contrived, because both ultimately result in a qualification (or that is the intention). A degree course may well be designed to satisfy the requirements of a professional body. It will be shown, later, that even if they are considered to be different types of course they can be constructed using the same method. On the other hand, some of the tasks involved in constructing the student-centred course(1) are distinctly different from those involved in constructing the other types of course.

The method for designing any course must take account of the other factors affecting course design, i.e. breadth, articulation(2) and module entry requirements.

4.2.1 The Target Group

To reduce the complexity of the description a specific target group was identified. Referring to the list of factors given in 2.11 we can identify a specific group of students by making decisions concerning their present characteristics.

1) The target group will consist of full-time students. If all other factors are the same, the only difference between a full-time course and a part-time course will be the number of modules in the course. A method which allows a course to be designed for a full-time student will therefore be applicable to the part-time student as well, by reducing the size of the course grid.
2) The student is likely to have some previous qualifications. These may be a mix of SCE subjects (e.g. Standard grades) and modules, as many schools now offer modules as part of their curriculum.
3) The students will be attending college for the first time. This merely limits

the number of modules already completed by the student to those he took at school. A method which can accommodate these students can be applied to those in a second or subsequent session at college - modules completed in one session become 'previous qualifications' when designing the later parts of the course.

The target group has been defined so that successfully designing a course for this group (or a student within the group) will involve all the tasks required in designing a course for any group. The process of designing courses for other groups (for example, students entering their second year at college) will be a variation on one of the methods described below.

4.3 The student-centred design process

The student-centred course will be discussed first because :-

a) this is the type of course that makes the greatest use of the flexibility inherent in the modular system;
b) it is the most difficult type of course to design;
c) no descriptions of the process have been found elsewhere.

The target group described above can be further refined to identify those students who would benefit most from the student-centred course. The students in the target group will not have specific qualifications in mind, but they will be able to identify an area or areas of interest to them.

The method of designing the student-centred course will be described verbally, initially. Specific tasks will then be identified, and the feasibility of performing these tasks will be discussed.

4.3.1 Selecting the modules

The student will identify an area of interest to him, by quoting a SUPERCLASS category. This may be fairly specific - "I want to be able to type" - or more general - "I want to work in an office".

The designer then looks up the Catalogue of Module Descriptors to locate the general area in which relevant modules may be found. In this example, he would look under Section A, 'Business and Management' and then Section A8, 'Office and Secretarial Skills'. This lists 37 modules (SCOTVEC 1992). Some of these modules may be rejected because the student has already completed them at school. Of those that remain, the student may reject some after simply looking at the titles; he may decide, for example, that 'Audio Typewriting - Legal', is too specialised. He may accept some modules without further consideration, for example, 'Audio Typewriting'. He will probably find that some modules cannot be either accepted or rejected without further information. This information can be found from the Module Summaries.

On the basis of this information the student must accept or reject each of the suggested modules. As a result of these deliberations a 'Module List' can be written, listing the name and reference number of each module that the student intends taking. The Module Summary for each module in the list must now be examined to find the length of the module and whether it has any entry requirements. If a module has entry requirements, they must be added to the list unless, (a) the student already satisfies these requirements because of his 'previous qualifications', or, (b) the entry requirements have already been entered in the list. The Module Summary for each of these entry requirements must now be checked and a note made of its length and any entry requirements. This process continues until none of the additions to the Module List have entry requirements.

The entire process, starting from the initial statement of an 'area of interest', can now be repeated until the number of modules identified is large enough to constitute a valid course of study. The student should be encouraged, at some point, to include modules from the 'leisure and recreation' field.

Before discussing the feasibility of this approach to course design, two additional points must be made.

a) A module may be relevant to a number of interest areas, but each module appears in only one category of the Catalogue of Module Descriptors. If that category happens to be the one identified by the course designer from the details given by the student, then the module will be found. The module will also be found if it is named as an entry requirement to a module in the Module List. If neither of these conditions is true, the module will be added to the student's course only if the course designer is sufficiently familiar with the Catalogue to be aware of it. This is not a serious omission, in that the course will still be viable (entry requirements have been included) and the course is still constructed from modules relevant to the student's needs, but it is a flaw in the method.

b) Before a module is added to the List, two checks are made :- "Is it already in the List?", and "Is it one of the student's 'previous qualifications'?". If the answer to either of the questions is "Yes", the module is not added to the List. Checking the module against other modules is easy - "Do the reference numbers match?" - but checking it against SCE subjects requires that the course designer list the modular equivalents of each subject, i.e. the list of modules which together teach the same skills as the SCE subject. These equivalences have been agreed between SCOTVEC and the SCE - an example is given in Appendix 6. Failure to check these lists will result in the student repeating, under a different name, work he has already completed. This would be undesirable because it wastes time that the student could be using to improve his skills, and because of the effect on motivation of repeating work already satisfactorily completed.

The verbal description above is given because of the need to explain and elaborate on some of the tasks. However, the process can be 'formalised' and given as a list of instructions to the course designer. In converting the verbal description into a set of instructions, it will be realised that the process, as described above , is not in a logical order and one step has been omitted.

The output from the process is the Module List. As this list will be retained, either for future discussion or as a record of the modules to be attempted by the student, the student must be 'identified' and the list associated with him. In simple terms this means writing his name at the top, but will also, probably, include a college identifier, e.g. his matriculation or student number.

Input to the process comes from various catalogues, e.g. the Catalogue of Module Descriptors, and from the student - specifically, the area(s) of interest and his present qualifications. As the course designer will have to refer to the present qualifications each time he is about to add a module, he must write down these qualifications - this list will be called the 'Pre-entry List'. Unless the course designer trusts his memory, he will also write down the reference numbers (at least) of modules being considered for inclusion - the list of these modules will be called the 'Possibles List".

The overall task of constructing the Modules List can, therefore, be decomposed into three tasks;

a) construct the Pre-entry List;
b) construct the Possibles List;
c) construct the Module List

Figure 4.3.1.1 shows a set of instructions for constructing the Pre-entry List.

Figure 4.3.1.1

Constructing the Pre-entry List

```
Note the student's 'identifier'.
List the student's existing qualifications in the Pre-entry List.
REPEAT, for each SCE subject
:   look up the SCE Equivalent and add these modules to the Pre-entry
:   List.
END REPEAT
```

It is hoped that the instructions given in this form are in sufficiently plain English to be understandable to the non-specialist, and succinct enough that the overall 'form' of the task is clear. In particular, it should be noted that any instructions within a REPEAT....END REPEAT loop must be performed repetitively - in the case above the instruction will normally be carried out seven or eight times(3).

Constructing the Possibles List is a temporary phase leading to the construction of the Module List, and is therefore contained within the latter. Instructions for constructing the Modules List are shown in Figure 4.3.1.2.

Figure 4.3.1.2

Constructing the Module List (A)

```
REPEAT for each area of interest :-
:  Note the student's area of interest.
:  Locate the relevant category and section of the Catalogue of
   Module Descriptors.
:  Add the relevant modules to the Possibles List.
:  REPEAT until the Possibles List is empty :-
:  :  Look at the first module in the Possibles List.
:  :  IF it is in the Pre-entry List OR it is in the Module List, erase it
          from the Possibles List,
   :  ELSE add it to the Module List, and erase it from the
          Possibles List.
:  :  Look up the Module Summary and note any Entry
           Requirements, adding them at the end of the Possibles List.
:  END REPEAT
END REPEAT
```

The result of this process will be a list of modules, all of which are relevant to the student's interests, either directly or because they are necessary entry requirements. The student is now given the opportunity to delete some of the modules. He may feel that some are unnecessarily specialised, or are simply not of interest to him. Before a module can be deleted, however, the possible consequences must be considered - if the module is an entry requirement to another, then that must be deleted as well (and so on - the second module deleted may itself be an entry requirement to a third).

To avoid having to look up the Module Summaries again, it would be advisable for the course designer to keep a note of any 'sequences' that are built up. For example, suppose that module 'A' is identified as relevant, and it has an entry requirement 'B'. This should be noted as 'B $\rightarrow$ A', meaning 'B is an entry requirement to A'. If B has an entry requirement, 'C', the note would be amended to read 'C $\rightarrow$ B $\rightarrow$ A'. If the student now decides to delete 'C' from the Module List, it can be seen immediately that 'B' and 'A' must also be deleted. Given that information the student may decide to keep 'C' in the List.

These sequences can be more complicated, because a module can have more than one entry requirement.

A sequence such as

$$\begin{tikzcd}
 & & & & F \\
 & C \arrow[r] & B \arrow[r] & A \arrow[ur] \arrow[dr] & \\
D \arrow[r] & E \arrow[ur] & & & G
\end{tikzcd}$$

, for example, is possible.

The instructions to the course designer are shown in Figure 4.3.1.3, below.

Figure 4.3.1.3

Deleting Modules

```
REPEAT until the student is satisfied with the Module List;
:   Note the module to be deleted.
:   Check the sequence diagrams.
:   IF the module appears in a diagram or diagrams,
:   :   REPEAT, for all diagrams in which the module appears;
:   :   :   Note all the modules that follow it.
:   :   :   IF the student confirms his decision, delete all the modules,
:   :   END REPEAT
:   ELSE delete the module.
END REPEAT
```

Adding modules and deleting modules have been shown as two separate processes, but a satisfactory Module List is unlikely to result from one application of each. It is more likely that the student will add modules, delete some, add some more, and so on until he is satisfied. The complete task of constructing the Module List can now be described, by combining the processes as shown in Figure 4.3.1.4, below.

Figure 4.3.1.4

Constructing the Module List (B)

```
Construct the Pre-entry List                       (Figure 4.3.1.1)
REPEAT until the Module List is satisfactory
:   Add modules to the Module List                 (Figure 4.3.1.2)
:   Delete modules as appropriate                  (Figure 4.3.1.3)
END REPEAT
```

While this description of the task is succinct, being merely the repetition of various defined sub-tasks, a more 'expanded' description shows more clearly the scale of the work involved.

Figure 4.3.1.5

Constructing the Module List (C)

```
Note the student's 'identifier'.
List the student's existing qualifications in the Pre-entry List.
REPEAT, for each SCE subject :-
:  look up the SCE Equivalent and add these modules to the Pre-entry
:  List.
END REPEAT
REPEAT, until the Module List is satisfactory :-
:  REPEAT for each area of interest :-
:  :  Note the student's area of interest.
:  :  Locate the relevant category and section of the Catalogue of
   :  Module Descriptors.
:  :  Add the relevant modules to the Possibles List.
:  :  REPEAT until the Possibles List is empty :-
:  :  :  Look at the first module in the Possibles List.
:  :  :  IF it is in the Pre-entry List OR it is in the Module List, erase it
            from the Possibles List,
   :  :  ELSE add it to the Module List, and erase it from the
         Possibles List.
:  :  :  Look up the Module Summary and note any Entry
           Requirements, adding them at the end of the Possibles
          List.
:  :  END REPEAT
:  END REPEAT
:  REPEAT until the student is satisfied with the Module List;
:  :  Note the module to be deleted.
:  :  Check the sequence diagrams.
:  :  IF the module appears in a diagram or diagrams,
:  :     REPEAT, for all diagrams in which the module appears;
:  :     : Note all the modules that follow it.
:  :     : IF the student confirms his decision, delete all the modules,
:  :     : ELSE don't delete any.
:  :     END REPEAT
:  :  ELSE delete the module.
:  END REPEAT
END REPEAT
```

The processes described above are not difficult, in the sense that none of the steps requires great expertise or specialist knowledge. However, there are problems associated with carrying out these steps manually.

a) Modules have five or seven character references (see 3.5.1 and 3.5.7), but the modules are not written in reference number order in the

Catalogue of Module Descriptors. To locate a given module it is necessary to know the category and section of the Catalogue where it has been placed. The course designer is therefore working with lists of seven or nine character references. Any mistakes in copying these references may make it impossible to find the module in any of the catalogues. A mistake in only one character may allow the module to be found, but it will be the wrong module. The course designer must, therefore, be very careful when writing these reference numbers.

b) If the course designer only notes the reference numbers, the student cannot tell what the module is about - he must refer to the Module Summary for information. Given the title, the student may be able to decide to accept or reject the module without looking at the Module Summary. Also, writing the title gives an additional check that the correct module has been noted. It is therefore useful for the course designer to write the module title as well its reference number. Simply writing down, say, thirty module references and titles is time-consuming - and, as noted above, it must be done carefully.

c) There are about three thousand modules available. Each one has a one page Module Summary. The catalogue of Module Summaries is, therefore a three thousand page document - about eight times thicker than this thesis. If the document is sub-divided in the same way as the Catalogue of Module Descriptors, the Summaries will be arranged in seventeen volumes, one for each category. Tracing the entry requirements for a module may require looking in different volumes (for example, a module on office procedures may have English and Arithmetic entry requirements). The process, and the time taken, is similar to looking up a number in one telephone directory and being referred to two others where other numbers have to be found - and this has to be repeated for every module in the Module List.

d) The number of modules in a course varies, depending on the length of the modules included. The worst case would contain forty-eight half-modules. However, in arriving at these, other modules will have been identified and rejected. While this is a worst case, the most likely case course will contain twenty-four modules, with others having been rejected. The process of looking up the Module Summaries may have been repeated thirty to forty times before the course is designed.

e) At various points the course designer may become involved in conversation with the student about the relative merits of different modules. This will interrupt the sequence of steps he is following. Under these conditions the designer may well make mistakes - forgetting to look up the entry requirements for a module, for example.

f) In the processes described above, no mention has been made of the need to provide breadth and articulation.

To summarise :- the individual steps are trivial and must be repeated a large number of times. The process is therefore time-consuming and tedious. The steps must be carried out in sequence, but the sequence may be interrupted. Under these conditions, of boredom and breaks in concentration, mistakes are possible (if not probable).

4.3.2 Fitting modules into the Course Grid

As described above, the process of constructing a programme starts by constructing a list of modules which may be in the final programme. The procedure for checking that the combination of modules can be fitted into the Course Grid is best illustrated by a simplified example. For the purposes of the explanation, the following notation will be used :-

M1 is a module that is not an entry requirement to any other;

E1M1 is an entry requirement to module M1; E2M1 is another entry requirement to M1.

E1E1M1 is an entry requirement to E1M1

Figure 4.3.2.1, below, shows, on the right hand side, the set of modules to be fitted. On the left the partially completed Course Grid is shown.

Figure 4.3.2.1

Fitting Modules into the Course Grid (A)

Block 1	Block 2	Block 3	
			M1 M2 M3
M1			
M2			
M3			
E1M4	M4		E1M4 → M4
E1E1M5	E1M5	M5	E1E1M5 → E1M5 → M5
E2E1M5			E2E1M5
E1E1M6	E1M6	M6	E1E1M6 → E1M6 → M6
E1M7	M7		E1M7 → M7
			E1E1E1M8 → E1E1M8 → E1M8 → M8
			E2M8
			E1E1M9 → E1M9 → M9
			E1M10 → M10

The modules M1 to M7 have been taken in the order in which they appear on the right, and have been fitted into the first available space in the Course Grid. Placing M1, M2 and M3 in the 1st block can be partly justified on the grounds that, as they have no entry requirements, they may be 'basic' modules. If that is the case it is desirable to place them in the first block, as they allow a 'settling in' period for the student. (This point is not particularly important.)

Before attempting to fit the next sequence (ending in M8) it will be noted that M8 itself cannot be fitted, because the sequence is too long. This could probably have been recognised at an earlier stage. To fit the earlier part of the sequence (from E1E1E1M8 to E1M8, and E2M8) space must be created in block 1 (space already exists in blocks 2 and 3). This can be done by moving one of a number of possible modules. Following the argument given above, M1, M2 and M3 are left in block 1. It is decided to move the M4 sequence, simply because this is the next sequence encountered in the Course Grid. The M7 sequence could also have been moved. The M8 sequence is then fitted, resulting in the grid shown below (Figure 4.3.2.2).

Figure 4.3.2.2

Fitting Modules into The Course Grid (B)

Block 1	Block 2	Block 3
M1	E1M4	M4
M2		
M3	E2M8	
E1E1E1M8	E1E1M8	E1M8
E1E1M5	E1M5	M5
E2E1M5		
E1E1M6	E1M6	M6
E1M7	M7	

M1 M2 M3

E1M4 → M4
E1E1M5 → E1M5 → M5
E2E1M5
E1E1M6 → E1M6 → M6
E1M7 → M7
E1E1E1M8 → E1E1M8 → E1M8 → M8
E2M8
E1E1M9 → E1M9 → M9
E1M10 → M10

Fitting the sequence ending in M9 requires another space in block 1; again resisting the urge to move M1, M2 or M3, the M7 sequence can be moved to the right, as shown below (Figure 4.3.2.3), and the M9 sequence fitted.

Figure 4.3.2.3

Fitting Modules into the Course Grid (C)

Block 1	Block 2	Block 3
M1	E1M4	M4
M2	E1M7	M7
M3	E2M8	
E1E1E1M8	E1E1M8	E1M8
E1E1M5	E1M5	M5
E2E1M5		
E1E1M6	E1M6	M6
E1E1M9	E1M9	M9

M1 M2 M3

E1M4 → M4
E1E1M5 → E1M5 → M5
E2E1M5
E1E1M6 → E1M6 → M6
E1M7 → M7
E1E1E1M8 → E1E1M8 → E1M8 → M8
E2M8
E1E1M9 → E1M9 → M9
E1M10 → M10

Finally, M10 can only be fitted after moving M1, M2 or M3 - the *possible* educational justification for keeping these in Block 1 is subordinate to the *definite* fact that the chosen modules cannot be fitted unless one of those basic modules appears in a later block.

Figure 4.3.2.4

Fitting Modules into the Course Grid (D)

Block 1	Block 2	Block 3
M1	E1M4	M4
M2	E1M7	M7
$\boxed{\text{E1M10}}$	E2M8	$\boxed{\text{M10}}$
E1E1E1M8	E1E1M8	E1M8
E1E1M5	E1M5	M5
E2E1M5	$\boxed{\text{M3}}$	
E1E1M6	E1M6	M6
E1E1M9	E1M9	M9

M1 M2 M3

E1M4 → M4
E1E1M5 → E1M5 → M5
E2E1M5
E1E1M6 → E1M6 → M6
E1M7 → M7
E1E1E1M8 → E1E1M8 → E1M8 → M8
E2M8
E1E1M9 → E1M9 → M9
E1M10 → M10

Having worked through this example, the following points can be made;

a) The fact that there are fewer modules than spaces does not guarantee that all the modules can be fitted (M8 could not be fitted even though there were spaces available at the time).

b) The sequence ending in M8, which had to be deleted, could have been recognised as impossible while the Module List was still being constructed - the maximum length of a sequence is 3, because of the width of the Course Grid. M8 could have been deleted at that stage, before the Course Grid was considered. The student and designer would then have realised that they did not have twelve viable modules, and might have looked for an alternative to M8.

c) There is no significance to the row in which a module is placed.

d) An entry requirement must precede the module to which it refers, but the modules need not be in adjacent blocks. (E1M10 is in block 1, but M10 is in block 3.)

e) The process is basically 'trial and error', with modules being given an initial position and then, possibly, being moved in an attempt to create space for others.

f) Because of (a) and (e) there is no guarantee that a particular set of modules can be fitted into the Course Grid, even when no sequence is longer than three blocks and the number of modules does not exceed the space available. In attempting to fit a module into the Grid, how many attempts at finding a solution should the designer make before he accepts (rightly or wrongly) that there is no solution; that the module cannot be fitted?

This example is a fairly straightforward case; in the real situation the following factors complicate the process;

a) In the example, modules are clearly identified as entry requirements because of the notation adopted - in reality the course designer is working with the reference numbers of the modules, which do not contain that information. He must constantly refer to the 'sequence diagrams' to ensure that any sequences are entered correctly.

b) In a real course the modules vary in length from a half module (occupying only half a block) to three times the normal length (occupying an entire row). For this reason, it was suggested, in 2.1, that the Course Grid be shown with each block divided into two, and that the module lengths should be expressed as multiples of the half-module (see figure 2.1). The 'real' Course Grid is therefore twice as big as that used in the example. In addition, when attempting to fit modules, constant reference must be made to the lengths of modules as well as their appearance in sequence diagrams.

d) Because of the variation in length, it may not be possible to fill the Course Grid. For example, a whole module space may be empty, but the remaining module may be one and a half modules long. It may be desirable to delete a half module from the grid to allow the one and a half module to be fitted, thus giving a full Course Grid (and a full course for the student).

When 'choosing modules' was discussed a set of instructions was presented (Figure 4.3.1.5). A set of instructions to allow modules to be fitted into the Course Grid will not be derived at this point. It is felt that the example given, with the comments on the real situation, show that the process would be time-consuming and, possibly, even more likely to result in mistakes. A set of instructions for this process, which reduce the amount of trial and error, will be described later, in Chapter 7. It will be shown there that a method that does not rely on guess-work, hunches or intuition necessarily involves an amount of note-taking and repetitive computation that would render the method impracticable if attempted manually.

4.3.3 Selecting the best course

The possible courses resulting from the processes described above all satisfy the basic criteria;

a) the modules are relevant to the student's interests;
b) the student has deleted any modules which, though relevant, are not to his liking;
c) all necessary entry requirements have been included;
d) no modules or their equivalents repeat work previously completed by the student;
e) all the courses can be completed in the time available.

(c), (d) and (e) may be said to be 'absolutes'; each statement is either true or false, and if any statement is false the course would not be considered an option. These criteria cannot therefore be used as a means of distinguishing between possible courses. "Relevance' and 'liking' are not absolute; a module may be more or less relevant, a student may like one module more than another. Assuming, for the moment, that scores for relevance and liking can be assigned to each module (e.g. this module is 0.8 relevant, that module is only 0.7) the 'value' of a course could be calculated by aggregating the values on both criteria for all the modules in the course. This would then give an overall score for the course. Two or more courses could then be compared and the one with the highest score chosen.

If this approach is to be adopted, the following questions must be answered;

a) Can the course designer assign scores for relevance? Is it possible to distinguish between 0.8 and 0.7 relevance? Should we settle for 'broad banding', i.e. modules are 0, 0.25, 0.5, 0.75 or 1.0 relevant, or some other scale?

b) Similar questions may be asked of the student concerning his liking for one module rather than another. In addition, should the scale allow negative values? That is, should we allow for the student actively disliking a module in an otherwise acceptable course?
c) How should the course designer rate a module which is included because it is an entry requirement to a highly relevant module, but is disliked by the student?

The approach can also be criticised because each module is considered in isolation; there is no consideration of the course as a whole.

A more satisfactory, and more complicated, approach will be described in Chapter 6. At this stage it is enough to state that even this simple approach, considering only two variables, will involve the course designer in allocating values to modules and then summing these values for the course - for a course containing thirty modules, the course designer must add sixty values together. Given a calculator this is not a particularly time-consuming task, but keying that number of values is likely to result in errors. The process must be repeated for a number of courses, increasing the time required. Given that all courses being considered will satisfy the basic requirements, the difference in overall value between courses may be small, in which case the wrong decision may ultimately be made. This is not serious, as an acceptable course will still have been chosen, but does raise the question, "If the wrong decision can result, is it worth evaluating the courses at all?"

4.3.4 Conclusion to Chapter 4.3

The overall task, of selecting modules, arranging them into viable courses, and evaluating the courses so that the best can be chosen involves a large number of repetitive, low level operations.

Performed manually,

a) the task would be so time-consuming as to render it impracticable.
b) a number of errors are likely to occur. Some of these errors will result in 'fatal' flaws, e.g. where an entry requirement has been omitted. The

existence of these errors requires that the task be repeated. Other errors, though not fatal, reduce the value of the exercise, in that the best course would not have been chosen.

c) the low level of skill required for each operation makes this an inappropriate task for skilled personnel.

However, these same factors make this an ideal subject for a computer program.

a) Computers can perform simple tasks very quickly - very large numbers of repetitions can be performed in an acceptably short time.
b) Computers do not become bored, lose concentration and make mistakes.
c) The cost of even a sophisticated desk-top computer (and it will be shown that a relatively simple computer can perform the tasks required) is considerably less than the cost of an F.E. lecturer. The lecturer can be released to perform more appropriate tasks which will use the skills he is paid to possess.

4.4 The Qualification-centred design process

In the previous section the student-centred design process was described. The more common, and more restrictive, qualification-centred course will now be considered. In this case, the student specifies a recognised qualification. The modular equivalent of the qualification is then examined. Because of the highly prescriptive nature of most of these equivalences, most of the modules required to satisfy the needs of the qualification will already be listed. (See Chapter 2.9, "Equivalences")

4.4.1 Selecting Modules

The task of selecting modules for this course reduces to :-

a) Note the compulsory modules listed in the 'equivalent'.
b) Delete from the list any module that the student has already completed.
c) Choose between the optional modules, of which there will probably be very few. If the student cannot base a decision on the title alone, he

may want to look at the Module Summary.

d) Choose the few modules required to complete a full course, possibly from 'leisure and recreation'. Although many of the equivalences leave little unused space in a programme, the student may have already completed some of the required modules. Omitting these modules, (see (b), above), to avoid repetition of work, leaves gaps in the programme. This leaves scope for more choice on the part of the student.

A possible set of instructions for the course designer is given below, in Figure 4.4.1.1

Figure 4.4.1.1

Selecting Modules for a Qualification

```
Construct the Pre-entry List                          (Figure 4.3.1.1)
Note the qualification sought by the student.
Look up the modular equivalent of the qualification.
Copy the modular equivalent - this is the Possibles List.
REPEAT for each module in the Possibles List :-
:       Look at the first module in the Possibles List.
:       IF it is in the Pre-entry List, delete it from the Possibles List.
END REPEAT
Deal with the Optional Modules.                (See note, immediately below)
Add modules to complete the programme                 (Figure 4.3.1.2)
```

Note :- The instructions vary from one qualification to another, and are given with the list of compulsory modules.

This appears to be a longer task than that described earlier for the student-centred course ("Add modules to complete the programme" represents a page of instructions). However, the following points can be made :-

a) The designer does not have to identify the majority of the modules in the course - they are already listed.

b) The Equivalents list contains all the compulsory modules, including their entry requirements, so the designer does not have to look up the Module Summaries for the majority of the modules.

c) Where an optional module has an entry requirement, the two are listed as a pair - "choose one, choose both". Again, the designer does not have to look up the Module Summary to identify entry requirements.

d) The time-consuming part is "Add modules to complete the programme", but these steps are only required for a small number of modules.

Selecting modules for this type of course is therefore less time-consuming than for the more general case described earlier, in 4.3.1, because most of the modules will already have been selected. The procedure for identifying the unspecified modules is the same as was described for the more general case, with one difference - the student cannot delete from the course any module specified in the equivalence list, as this would prevent the student satisfying the requirements of the qualification. The method explained in 4.3.1 can be used, provided the course designer notes those modules listed as compulsory to avoid their accidental deletion.

At this point the reader may put forward the following argument :-
Most existing courses are designed to meet the needs of qualifications. These courses can be designed using the same method as described earlier. But it was stated, in 4.3, that no description of the process had been found. This is in accordance with the findings of others :-

> *"Issues of whole-programme design have been relatively neglected in the National Certificate. In the original presentation of Action Plan, there was perhaps an over-emphasis on the new principle of modularisation at the expense of issues of whole-programme coherence."*
> (SED, 1991, p56, para. 7.11)

Speaking of the theoretical case for a modular approach :-

> *"Not only are there no answers available to relevant questions, there is no systematic consideration of what the relevant questions might be;....."*
> (Jonathan 1997, p87)

These courses must have been designed, so why is there no description of how they were designed?
The answer is that these courses were designed 'in reverse'. Before the

development of the National Certificate the courses already existed, having been built up over a number of years. As a result of the Action Plan the courses were broken down into modules; simplisticly, we could say that they took a course, divided it into forty hour units and gave each unit a different name - these were the first modules to be produced. Effectively nothing had been changed. The course 'worked' before and continued to 'work' after being modularised.

> *"....though the Business and Technician Education Council fears that modules may not be developed in their own right as part of a total development, but rather by sub-dividing existing curricula, ignoring coherence, repetition and inter-relationships."* (Jonathan 1987, p88)

> *"Every course has been analysed and redesigned in the form of 40-hour modules."* (Gatherer 1989, p119)

That is a rather superficial description of the process of changing from the previous courses to the modular equivalents. Nevertheless, the problem of designing a course from that starting point is completely different from that of designing a new course from existing modules. As an analogy, dismantling an existing washing machine and listing the components is completely different from building a new washing machine using existing components designed for other washing machines.
Since the start of the National Certificate new qualifications have been added, but where no suitable modules were found new modules were designed to fit the course (that is one reason for the increase in the number of modules available). Designing new modules to meet the needs of each student is not feasible, so that approach is not applicable to the problem of designing individual courses tailored for individual students.

4.4.2 Fitting Modules into the Course Grid

Most of the modules in the course are specified by the nature of the qualification, and because of the way those modules were developed (as explained above), they will fit into the Course Grid. There only remains the task of fitting the relatively small number of modules that the student is free to choose. Fitting these is accomplished in the same way as was described for the student-centred course. Again, it might be thought that, if modules are fitted by that process then there must be a description of the process somewhere. In fact it appears that in existing courses modules are fitted by reference to a pre-determined list which has been timetabled to ensure that it can be fitted - see Appendix 3.1.

4.4.3 Evaluating the course

The method described for the student-centred course can be used to choose between variations on a qualification-centred course. However, most of the modules will be determined by the qualification - the only difference between courses designed for a specific qualification will lie in the optional modules and those used to complete the programme. The optional modules (those specified in the qualification from which the student must choose) are unlikely to differ to any great extent, as they will have been identified as necessary to complete the requirements of the qualification. The main difference will therefore lie in those modules which the student has chosen freely. If the course consists mainly of the prescribed modules, then a 'good' course can be completed by selecting leisure/recreational modules or, possibly, more academic modules not related to the qualification itself. A 'bad' course would be one in which the student has completed the programme by choosing modules closely related to the qualification, as this will have no breadth or variety in it. However, the decision as to which type of module to use in completing the course is probably best made after consulting a counsellor, rather than by evaluating the course numerically. The conclusion is that, although the qualification-centred course could be evaluated using the same technique as for the student-centred course, it is probably not worth the effort or time as the difference between courses will be small, and criteria used for the evaluation may not be as appropriate.

4.4.4 Conclusion to Chapter 4.4

The qualification-centred course can be designed using the same techniques as for the student-centred course, but because many of the decisions have already been made before the course designer becomes involved, the process will be less time-consuming and some features of the design process, e.g. evaluation, might be considered irrelevant.

If the qualification-centred course were the only type of course to be designed there would be no need to 'invent' a method for designing courses - this may be why there are so few references to course design in the National Certificate.

4.5 The Higher education-centred design process

In this case the student wants to follow a course in the higher education field, but lacks the entrance qualifications. He is therefore attending the college to gain those entrance qualifications.

4.5.1 Selecting modules

Certain modules will probably be specified, perhaps indirectly; for example, entrance to a degree course in Engineering may depend on having Higher-grade Mathematics. The modules deemed to be equivalent to this subject will therefore be required. The number of modules specified in this way will vary according to the student's previous academic record. If a large number are prescribed the process of designing the course is very similar to that of designing the qualification-centred course. If very few are specified, because the student already has most of the entrance requirements, then the process becomes more similar to that required in designing the student-centred course. As with the qualification-centred approach, the necessary modules must be noted to avoid their accidental deletion.

It may be noted that the specification of the target group, in 4.2.1, while intended to include the largest group of students, those moving to college straight from school, does not preclude the student who left school some years earlier and has now decided to re-enter formal education. There are probably relatively few students transferring from school to college with the immediate intention of gaining entrance to higher education - these students are more likely to stay on at school and sit Higher-grade examinations.

4.5.2 Fitting Modules Into the Course Grid

Although the process of selecting modules may vary between the student-centred and the qualification-centred approach, fitting modules into the Course Grid for these students is more likely to follow the approach of the student-centred design. This is because there will probably be more variation in the modules selected than in the qualification-centred course, and it is less likely that a pre-designed course will be available.

4.5.3 Evaluating the Course

The main purpose of this course is to acquire the necessary module passes to allow entry to higher education. If a variety of courses can be designed they will all contain these necessary modules and they will not offer any basis for discrimination between courses. The evaluation will therefore concentrate on the other modules, those not prescribed. There could, however, be some argument as to the criteria to be used in determining the 'best' course. If the student has been out of formal education for some time he may feel the need to choose modules that will complement those that he must take, rather than aim for breadth, in order to 'brush up' on skills that he may not have used for some time - some students may need to re-learn the habits of regular study and organisation of time in preparation for entering higher education.

4.5.4 Conclusion to Chapter 4.5

The approach to designing the higher education-centred course will vary according to the number of modules deemed to be compulsory. With a small number of compulsory modules the task is similar to that of designing the student-centred course; as the number of compulsory modules increases the task resembles that involved in the qualification-centred course, and the criteria by which a best course can be identified may be less easy to define mathematically.

4.6 Conclusion to Chapter 4

Three types of course were identified at the beginning of this chapter. Following the comments on the qualification-centred and higher-education centred courses it can be now be argued that there is a spectrum of courses, with the student-centred at one end, the qualification-centred at the other, and the higher education-centred lying somewhere in between (its exact location depending on the number of essential modules).

A method for designing the most flexible type of course, the student-centred, has been described. This approach can be used for all courses, (with a small alteration to be described below), but the extent to which it is useful or necessary depends on the position of the course along the spectrum.

Where the method is used in full, to design a complete course with maximum choice, it is impraticable to do it manually - it would be too time-consuming, prone to error, and an inappropriate use of a lecturer's expertise.

On the other hand, as the individual tasks consist mainly of searching for data and performing relatively simple but repetitive arithmetic calculations, a computer appears to be an ideal tool.

The 'small alteration', referred to above, is that the method described in Figure 4.3.1.1 should be extended to allow for modules to marked as 'essential'. In the qualification-centred course, all of the compulsory modules and the chosen optional modules would be marked in this way to prevent their deletion. In the higher education-centred course the modules directly related to admission to higher education would be so marked, for the same reason. This feature is also of value for the student who is not constrained by the requirements of a 'third party', (the course designer being the second). By choosing to mark certain modules as 'essential' the student can tell the course designer that they must be included - in particular, if there is any difficulty fitting all the modules into the Course Grid, the designer will fit the essential ones first, thus ensuring that all the courses based on the identified modules contain those considered most desirable.

4.7 Notes

1) It is the design of the course that is described as student-centred; the phrase does not imply that the methods used in delivering the modules are student-centred.
2) In the case of the student-centred course, the meaning of "articulation" is restricted to "a progression from general to more specialised modules within the course". (See 2.11)
3) The vast majority of school pupils now take Standard grade examinations in their fourth year. These are intended to be accessible to all pupils in the normal secondary school, whatever their ability. Virtually all schools offer either seven or eight standard grade subjects to each pupil, and an award is made no matter what the pupil's performance in the examination. Each pupil who has attended S4 will therefore have an award, at some level, in seven or eight subjects.

Chapter 5
Choice and Negotiation

5.1 Introduction

The intention of the Action Plan was to increase the amount of choice open to students. Some writers appear to have taken the view that the National Certificate, because it does not allow free choice, is a failure. In this chapter it will be argued :-

a) that certain factors will always constrain choice in the construction of modular programmes,

and

b) that these factors were known to the authors of the Action Plan, from which we can imply that they did not intend totally free choice.

c) that choice is limited by other factors which are not inevitable but are dependent on the methods used to construct programmes of study; these limitations on choice can be reduced by the introduction of more appropriate methods.

The Shorter Oxford dictionary gives various meanings of choice, of which the most relevant to this discussion are;

a) "*preferential determination between things proposed*",

and

b) "*the preferable part of anything*"

In abstract terms these imply;

a) that a list must be presented, from which some parts will be selected and others rejected;

b) that the selection should be based on the relative value of the members of the list, according to some criterion or criteria.(1)

In the context of the National Certificate, what is the student to choose between?

He must have chosen to participate in further education, as this is voluntary;

he must have chosen a college; and he must have chosen a qualification to be gained or a course to be followed.

But these were the choices under the old system; the Action Plan implied more than these. It was clearly the intention of the Action Plan that choice should be applied to, among other things, the modules in the student's course. Speaking of the modular framework it says,

> "....; *it will also allow young people to negotiate individual programmes of study,....* " (SED Jan 1983)

5.2 Factors tending to limit choice

The factors constraining choice will now be listed and described. As will be seen, there is considerable overlap between some of these factors; the fact that factors are described under separate headings should not be taken to imply that they are independent.

5.2.1 The number of modules available

The SCOTVEC catalogue listed 2,969 modules for use in session 1992/93. The first constraint on choice may be claimed (somewhat pedantically) to be that the student cannot choose a module that does not exist. The point is hardly worth debating, but if the meaning of choice is taken to be "preferential determination between things proposed", (Shorter Oxford dictionary) then it can be argued that the restriction to modules in the catalogue is not a limit on choice; the 'things proposed' are the modules in the catalogue - choice comes after the catalogue is established.

5.5.2. The number of modules that the student can study

There is a limit to the number of modules that a student can study over any particular period - for even the most hard-working student, there are only twenty-four hours in a day and seven days in a week. The Course Grid was introduced in Chapter 2 and 'used' in Chapter 4. If this version of the Grid is adopted the student is limited to the equivalent of twenty-four modules. The actual number of modules is a function of the length of the individual modules.

5.2.3 Breadth

"....it is desirable that students develop broad competencies including communication skills, interpersonal skills, planning and problem-solving,..."

(CAST 1987, p22. Also quoted in Chapter 3.6)

If a student is required to study modules relating to these areas, then choice has been limited, because (a) the student has to study modules he might not want to, and (b) because the time available for the modules he *does* want to study has been reduced.

"In such a programme the desirability of choice and the desirability of breadth and balance may temper rather than complement each other."

(Jonathan 1987, p90)

5.2.4 Articulation

".....the Action Plan used the word articulation to mean the process by which syllabuses supply progression from basic to more advanced stages, and that this was desirable." (see Chapter 2.4.2)

The need for articulation, both within the course and towards later, more advanced work, again restricts choice;

a) The student is expected to take modules which are at a more advanced level than modules taken earlier.

b) The student must initially choose modules that allow this progression, i.e. he must choose modules that are entry requirements to other modules.

c) The time occupied by these modules is not then available for other modules which may be more to his liking.

"Indeed, for both progression and coherence, choice must become increasingly constrained, since earlier choices restrict the range of later ones." (Jonathan 1987, p95)

It is arguable whether choice is *increasingly* constrained; because of the existence of entry requirements it is possible for choice to become *less* constrained, as earlier modules open routes to later modules which would not otherwise be possible. Nevertheless, any requirement that students take a specified type of module is a restriction on choice (2).

5.2.5 Integration

Integration, by which the common components of a number of modules are extracted and taught separately, can restrict choice in the same way as do entry requirements; indeed, the process of integration creates a module which is a prerequisite to a number of others (see Chapter 3.7.4). Where the process is applied to a larger group of modules then they all become interdependent and flexibility and choice are lost .

> *"Where such schemes have been in existence for some time (in higher education in Britain and high schools in the USA), it has been universally found that structures harden over time, with well-marked pathways forming through options and crypto-cores emerging, such that a programme of modules comes to resemble something very like a 'course'."*
>
> (Jonathan 1987, p88)

However, if integration is applied on a smaller scale it can be argued that very little freedom is sacrificed. Consider a number of modules, A, B, C and D, which all have a common element; for example, they might all contain elements of basic arithmetic. By integrating them the common element can be extracted and will appear as a fifth module, E, which is an entry requirement to the other four. These other modules are now re-written with the teaching of arithmetic removed. Suppose a student wanted to take module D. Before integration that would be possible; after integration he is now required to take module E first. To what extent is he being forced to take a module that he has not chosen? If he wanted to take module D (before integration) then he was accepting the need to learn arithmetic. The only difference after integration is that the arithmetic is taught separately. The danger in integrating too many modules is that the common element may be more diffuse. In the example given, if A, B, C and D contain *different* arithmetic skills, and the new module, E, teaches all of them (so that it can be an entry requirement to all four) then the student who was prepared to study the restricted range of arithmetic required by the old-style module D now has to study all the skills required by all of them. He is therefore being forced to study material he did not choose in order to be allowed to study the material he did choose. In mitigation, he has broadened his education by acquiring skills which he may find useful at a later date.

5.2.6 The student's educational background

Where a student has few relevant qualifications (SCE subjects or modules) his choices are limited by the need to attain the necessary entry requirements before studying certain modules. If, for example, a module 'A' is a necessary prerequisite to studying module 'B', and the student does not have a pass in module 'A', then he is barred from taking 'B' without first taking 'A'. It is tempting to say that the student with few relevant qualifications has fewer choices than a student with a number of appropriate passes. This, in fact, is not true. Continuing the example given above, if a student has a pass in 'A' he is normally barred from taking it again at college, because it is generally accepted that students should not repeat work already completed and, more positively, should be advancing their studies, i.e. the student should use his existing pass in 'A' as an opportunity to study 'B'. The student with a pass in 'A' will only have more choices if 'A' is an entry requirement to a number of other modules.

In some cases the possession of previous qualifications may itself be considered to act as a limit to choice - a student with a number of passes in Science modules might be expected to use these, by taking more advanced Science modules at college, rather than divert into Arts subjects of which he has no previous experience. Whether he would be **prevented** from taking Arts modules or not depends on the attitude of the college.

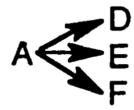
5.2.7 Entry Requirements

As explained above, most of the limit on choice associated with 'educational background' is the result of imposing entry requirements on some modules, and it is arguable whether the two can be considered separately. The situation can be ameliorated, to some extent, with careful module design :-

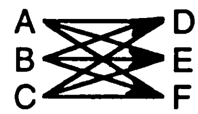
> *"But if the system is to offer choice, a given module must be a possible prerequisite for several others. Unit A, B and C must all contain certain prerequisite material for D, E and F, for if only A contained it, there*

would be no choice between A, B and C for those wishing to keep options open between D, E and F. The handling of prerequisites is therefore a crucial matter both for the maintenance of choice and for the efficient use of teaching time and resources." (Jonathan 1987, p94)

The first sentence can be represented as



while the second would be shown as



For clarity, only some of the arrow heads are shown. For all paths, the direction is left to right.

These are slightly different (but equally valid) points; the first (which repeats an observation in the preceding paragraph) shows choice increasing after 'A' has been completed. The second shows a further increase in choice (compared with the first) because the student is not constrained to take 'A' in order to gain access to D, E and/or F. However, if A, B and C are all equally valid as prerequisites to D, E and F, how different can they be? If they are very similar, can we claim to be offering choice? The answer to the first question depends on the other skills taught by A, B and C, i.e. those skills which are not prerequisites to D, E, and F. In the author's opinion, the answer to the second question is that there is no *meaningful* choice to be gained from offering, effectively, one module under a number of different names.

This example raises another issue, which will be discussed in the next paragraph.

5.2.8 Viability of classes

Colleges cannot afford to run small classes. Where a lecturer could be teaching a class of twenty it is an inefficient use of his time to have him teach a class of five. The Action Plan had criticised both the duplication of courses and the small class sizes found in some existing courses (SED 1983a p47). But consider the implications of the example given above. The probable similarity of A, B and C has already been highlighted. In addition, suppose that, given this choice of routes to D or E or F, five students chose A, five chose B and five chose C. These classes are too small to be allowed - if that results in all three classes being cancelled then the apparent choice offered by the overlap cannot be achieved in practice. Not only are the students prevented from taking the initial module, they are also barred from the second set (D, E and F) because they cannot gain the necessary entry requirements. By restricting choice to one of A, B or C, a viable class of fifteen results, which then opens up the choice of D and/or E and/or F. In the quotation given in the previous section, the last sentence is of interest.

> *"The handling of prerequisites is therefore a crucial matter both for the maintenance of choice and for the efficient use of teaching time and resources."* (Jonathan 1987,p94)

If 'efficient use of teaching time ....', means the maintenance of reasonable class sizes, then this would appear to be at variance with 'the maintenance of choice', as it implies that choice will be restricted to ensure that class sizes are maintained. However, as explained above, there are situations where it may be desirable to restrict choice in one area in order to increase choice in another. It is not clear what the author intended by this remark, but the first clause of the sentence is indisputably correct. The reader may already have noted the number of times that 'entry requirements' and their treatment have featured in this work. Although entry requirements are a problem, they are necessary :-

> *"It is stated that the art of managing a modular course is to avoid 'unthinkable situations' arising by using prerequisites to ensure that topics or skills thought to be essential have been previously covered."* (Ibid.)

Her next comment :-

> *"However, the larger the number of modules on offer, the more difficult this is; indeed the maximum number which permit it to be done rationally may be really quite small."* (Ibid.)

is perhaps based on the assumption that the modular course will be designed manually. That, of course, was the conclusion reached in Chapter 4.

5.2.9 Core Modules

There is one other interpretation of 'prerequisites', apart from specific entry requirements to certain modules. This involvės the idea that some modules are of importance to all students and should feature in all programmes.(3) They are prerequisites in the sense that the student will benefit from studying these modules at an early stage in his course. These 'core' modules are thought to be a feature of all modular programmes. Warwick, writing of modular courses in general rather than National Certificate courses in particular states :-

> *"To ensure that key elements are not omitted core modules are identified - these are compulsory for all pupils."* (Warwick 1987 p18)

This is another constraint on freedom of choice - students are not free to choose not to take these modules, and the presence of these modules in the programme limits the number of modules that can be then be chosen.

5.2.10 The student's needs

If the student intends to gain a recognised qualification, he is obliged to study the compulsory modules prescribed for that qualification. As shown in Chapter 2.9, some of these 'equivalents' of previously existing courses are highly prescriptive and allow very little choice, once the initial choice of the qualification itself has been made.

The government, responsible for the Action Plan, and the colleges, seeking to implement it, are not free agents. Further education exists to serve industry and commerce (among others), and their responsibility is to maximise profits, not necessarily to follow the principles laid down in the Action Plan.

> *"Many of the social benefits of modularisation require learners to respond to market forces - choice is constrained by employers rather than lecturers/colleges."* (Jonathan 1987, p94)
>
> The prescriptive nature of many of the programmes leading to recognised qualifications may be an indication that employers and professional bodies (who decide which modules must be studied) have not been convinced of the need to encourage choice.

5.2.11 Negotiation

The Shorter Oxford Dictionary gives the following meanings for 'negotiate';

> *To confer (with another) for the purpose of arranging some matter by mutual agreement;*
>
> *to discuss a matter with a view to a settlement or compromise;*
>
> *to deal with, manage, or conduct (a matter, etc. requiring skill or consideration).*

It follows, therefore that negotiation;

a) involves more than one person (it will be assumed that it involves a student and a lecturer)

b) is designed to achieve mutual agreement; this may involve compromise, which itself implies that neither participant can expect to get everything he wants.

c) involves discussion, the interchange of ideas.

d) may require skill in the negotiation process itself, (i.e. getting what is wanted) and skill in deciding what is wanted.

If negotiation implies the possibility of compromise, then the student does not have totally free choice. If there was any doubt as to the degree of choice intended by the Action Plan, the equal prominence given to negotiation should have settled the matter. Clearly the Plan did not intend that students should be given a totally free choice, even with the provision of advice and information to guide the student to good choices.

The use of the word 'negotiation' implies that the student must make some choices. There cannot be negotiation without choice; if there is no choice to be made there is nothing to negotiate.

Equally, negotiation implies that the lecturer must let the student make some choices; the lecturer must be prepared to compromise in order to reach agreement.

Negotiation need not necessarily imply a difference of opinion. If the student has chosen some modules, and the lecturer agrees that these are acceptable, then mutual agreement has been reached. It can be argued that in that case no negotiation took place, because none was necessary. If the process of negotiation is desirable as an educational experience in its own right then negotiation must be forced to happen. Where the lecturer and student are in agreement without negotiation, the only way that negotiation can be made to occur involves the lecturer deliberately advocating an alternative simply to be different, so that the student can be made to justify his choices. However, this writer feels that if the ability to negotiate is worth developing it should be done in a more appropriate educational context, rather than constitute an additional task to be added to the others involved in designing a modular programme.

Can there be choice without negotiation? Where the student has made the same choices as the lecturer would have made, there is no need for negotiation. However, the student will need information in order to make good choices, and in the giving of that information the lecturer will almost certainly give advice; indeed, he may feel professionally bound to do so. If the advice is contrary to the student's wishes he will almost certainly wish to argue his case; he is unlikely, (for reasons given below), to simply reject the advice out of hand.

5.2.12 The context in which negotiation takes place

Is a student able to negotiate on an equal basis with a lecturer?

The lecturer is an authority figure, in both senses; he has authority, by virtue of his position, and he is an authority by virtue of his superior knowledge, experience and expertise, not just about teaching and the subject to be taught, but also in the field of choosing modules.

The majority of students enrolled in National Certificate courses are in the 16-18 year old group. They have little or no experience of negotiation with an adult. Coming from secondary school, the only similar process in which they have been involved would be the choosing of subjects to be followed in S3/S4 at Ordinary or Standard grade. There the choice would have been much more limited. Many schools, perhaps the majority, use a column structure for course choice; in simple terms, all the available subjects are grouped into seven or eight columns, and the pupil is required to select one subject from each column. The amount of choice is actually more restricted than even that would imply. All pupils are required to take English and Mathematics at some level, most will have been required to take a Science subject. More recently pupils may be required to take a Modern Language. Arriving at a college they may be bewildered by the apparent choice available, they may be over-awed by the size of the institution. They have come from a system which they have known for four years; whether they have enjoyed the experience or not, they have become familiar with it. In college they find themselves in unfamiliar surroundings, with staff they have never met before, in a system they may not fully understand. It may not be insignificant that they come from an institution in which they have had time to establish a place for themselves, to a college in which they suddenly find themselves the youngest and least experienced members. It is unlikely that they will have the confidence or the ability to argue their case with a lecturer whom they see as an expert; the natural tendency will be to leave it to the professional. Whether the lecturer agrees with this opinion of his skill in negotiating modules is irrelevant; it is the student's perception of the lecturer's skills that matters (and teachers are generally good at hiding any lack of confidence from their students).

Obviously, students will vary in the degree of confidence they display, but for many, the tendency will be to simply accept any opinion offered by the lecturer; in effect, while presenting the appearance of negotiating and choosing, they will have allowed the lecturer to make all the decisions. Whether these decisions are in the student's best interests or not will depend on how well the lecturer understands the student's needs.

5.2.13 The quantity of information

From the considerations given earlier it can be seen that the student will need advice and information if he is to make informed, 'good', choices.

> "*....the consumerist commitment assumes that choice is a good thing per se, with the ability to exercise informed choice amenable to solution by recourse to further expertise in the form of guidance.*" (Jonathan 1987, p87)

It was argued in Chapter 3.3.5 that choice is only valuable if it matters which choices are made. If it is important that correct or 'good' choices are made, then choice must be informed. But the quantity of information that must be made available is likely to militate against its being used properly. If a student has to choose between two modules he can look up the (two) Module Summaries and make a decision - if he has to choose between twenty he may decide to make a more hasty decision, rather than face the task of finding and comparing the twenty Summaries. This might be considered a limit on choice - because of the quantity of information to be considered, the student might rationally consider fewer modules than would otherwise be the case.

5.2.14 The motivation of the student to choose

A student may prefer not to choose between modules if

a) he does not understand the benefits of choosing rather than being told what modules to take;
b) he does not understand the criteria on which choice should be based;
c) he does not accept the benefits of choice;
d) he believes he will have a better programme if it is designed by an 'expert';
e) he finds the task of making informed choices too arduous.

If the student decides not to choose then choice has been constrained, although in this case it is a voluntary restriction rather than being imposed by external factors.

5.2.15 Provision of resources

Where a module requires the use of specific physical resources, e.g. computers, laboratory space or a workshop, a limit may have to be placed on the number of students who can be allowed to take the module. Similarly, where there are agreements on maximum class sizes and the maximum working hours of staff some students may be prevented from taking certain modules.

5.3 Minimising constraints on choice

The National Certificate does not allow unrestricted choice; nor was that the intention of the Action Plan. Many of the restraining influences described above were recognised by college staff from the outset (as described in Chapter 3.3). If we assume that the authors of the Action Plan had some knowledge of the further education system, and had given some thought to the consequences of their recommendations, then they must have anticipated these constraints as well. The use of the word 'negotiation', which implies compromise, in conjunction with the word 'choice' indicates that they were not advocating free choice but guided choice. The National Certificate was intended to give more choice; more choice when compared with the previous system.

5.3.1 Of the factors described above, some are inevitable; no design process can escape the constraints imposed by :-

the number of modules available
the number of modules the student can study
the student's educational background
entry requirements
the student's needs

While the catalogue of modules may well increase with time, as it has done since 1985, there will always be a limit to the number of modules offered. Although continuing or recurrent education may increase the total number of modules that a student can study throughout his career, there is a limit to the number he can study in a specified time, e.g. one session, and therefore a limit to the number he can plan to study (4).

While appropriate career guidance may direct his studies more usefully before he approaches the college, his qualifications on arrival at the college are already determined.

As long as articulation within a modular course is desirable, some modules will always have entry requirements.

Students have a variety of reasons for attending college. Unless the course design process is seen as including the possibility of changing the student's needs, the amount of choice is determined by the extent to which his needs can be met by flexible programmes rather than fixed, rigid programmes.

5.3.2 On the other hand, some of the factors given above are functions of the course design process :-

the need for breadth
articulation
core modules
negotiation
the quantity of information

If the need for breadth, articulation and core modules is seen as being of paramount importance, then the amount of choice available is limited accordingly. If they are seen as subject to negotiation the choice is increased. The need to negotiate, i.e. compromise, can be seen as an opportunity to increase choice, according to the number of areas to be negotiated. If, for example, a student can be 'deemed' to have completed an entry requirement to a module because of his experience then more choice is available. If, on the other hand, entry requirements are rigidly enforced and not subject to discussion then an opportunity to increase choice is lost.

The difficulty of making informed choices caused by the quantity of information to be sought, in particular by the time taken to obtain the information, can be reduced by proper organisation of the information. In the context of this work, that is seen to imply the use of computerised data-bases, allowing information on a variety of modules and qualifications to be located and displayed very rapidly.

5.3.3 Finally, the remaining factors are not fixed, but are not subject to change by the choice of an appropriate course design structure. :-

integration
the context in which negotiation takes place
motivation of the student to choose

viability of classes

provision of resources

The drawbacks of integration can be avoided only if the staff who 'design' the integration keep in mind the need to allow choice and avoid integrating too many modules.

The danger of the lecturer making all the important decisions while appearing to negotiate can only be overcome by the training of staff in the negotiation process. This may require attention to staff attitudes to student choice (see Chapter 3.3).

Similarly, the problem that students may not want to choose can be addressed by education, showing them why they will benefit from exercising choice (5).

The viability of classes is linked to the provision of resources. If there was an unlimited supply of lecturers, accommodation and equipment, then a large number of small classes would be possible. Realistically, however, it may only be possible to address the problem of small classes by the use of individualised learning methods, a class consisting of students who may be studying different modules in the same room with the same lecturer, with most of the teaching being done by, say, worksheets, and the lecturer acting more as a tutor answering individual questions rather than addressing the whole class in the traditional sense (6).

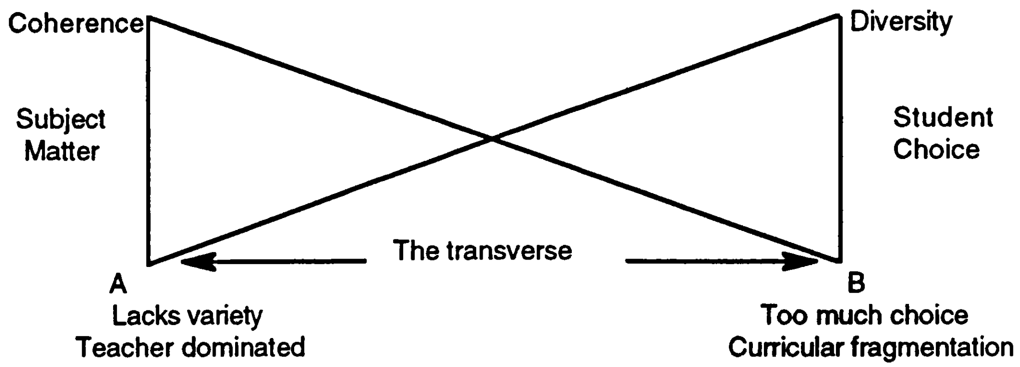
5.4 Summary

A number of factors act to restrict the amount of choice available to students. Some of these are inevitable consequences of adopting a worthwhile modular framework, some can be ameliorated by the adoption of an appropriate course design process, some can be minimised only by appropriate education of staff and students. Most of these factors (if not all) were known to the writers of the Action Plan, who clearly did not anticipate totally free choice as a result of implementing their recommendations.

Warwick, (1987 p18), makes comparisons between the traditional and the modular curricula using a diagram he calls the 'Modular Transverse', as shown below.

Figure 5.4

The Modular Transverse



At A the teacher chooses the material to be studied, the sequence in which it will be studied and the pace. The perceived advantage is that students progressively master the content because each part follows another naturally and leads on to further study, success leading to further success. At B the students have complete choice. This leads to greater diversity in the subject matter and the possibility of teaching being more suited to the specific needs of the individual, whereas at A the course is dominated by the teacher and, Warwick assumes, it will lack variety, i.e. it will not be tailored to suit the student. This need not be the case however. If the teacher is sensitive to the differing needs of students in a class then the material taught, or the way in which it is taught may vary from one student to another. The disadvantage of free choice is seen as curricular fragmentation, no part of the course 'fitting' with any other.

The old style further education courses were at A in the diagram. Given the restrictions on choice discussed above, National Certificate courses are not at B but can, at best, be some distance from A. In Chapter 4.6 it was concluded that courses lie on a spectrum ranging from highly prescriptive (the qualification-centred courses) to highly flexible (the student-centred courses), which is in agreement with Warwick's opinion. Because choice is not free but guided, and limited by the need for breadth and articulation, the fragmentation of courses is less likely to occur. Student-centred courses are never likely to be as coherent as teacher-generated courses; in simplistic terms, if there is one course which is most coherent, then others must be less so, and if each student chooses his own curriculum only one can choose the best. What is required is a balance between all the factors involved. Diversity, to the extent that it meets the needs of individuals, is good, as is a coherent plan to be followed through a course.

5.5 Conclusion

Gatherer gives the following succinct statement of the advantages to be gained from the modular framework.

> *"..... it allows more curricular balance because a variety of studies can more easily be linked; it facilitates the construction of individual curricula, since any student can virtually design a personal curriculum by selecting modules; it allows students more choice;....."* (Gatherer 1989 p119)

These advantages, however, are only made *possible* by the National Certificate - they will not occur automatically; a conscious effort must be made to achieve them.

It is accepted that in some areas of life, choice, in itself, is desirable. In the context of the Action Plan it is the benefit to be gained from allowing choice that is desirable; choice is merely the means to that end. If the advantages to be gained from choice are outweighed by the disadvantages of making misinformed or uninformed choices, then it is perfectly reasonable that choice should be limited; it is the student's interests that are to be served by participation in further education, and if these interests are not best served by allowing choice then it is acceptable that choice is curtailed. The emphasis must be on designing procedures and adopting practices which allow choice within a 'safe' environment, safe in the sense that mistakes cannot be made and that important aspects of the student's education cannot be omitted.

5.6 Notes

1) In the context of the National Certificate, this appears to imply that each module is considered on its own. No such implication is intended in this

work - it is the *course* itself that must have value. See 4.3.3, "The approach can also be criticised because each module is considered in isolation; there is no consideration of the course as a whole." and the description of the evaluation process given in Chapter 6.

2) As modules which increase the breadth of the educational experience tend to come from outside the student's main interest area, while modules which increase articulation come from within the main interest area, there is a contradiction between the two requirements. However, it is not thought that this decreases choice still further, and that is the subject of this chapter.
3) Core modules may be related to those used to increase breadth - in some cases they may be the same modules, e.g. communication skills. They are listed as a separate factor because the *reason* for including these modules is different.
4) A student's plans may extend beyond one session. He may be choosing modules for a two-session course.
5) It must always be accepted that for some students choice is not appropriate; i.e. those who choose a recognised qualification will probably have very few choices to make.
6) "*.... central to the recommended strategies for many modules is an emphasis on student-centred learning and throughout the literature there is a commitment to the use of assessment for formative or diagnostic purposes as well as for certification.*" (Black 1988 p108)
Given that approach, the situation envisaged here may not be as far away as may have been thought.

Chapter 6
Designing a Course by Computer

6.1 Introduction

In Chapter 4 it was concluded that the only satisfactory way of addressing the problems associated with the provision of choice and negotiation is to use a computer. In this chapter the broad outline of such a system will be described. Certain parts of the system were constructed and tested - these parts will be described in more detail in the next chapter.

6.2 The Requirements Definition - What the system must do.

> *"If we cannot precisely state what the system should do, how can we develop the software with any confidence, and how can we hope to check that the end product meets its needs?"* (Bell, 1992, p20)

a) Given a statement of the student's needs, the system must generate lists of relevant modules - possible programmes to be followed by the student.

b) The programmes must also be presented in a form which shows the order in which the modules must be studied, this ordering being such as to comply with the requirement that before a module is studied any prerequisite modules (entry requirements) have already been completed.

b) Each programme must satisfy the student's needs, to some extent.

c) Each programme must be capable of completion within the time available to the student.

d) The programmes should not include modules (or their equivalents) which the student has studied previously.

e) In the first instance, the system will be operated by the student.

f) Where staff are required to assist, this assistance should be directed at improving the programme selected, not at explaining how to operate the system, nor at operating the system for the student.

g) The programmes must be capable of being modified, at a later date, without the need to re-supply information given earlier.

h) There should be an evaluation of each programme, the evaluation to indicate :-
   1) the extent to which the programme satisfies the student's needs;
   2) the degree of breadth provided by the programme;
   3) the degree of articulation between modules within the programme;
   4) the degree of articulation with modules in possible later programmes.

i) The time required to learn how to use the system should be minimised.

j) The time taken to produce evaluated programmes should be minimised.

6.2.1 Notes on the Requirements Definition:

It is written in natural language.

> *"A software requirements definition is an abstract description of the services which the system is expected to provide and the constraints under which the system must operate."*
> *"It should be written in such a way that it is understandable without knowledge of specialized notations."* (Sommerville 1989, p88)

The client (purchaser) of the system will be a college, seeking to increase choice among its students without using too much of staff time. The requirements definition is given in natural language so that the client can see that it meets his needs. On the other hand, the system designer can deduce certain requirements from it :-

The user of the system will be a student - an interface must be designed with students in mind, not staff. The students will be novice, intermittent users.

The time requirements are vague but can be expanded as follows :-

a) Ideally, once the purpose has been explained to the student he should be able to use it without further training, working in response to on-screen messages. He might use it only once a year (indeed, he might only use it once) ; the most that can be expected would be three times, before each teaching block.

b) The time taken to produce programmes is difficult to quantify. The time should be short enough that the student does not feel frustrated or begin to question the value of the exercise. If that is not possible, the program can be broken into, say, four parts. In the first part the student would give details of his present qualifications; in the second he would select a number of modules; the third would fit the modules into the Course Grid to give a number of possible programmes; and the fourth would evaluate the programmes. The time between the second and third use could be utilised by discussing the output with a counsellor.

6.3 The System

A possible system would include the components shown below.

Figure 6.3.1
Components of The System

Data Files

(1)	SCE-CODES	Codes for SCE subjects
(2)	COLLEGE	College policy
(3)	SUPERCLASS	Job names and codes
(4)	JOBS-MODS	Modules related to vocational areas
(5)	QUALIFICATIONS	Modular equivalents of qualifications
(6)	SCE-EQUIV	Equivalents of SCE subjects
(7)	ADVANCED	Modules for entry to Higher education
(8)	MODULES	Module descriptors

Programs

(9)	Module Selector
(10)	Sequencer
(11)	Course Evaluator
(12)	Timetabler

6.4 The function of each component

6.4.1 SCE-CODES

The qualifications possessed by a student on entry to the system have to be identified. Qualifications may be specified in various ways; for example, a student may have some O-grades, some S-grades, some H-grades in SCE examinations, and may have studied a number of SCOTVEC modules while at school. It must be possible to identify these unambiguously for the purposes of the system. Modules will be listed in the MODULES data-base, described later in 6.4.8, but the other possible entry qualifications must be listed in this data-base.

6.4.2 COLLEGE

Some colleges have established policies that all students must include, for example, literacy and numeracy in their programmes. 'Computer Awareness' may be another desired feature in a modular programme. A college may decide that all students should take one or more leisure modules as part of the their programme. Therefore, there has to be a data-base in which the college can express these requirements. The main use of this file, however, would be to indicate which of the modules in the SCOTVEC catalogue is being taught in the college.

6.4.3 SUPERCLASS

The student may refer to this file to identify the code for a job area in which he is interested. Alternatively, the file can be used to produce a printed list of the codes and job areas.

The file is also used to supply the name corresponding to any job code - it is anticipated that internally the system will use the codes without the names. It should be noted that the SUPERCLASS file contains only names of activities or jobs and their references numbers; it does not mention modules.

6.4.4 JOBS-MODS

When a student specifies a job as being of interest, it must be possible to find modules relevant to that job. This file will contain a list of job codes and, for each code, a list of such modules. The alternative to this would be to search the entire file of module descriptors to find those modules that are relevant to the job - a much more time-consuming approach.

6.4.5 QUALIFICATIONS

It has been said that some students still think in terms of the old style courses - or, at least, in terms of the old style qualifications. To allow students to continue choosing modules on this criterion a list of the agreed equivalences will be necessary, to ensure that the correct modules are chosen. In addition, some groups of modules are recognised by certain bodies as having particular relevance. These include more recent qualifications that did not have recognised courses under the old system. These groupings would appear in the 'QUALIFICATIONS' data-base.

6.4.6 SCE-EQUIV

Some modules specify as 'Preferred Entry Level' SCE subjects at various grades. If a student does not possess a relevant O-grade (for example), it must be possible, within the modular system, to allow the student to study this subject. This may be done by including O- or S-grade subjects within the college, or, more appropriately to a modular system, modules equivalent to these subjects must be identified. There should therefore be a list of equivalences between SCE subjects at various grades, and modules.

This is not the most frequent use of the term 'equivalence', and it may be argued that these lists should appear in the previous component (Qualifications). However, there is a difference in the use made of the two files, and as this will probably mean a difference in the way the file is accessed, it was thought better to keep them separate. The Qualifications list is normally used in only one direction; given the name of the qualification the list of modules which the student will have to study can be presented. The SCE-EQUIV file may be used in the same way, the student stating that he wants the equivalent of, say an O-grade in English. The file can then provide the list of modules required. It will be used more often in the opposite direction; if the student states that he has already passed O-grade English, the file can provide the list of modules which he is already deemed to have completed, the modules he would <u>not</u> need to include in his course.
It is, of course, possible that the Qualifications lists can also be used in this way. If an adult returns to college after previously gaining a qualification he would want to enter the relevant modules as having already been taken. The quickest way to do this would be to specify the name, or code, of the qualification already gained and let the computer look up the modules and enter them along with any SCE subjects.

6.4.7 ADVANCED

National Certificate programmes must, where possible, articulate with more advanced educational courses. Where a student states that his intention in taking SCOTVEC modules is to secure the entry requirements to higher education, lists of the higher education requirements must be available.

6.4.8 MODULES

In order to select modules appropriate to a student's needs, there has to be a data-base containing 'descriptions' of the modules. This will have to contain the information about the modules, required by other parts of the system.

For example;

a) a 'candidate' programme of modules has to be checked to ensure that it can be completed within the time available to the student. The module lengths will therefore have to be included in the information about the modules.

b) At some point in the process, the student will have to personally choose between two modules which are very similar, and cannot be distinguished by the computer. For the student to make this choice he will have to be given textual description of the module in language he can understand (the Module Summary). This data-base is the most appropriate place to store such descriptions.

c) In addition, a more complete description might be required by lecturers to assist in planning their work. While this is not directly relevant to the task of a student selecting modules (with computer assistance), if such descriptions are to be available this would be the place to store them.

It is possible that, in a working system this file would be split into three sub-components. One would contain the 'simple' data, for example, the length of a module or its list of entry requirements. The second would contain the 'plain language' information about the module, for the use of students. The third would contain the full Module Descriptor (1). The reason for splitting the file in this way is mainly concerned with speed. If all the user wants is the length of the module, he does not want to wait while the computer reads the whole record including a completely irrelevant five page description of the module. However, the way in which the file is to be read, and the characteristics of the hardware used will determine the ultimate choice of file structure. If the computer operates very quickly, or is programmed so that only the necessary information is read, there would be no real need to separate the files as far as the use of the data is concerned. There may still be advantages in having separate files, for the purpose of editing and maintaining the data. This decision will depend on the way the editing is to be done, and by whom.

6.4.9 All of the components described above, 6.2.1 to 6.2.8, are essentially passive data-bases. They will be 'active' only in the sense that they can be edited, but they are basically blocks of relevant information. Each will have its own 'manager' or editer program to allow it to be updated and modified as required. These editors cannot be independent, as alterations to one file may require alterations to others. More will be said of this in Chapter 9.

6.5 The situation at present

The files to be used by the 'integrated computer system' have been described, so far, at an 'ideal' level; we have stated what we would look for in such files.

We will now consider the extent to which these components are actually available. It will be seen that in most cases some kind of file similar to that required already exists. In some of these, however, it is not clear whether the lists that are published are the output from a word-processor or a data-base.

6.5.1 SCE-CODES

The SEB identifies every subject that it examines by a code number e.g. Physics is 3220. Grades are also distinguished e.g. Physics, Standard Grade, General = 3220/285, Credit = 3220/286. These numbers are also used to identify different parts of an examination e.g. Physics Higher Grade, Paper I = 3220/468, Paper II = 3220/469. The code numbers are not published, being intended only for the use of the SEB.

6.5.2 COLLEGE

The only reference to such a file is that mentioned in the CAST Computerised Guidance System, which is described in Appendix 5. This file was to contain details of the classes offered by the college, and it was to be maintained by the college. This seems to be appropriate.

6.5.3 SUPERCLASS

The original file is maintained by the Department of Employment. For the purposes of this system a sub-set of that file may be more useful, the sub-set consisting of those job areas for which modules are available. This will be discussed in more detail later.

6.5.4 JOBS-MODS

In describing the CAST Computerised Guidance System (see Appendix 5), reference was made to its ability to identify relevant modules, given the code for a job. The data-base which they have developed is similar to that required here.

6.5.5 QUALIFICATIONS

As explained in Chapter 2, various professional bodies specify groups of modules that satisfy their requirements. An example was given in Table 2.9.1. Details of these agreed equivalences are published by SCOTVEC. This is similar to the file required here.

6.5.6 SCE-EQUIV

In a similar way to that adopted for QUALIFICATIONS, modular equivalents of SCE examination subjects have been decided by some organisations. An example is given in Appendix 6.

If a module requires knowledge of basic electricity, for example, SCE H-grade Physics could be specified as a pre-requisite, as this subject contains a large proportion of electricity. If the student did not possess a pass in H-grade Physics he would be required to study the modules considered equivalent - but in this case only the electricity content of the subject is relevant. In this kind of case, where only part of the SCE subject is actually considered necessary, the entry requirement in the module is normally specified as "H-grade Physics or 69036 Physics 2 : Electricity and Electromagnetism".

6.5.7 ADVANCED

This file, as indicated above, contains groups of modules recognised by colleges and universities as entry requirements to courses more advanced than the National Certificate.

6.5.8 MODULES

The basis of this data-base is already maintained by SCOTVEC. At present it contains, essentially, only the information found in the Module Descriptors. 'Plain language' versions of these descriptors, the Module Summaries, have now been made available and are included in the SCOTVEC data-base. The Module Descriptors are intended for the use of lecturers; as students will have to be informed about the modules, the additional file of 'plain language' descriptors is also required.

6.6 The programs

The data-files have now been described; they will each have their own 'manager' program to allow them to be up-dated and edited, lists to be printed etc. In addition, to link these files together for the purpose of facilitating choice and negotiation, specific programs will be needed; these are the four components from Figure 6.3.1 which have not yet been described.

6.6.1 The Module Selector

This component has to select, from all the modules available, those that are relevant to the student's needs. This process will include the identification of;

a) all modules directly relevant to the student's stated needs;

b) any modules required as 'entry levels' into modules identified in (a), where these 'entry' modules have not already been identified in (a).

The Module Selector has to elicit the student's needs, and specify them in a form comprehensible to the rest of the computer program. The Module Selector will use the information in the data-bases described above to make its selections.

6.6.2 Sequencer

The sequencer has three functions;

a) It has to ensure that modules are arranged so that any entry requirements come before the modules to which they refer.

b) It must ensure that the programme is capable of being completed in the time available to the student. For example, where a sequence of modules is identified for possible inclusion in the student's course the Sequencer must check that the total time required for the sequence is not greater than the length of the session (or sessions if the student intends to be at college for more than one year).

c) The sequencer must also ensure that the modules, as a complete course, can be fitted into the Course Grid. The Grid contains forty-eight cells, each equivalent to a half-module. If the modules identified by the Selector have a total length greater than forty-eight, they cannot all be fitted into the Grid. The Sequencer must select from the identified list groups which together have a length not greater than forty-eight. These groups cannot be selected at random; if a module has entry requirements these must be in the group as well.

6.6.3 Programme Evaluator

The purpose of the Evaluator is to give a value to each of the groups of modules selected by the Sequencer programme. The course with the highest value would then be selected as the most appropriate for the student. In evaluating a course it is not sufficient to simply look at the value

of each module. A good course has certain characteristics; for example, it should allow the student to follow some lines of study to deeper levels. These are the sequences described above. Three modules, as a sequence, should be given a greater value than three unrelated modules.
The basic principle of such a scoring system is that it should give higher scores to features of the course that are considered desirable, and lower scores to features that are less desirable. Anything that does not contribute to satisfying the student's needs can be given a zero score, while undesirable features, to be discouraged, can be given negative scores.
To be pedantic, 'encouraging a desirable feature' means assigning it a value in such a way that the probability of it being included in the final course is increased.
At one stage, the use of probabilities was considered for the Evaluator; for example, a desirable feature would be given a high probability of being in the course. Although this approach had attractions, it seemed unnecessarily complicated because of the interactions between modules when probabilities are involved. In brief, one of the factors determining a probability is the amount of space remaining in the Course Grid; if there is no space left, any module not yet included has no chance of being included now. This meant that as the Course Grid was being filled, probabilities had to be continuously re-calculated.

In designing a scoring system for modular courses the following principles should be considered;

a) Modules forming sequences, i.e. modules that are entry requirements to others, are desirable. These imply progression to deeper levels of study.
b) To encourage breadth, modules should not all be taken from the same part of the catalogue. The college may decide that it is desirable that all students have some form of relaxation as part of their course. This requires that modules related to leisure, recreation or hobbies should be considered desirable, and encouraged by the scoring system.
c) Complementary modules, those that develop the same skills in a different context, or different skills in the same context, are desirable.

d) The college may decide that all students should have some study time, time when they are not attending classes and can use the library, for example. A certain number of empty spaces in the Course Grid would therefore be encouraged.

If the features described above are considered desirable, then their 'opposites' are presumably undesirable. If, however, the desirable features are encouraged by the scoring system the undesirable features will automatically be discouraged because they will not score points. On the other hand, some of the factors given above are, to some extent, contradictory. For example, modules that form sequences are desirable; by implication, independent modules, (those that do not form sequences) are undesirable - but these are likely to contribute to breadth or recreation. This is why a variety of 'good' programmes is possible. Similarly, the scoring system may encourage the inclusion of study periods, but these must be distinguished from 'empty space' in the Course Grid; in other words, only a certain number and distribution of study periods should be encouraged.

6.6.4 The Timetabler

As stated earlier, the Timetabler is not within the scope of this project. While not attempting to describe how one can be constructed, it is appropriate to consider its place within the system being described.

The modules selected by (or for) students have to be timetabled so that qualified staff are available to teach them, in suitable accommodation, with relevant equipment. Student numbers have to be considered to ensure that classes are large enough to be viable. It is possible to envisage a timetabling program that is integrated with the other elements to the extent that timetabling is performed (and probably amended at frequent intervals) as the students use the system to design their courses. However, identifying modules that cannot be run because of insufficient numbers can only be done when all students, or a sufficiently large sample of them, have been considered. It is therefore likely that the timetabling will be performed as a separate stage, when all module selections have been completed. On the other hand timetabling earlier in the process, by indicating the most and least attractive modules and the number of students wishing to enrol for them, would allow initial planning of resources to begin.

6.7 Interdependence of the programs

These four components, 6.6.1 to 6.6.4, will be mutually dependent, or at least mutually inter-active. It is unlikely that the best course will result from a linear application of the steps

(a) select modules,

(b) arrange them in sequence,

(c) choose the best sequence,

(d) timetable the programme.

For example;

a) It is possible that the student will have stated his needs too narrowly, in which case insufficient modules will be identified to constitute a worthwhile programme. When that is discovered, by the Module Selector, the student must be re-questioned, asking him to widen the scope of his requirements.

   Alternatively, he may have given a very general description of his needs, in which case too many modules can be found. Rather than spend time trying to evaluate a large number of virtually indistinguishable programmes (indistinguishable in terms of the extent to which they satisfy his needs), it would be more useful to ask the student to be more specific if he can be.

   In either case, the Selector program will be used repeatedly, adding and deleting modules as required, until a reasonable number of modules has been identified; they can then be passed to the Sequencer.

b) Some modules may be selected by very few students. The college may feel that such a small class is not viable, in which case the students affected would be required to reselect modules to complete their programmes. Ideally this should be identified as early as possible so that the minimum number of students is asked to re-select modules; it should also be late enough that a reasonable sample of the students has responded, so that the decision to abandon the module is based on valid evidence that numbers will be too small.

c) Depending on the resources (of staff, accommodation and equipment) the timetabler may find that two modules 'clash', have to be taught at the same time by the same staff or using the same equipment, and one must be withdrawn. Again, students affected by this decision must re-select modules.

Both (b) and (c) require that information is fed back from the Timetabler to the Selector. Where it is found that a module is not possible, because of either of these reasons, the module should be removed from those offered; this would be done by altering the MODULES file, flagging the relevant modules as not available in the college. This may be considered a temporary alteration to information in the COLLEGE file, if that file is used to indicate the available modules.

From these examples it can be seen that the modules in a student's course may have to be changed at an early stage in the design of the course, e.g. the total time required by the modules does not constitute a worthwhile programme, or at a much later stage, e.g. when the modules are being timetabled.

6.8 Presenting Information

Before concluding this section and proceeding to a more detailed description of the components, some thought should be given to the way information should be presented to the student, when he is required to respond.

Two possibilities can be envisaged;

6.8.1 On-screen display

All information to be presented to the student is presented on the computer screen. For example, lists of SUPERCLASS job names can be shown on the screen, the student indicating his selection by 'pointing' at it in some way;

a) A list of, say, ten items may be shown, each item being numbered; the student types the number of the item he wishes to select.
b) The student may be able to use the keyboard to move up or down the list until the correct item is highlighted; a 'select' key is then used to indicate that the correct item is highlighted.
c) The student may use a 'mouse' to indicate the appropriate item, and a switch on the mouse to register the selection.

6.8.2 Printed lists

Some lists may be provided on paper. Before approaching the computer the student chooses items from these lists. He might fill in a questionnaire as he selects, the answers to the questions being the code numbers of the selected items. The computer then asks the same questions as appeared on the questionnaire and the student types in the appropriate codes.

6.8.3 Both methods have advantages and disadvantages.

a) The 'all-computer' method, 6.8.1, has the usual advantage of speed in that selected lists can be found and displayed quickly, whereas, using printed lists, the student has to find the appropriate section from all those offered. The Equivalences list published by SCOTVEC (SCOTVEC 1986), runs to 160 pages. Even using the index it is time-consuming to search for a number of qualifications to see what they involve and to choose between them.
b) The computer can present lists in a hierarchical fashion, starting from general statements and, on the basis of the student's responses, proceeding to more specific descriptions. With some of the lists used in this system, however, there may be difficulties :-

   "Several authors have urged four to eight items per menu, but at the same time they urge no more than three to four levels. With large menu applications one or both of these guidelines must be compromised."

   (Shneiderman 1987 p98)

For example, the most logical way of dividing the SUPERCLASS list for presentation on screen would have the seventeen main classes in the first screen, and up to nine items in each of five subsequent screens. A more attractive alternative method would be to use the screen as a window, showing part of a 'map' of the SUPERCLASS hierarchy, although how much of the map could be shown at any one time would be limited by the screen display and the size of each job name and number. A suggested display method is explained in Appendix 11.
A combination of hard copy and on-screen display might be the best solution, as suggested by Shneiderman, discussing printed 'menu maps' to help users work their way through complex on-screen displays. (Shneiderman 1987 p100)

c) With the on-screen display the student is forced to follow the procedure dictated by the computer. With a printed list, the student might try the disorganised and inefficient approach of flicking through the pages trying to spot the item he requires. On the other hand, it is difficult to 'browse' in the computer unless it is designed to allow that.

d) The on-screen approach requires considerably more programming to show the lists properly on screen and to select the correct list.

e) Where lists change frequently, the on-screen lists, being read directly from the files, are always up to date, whereas old versions of the printed lists must be destroyed and new copies printed.

f) If all the operations, including the student's decisions, are performed at the computer, then each student will occupy the machine for a longer time; it may be difficult to find time for all students to have access.

g) The accuracy of the response depends on the way the student has to indicate his choice. When the student has chosen an item, can he tell the computer what the item is without making mistakes? Typing the code number of a job is likely to result in fewer mistakes than typing the name; highlighting it or pointing at it are even more reliable. Asking the student to select from a menu appears to be the most appropriate method for avoiding typing errors :-

"This simplified interaction style reduces the possibility of keying errors

and structures the task to guide the novice and intermittent user."
(Shneiderman 1987 p86)

and

"Menu selection is especially effective when users have little training, are intermittent in using the system, are unfamiliar with the terminology, and need help in structuring their decision-making process." (ibid.)

h) Some students will find it more difficult to scroll through a number of lists on screen, compared with turning a page in a printed list. For example, Shneiderman quotes a number of studies which have shown that hard copy is quicker to read than an on-screen display.(2) (ibid. p360)

These considerations must be taken into account when choosing the method to be adopted. It is felt that the on-screen approach is preferable, . but the system, at an interim stage in its development, could use both methods (but see below).

6.8.4 Presentation as lists.

One more factor should be considered; are all the data-files equally suitable for presentation as printed lists?

6.8.4.1 SCE-CODES

This would be a relatively short list, containing approximately ninety subjects, although it would have to be presented in the form of a matrix, as shown in figure 6.8.4.1 below.

Figure 6.8.4.1

A sample from the SCE-CODES list

	O-gd	S-gd	H-gd
Accounting	001	-	002
Agricultural Science	003	-	004
Biology	005	006	007
Chemistry	008	009	010
Science	-	011	-
Technological Studies	-	012	013

The student would find the subject name, read across to the appropriate grade, and note the code number for the subject/grade. This is not a particularly time-consuming task, except when a student has a large number of subjects to find. Finding and writing down the correct code could be prone to error, particularly if it is felt necessary to specify codes for all the different grades that can be awarded in a Standard grade course (there are seven possible grades).

6.8.4.2 SUPERCLASS

The full SUPERCLASS directory contains about seventy pages; the index occupies an additional sixteen. Finding a specified item in it requires some knowledge as to its organisation. This list is better handled by the computer, which can lead the student through the various levels until the desired degree of specificity is reached. Comments regarding the complexity of the directory have already been made, in 6.8.3 (b), above.

6.8.4.3 QUALIFICATIONS

The lists of modules equivalent to certain qualifications have always been published as printed lists in the past. For the purposes of selection, the list need not include the module lists. It need only be a list of qualifications and an appropriate code number. This list could be usefully offered in printed form. However, the data-file relating the qualification code to the list of equivalent modules must exist.

6.8.4.4 SCE-EQUIV

The modular equivalents of SCE subjects are mainly used internally by the computer in considering the student's existing qualifications. It might be useful to offer this list to students wishing to study the equivalent of an SCE subject, but this would only be for information.

6.8.4.5 ADVANCED

The comments for 6.9.3 apply equally to this list.

6.8.4.6 MODULES

A list giving all the information about all the modules would be extremely large. Each of the seventeen categories could easily fill a lever-arch file, if not several. The only information relevant to the student, however, are the Module Summaries; these normally occupy a single A4 page. But there are nearly three thousand modules now available! The computer is by far the best place for these lists to be manipulated. (It may be remembered that the quantity of information relating to modules was one of the main reasons for considering the use of computers :- see Chapter 4.)

Some of the lists are not used by the student :-

6.8.4.7 JOBS-MODS

This list relates job codes to module reference numbers, and is used internally by the computer. The student need not know it even exists. (In the small prototype system to be described later, the list was modified and used as a substitute for the SUPERCLASS file - this had advantages during the development of the program. However, being organised in the same way as the SUPERCLASS file, and being almost as big, it suffers the same disadvantages.)

6.8.4.8 COLLEGE

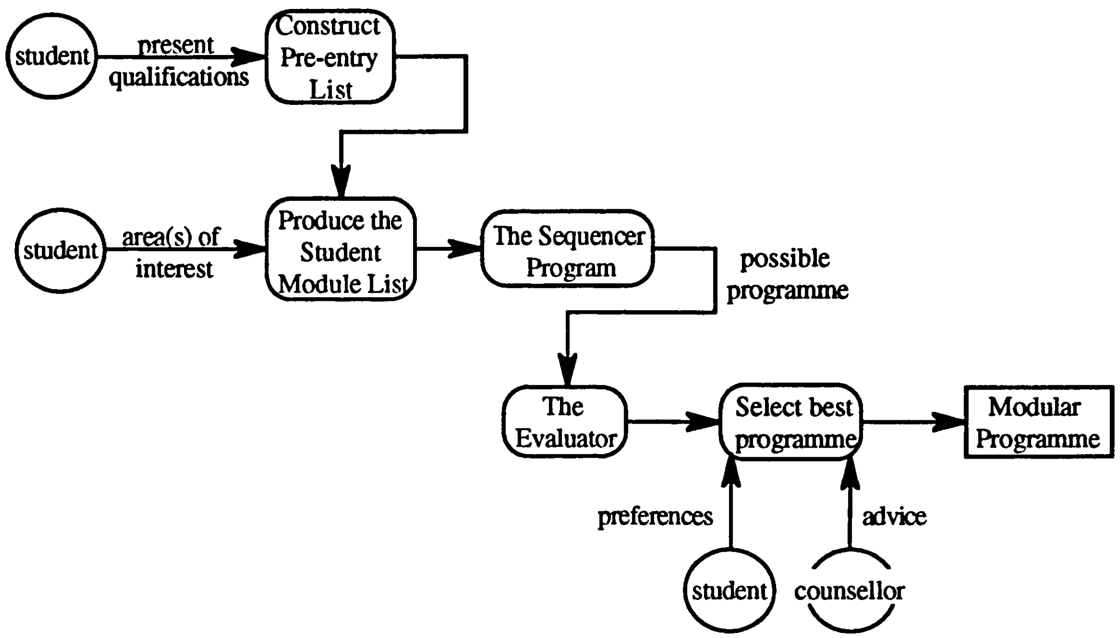
This file contains statements of college policy, and indicates the modules available at the college. As such, it does not require any response from the student. The policies might appear in printed form, for information, but the large list of modules might be rather daunting. The main use to be made of this module list is to control the modules that can be added to a student's course. As this is done automatically, there is no need for the list of modules to be presented to each student.

6.9 The Program Descriptions

Figure 6.9 shows the overall flow of data through the system.

Figure 6.9

Data Flow Chart of the System



As this is the first time such a chart has been used in this work, the following comments are given for the benefit of the non-specialist.

Various conventions are used in data flow charts. (3) In this work the following conventions are used :-

Round edged rectangles indicate 'transformations' or sub-programs, where data is changed in some way(A).

Rectangles represent a data store. These are usually files on disc.

Circles represent user interactions with the system.

Arrows show the direction of data flow.

Keywords **AND** and **OR.** AND OR

An arc symbol linking data flows to show precedence of **or** over **and**.

For clarity, the data input or output is only named (by labelling the appropriate arrow) where this is not obvious from the context. For example, the output from the "Construct Pre-entry List" sub-program is the Pre-entry List.

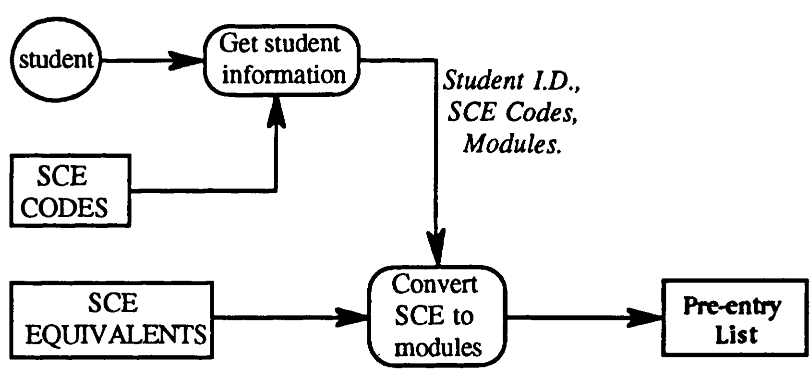
Figure 6.9 does not show all the data inputs to the system - this would complicate the diagram to the extent of making it unreadable. Each part shown in the figure is described in more detail below, where the data inputs are shown.

6.10 The Selector Program

The tasks identified in Chapter 4 form the basis for the Selector program. The initial task is to construct the Pre-entry List, i.e. the set of modules, SCE subjects and their modular equivalents which the student has before starting at college. The data flow diagram is shown below in Figure 6.10.

Figure 6.10

Construct the Pre-entry List.



6.10.1 Get student information.

The information supplied by the student will include :-

a) an identifier, e.g. name and matriculation number, so that any data resulting from the program can be associated with that student in future;
b) the reference numbers of any modules the student has already completed;
c) the names and grades of any SCE subjects taken by the student. These are converted into codes, using the SCE Codes data-file.

6.10.2 Convert SCE to modules.

Look up the modular equivalents of the SCE subjects.

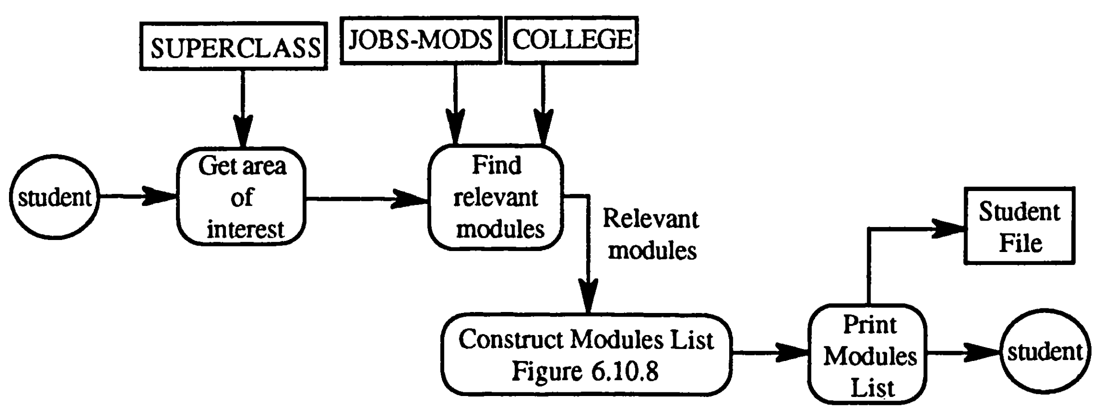
Add these to the modules listed in (b) above.

6.10.3 Constructing the Student File.

The Student File contains all the necessary information about the student, and the list of recommended modules (the Module File).

Figure 6.10.3

Produce the Student Module File.



6.10.4 Get area of interest.

The student identifies the area of interest to him. This can be done by presenting the student with the SUPERCLASS hierarchy (or map) and have him work through it to the required depth, finally identifying the name of the area. The output from this sub-program is the SUPERCLASS code for the area.

6.10.5 Find relevant modules.

If the college has policy that certain modules are to be taken by all students, these modules will be listed in the COLLEGE file and form the start of the Module List. Given a SUPERCLASS code, the JOBS-MODS file returns a list of relevant module reference numbers. This forms the beginning of the Modules List.

6.10.6 Construct Modules List.

The modules already identified are checked for entry requirements (which are themselves checked) and the student is given the opportunity to delete modules, until he is satisfied with the modules left in the list. This sub-program is described in more detail below.

6.10.7 Print Modules List.

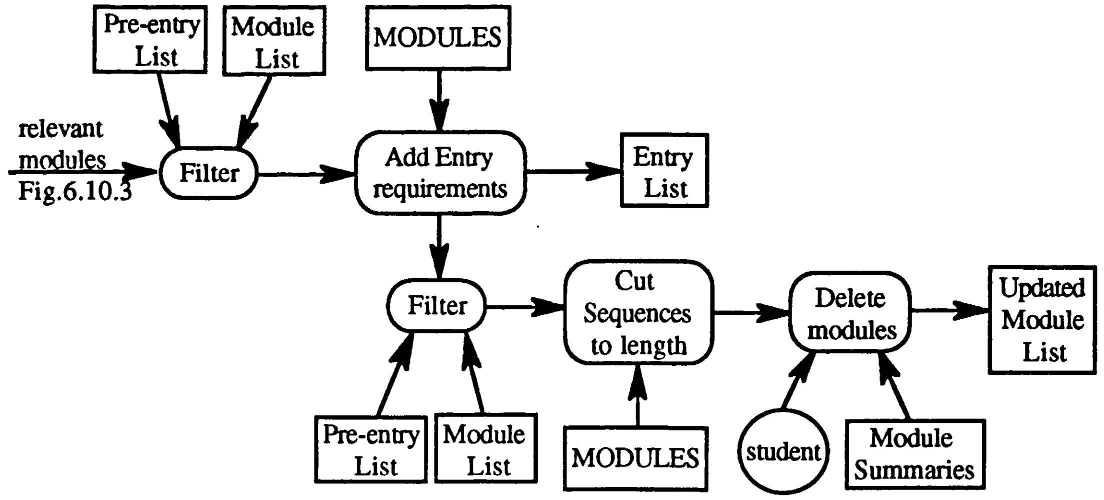
At this stage the Modules List can be printed for the student. This is useful if the student wants to stop using the system at this stage, perhaps to seek further advice or information. The Modules List is also saved on disc for future reference.

6.10.8 Constructing the Modules List

The detail of this sub-program is given below in Figure 6.10.8

Figure 6.10.8

Construct Modules List



6.10.9 Filter

Modules being considered for inclusion in the Module List are compared with the Pre-entry and Module Lists, and are discarded if present in either.

6.10.10 Add Entry Requirements

Every module added to the Module List must be checked for entry requirements, and these must be added. The implementation of this sub-program must allow for the possibility that an entry requirement may itself have entry requirements. One possible method will be described in Chapter 7. Whatever the method used, a record must be kept of the 'sequences' resulting from entry requirements; this could be in the form of the diagrams described in Chapter 4.3.1 or, probably more suitable for computer implementation, an 'Entry List' as shown as one of the products of the 'Add Entry Requirements' sub-program. The Entry List to be described in Chapter 7 may be visualised as shown in Figure 6.10.10.

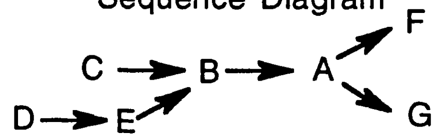
Figure 6.10.10

Comparing Entry List with Sequence Diagram

Entry List

1	2
A	F
B	A
D	E
C	B
E	B
A	G

Sequence Diagram



The list is read as 'the module in column 1 is an entry requirement to the module in column 2'. The reader may verify that the Entry List and the Sequence Diagram given in Figure 6.10.10 convey the same information. Data flow diagrams should state what must be done, rather than how it should be done :-

> *"One of the principal advantages of data-flow diagrams is that they show transformations without making assumptions about how these transformations are implemented."* (Sommerville 1989, p 235)

However, as more detail is added to the data flow diagram, **what** must be done in one process becomes dependant on **how** another process is to be carried out.

6.10.11 Cut Sequences to Length

Some of the sequences in the Module List may be too long to be completed in the time available. For example, the sequence A ➤B➤C➤ D would be too long if each module was the standard length. Module D must, therefore, be deleted from the List. The length of each module must be known in order that these decisions can be made; this information comes from the Modules file.

It is arguable whether this sub-program should be included as part of the process of constructing the Module List or as part of the process of fitting modules into the Course Grid (logically, it would appear to belong to the latter). If it is used each time modules are added to the Module List then the student knows as early as possible that some modules cannot be added, and, more importantly, that there is still space available in the Module List. In that case, however, the sub-program is used repeatedly and slows down the process of constructing the Module List. If it is placed at the start of the sub-program to fit modules into the Course Grid it is invoked only once each time that sub-program is used. The program will therefore run more quickly, but the student may then have to look for additional modules to complete his programme.

6.10.12 Delete Modules.

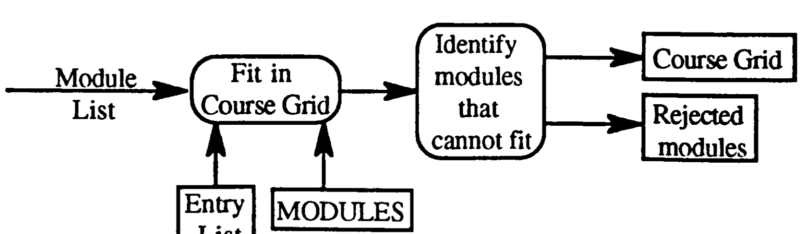
The student is given the opportunity to delete modules he does not wish to study. This sub-program must point out the consequences of deleting a module which is an entry requirement to another - hence the need to define an Entry List, in some form, in the 'Add Entry Requirements' sub-program.

6.11 The Sequencer

The Sequencer can be represented as shown in Figure 6.11.

Figure 6.11

The Sequencer Program



6.11.1 Fit in Course Grid

Modules have to be fitted in the correct sequence, i.e. entry requirements fitted before the module to which they relate, and the length of each module must be known - hence the references to the Entry List and Modules file. The process of fitting modules into the Grid and identifying those that cannot be fitted is very much concerned with the way in which the sub-program is implemented. This sub-program was prototyped to establish one possible method, and this will be described in Chapter 7.

6.11.2 Identify modules that cannot fit

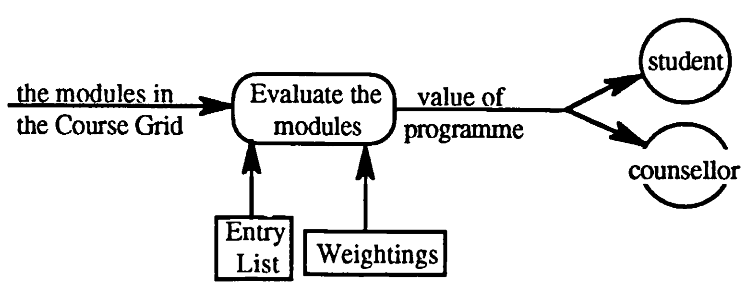
It can be argued that this is not a separate sub-program, and is simply a 'side effect' of fitting the modules into the Grid. However, it is important that modules are not simply omitted from the Grid, but identified so that the student can decide whether he wants to delete them or not. For example, he may decide that certain other modules are less desirable, and by removing them he can create space for those previously omitted.

6.12 The Evaluator

Most of this sub-programme is concerned with simple arithmetic. The input is the set of modules being evaluated. As value is assigned to modules which form part of sequences data is required from the Entry List, and, unless the values to be assigned to the various criteria can be fixed and written into the program, a file giving the weighting of each criteria is required.

Figure 6.12

Evaluating a programme



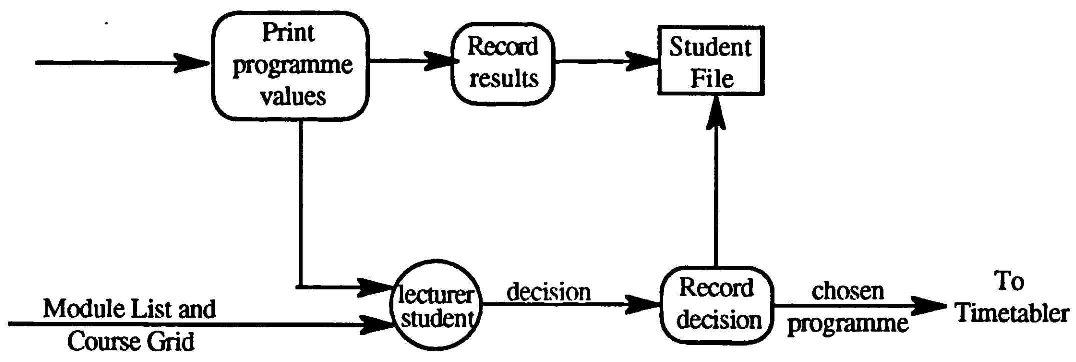
6.12.1 Selecting the best programme

While the computer can assign value to courses according to identified criteria, and can therefore indicate the course with the highest score, it is felt that the decision as to the final course to be selected should be in the hands of the student and counsellor. The relative importance of the criteria may vary from one student to another - in some cases more progression than is usual may be considered good, in other cases it may be felt that breadth should be emphasised. In any case, if the student is to feel 'in control' of his programme, i.e. if he is to feel the 'sense of ownership' that was to be one of the benefits of choice, he must take the final step.

To facilitate this process the computer will print the alternative programmes, with the values assigned to each of the criteria, and then await a decision. The results should be recorded so that the system can shut down while the decision is being made. Once the decision has been made it will be recorded and passed to the Timetabler.

Figure 6.12.1

Selecting the best programme



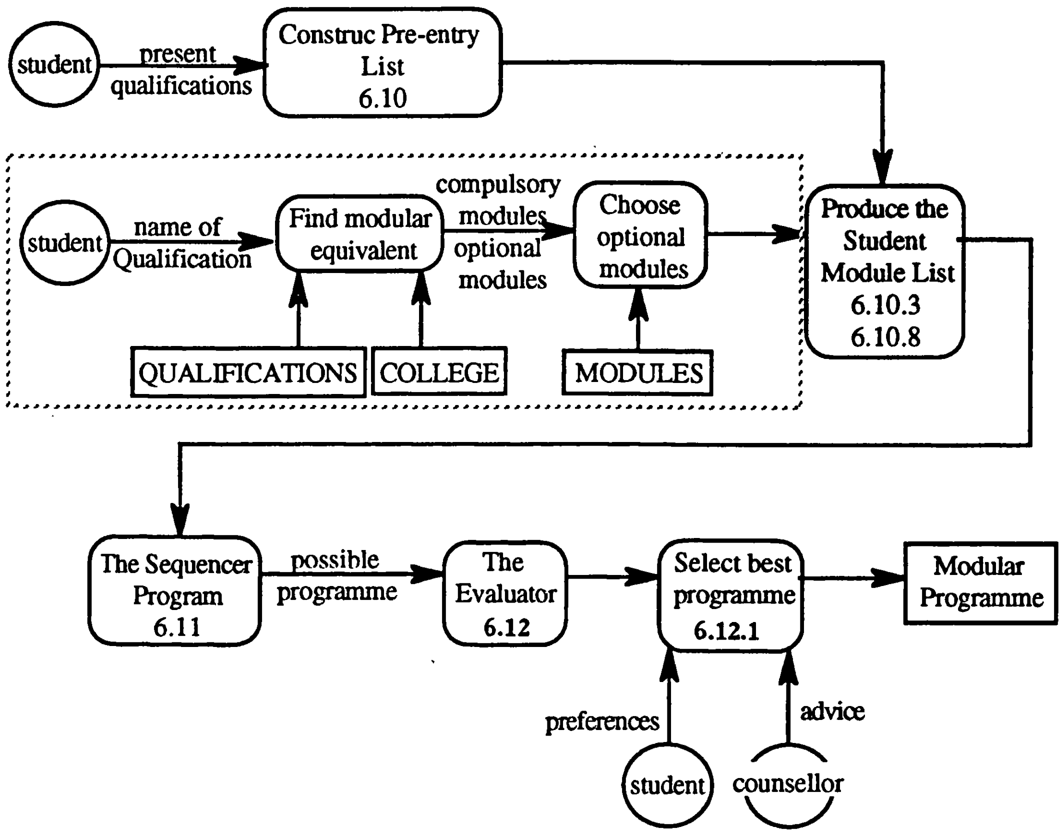
The descriptions given up to this point have been concerned with the operation of the most general case, the student-centred design process. The more restricted cases, the qualification-centred and higher education-centred cases will now be considered.

6.13 Selecting modules to gain a recognised qualification

Figure 6.13 shows the data flow through the system when the student wishes to study modules leading to a recognised qualification.

Figure 6.13

Gaining a Recognised Qualification



Comparing this figure with Figure 6.9 it will be seen that much of the system is identical. Only the area within the dotted box is different. As the other parts of the system have been described already (in the figures indicated under the processes), only the additional part will be described here.

The student names the qualification in which he is interested, by selecting from the list contained in the QUALIFICATIONS file. He might do this by reference to a printed list or to an on-screen display (see Chapter 6.9.3).

6.13.1 Find Modular Equivalent

The computer finds the relevant modules by looking at the list given along with the name of the qualification in the QUALIFICATIONS file. The student is informed of any compulsory modules; at this stage any modules demanded by the college are also added, from the COLLEGE file.

6.13.2 Choose optional modules

The student is presented with the optional modules and asked to choose according to the instructions associated with the qualification; for example, "Choose two from this list". To assist these decisions the student may ask to see the Module Summary for any of those offered; these are taken from the MODULES file.

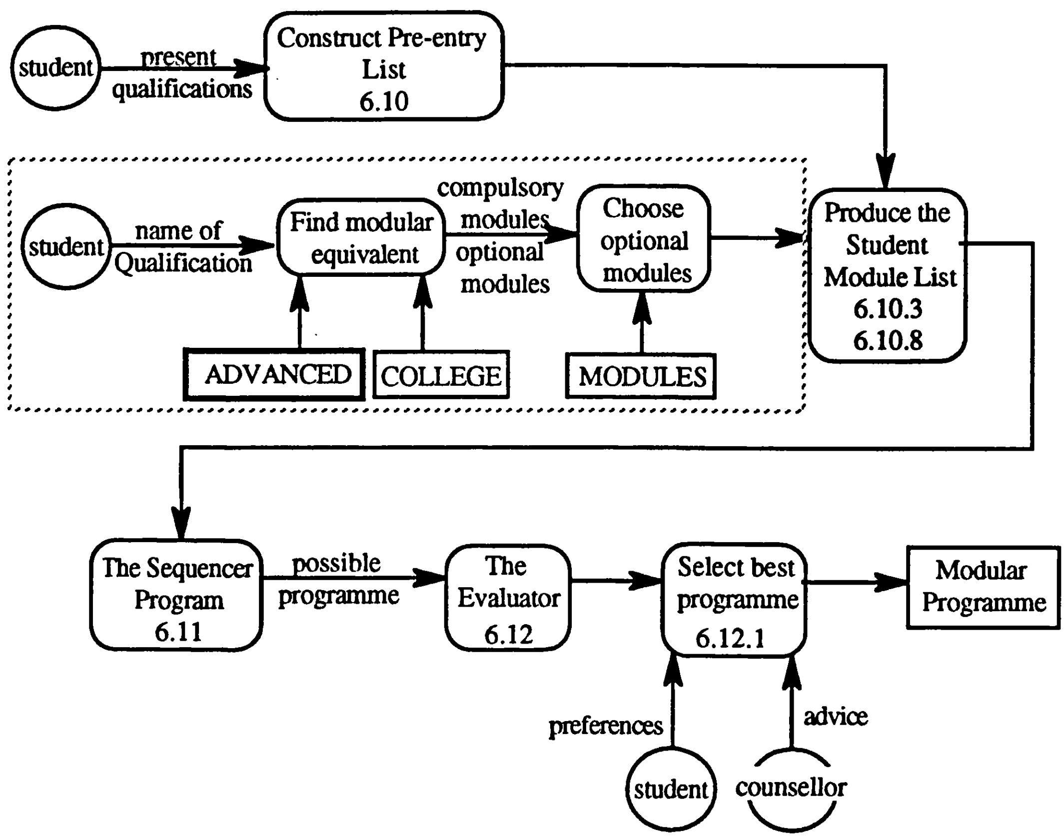
The only differences in the remainder of the system are due to the fact that the compulsory modules cannot be deleted from the programme. Hence, in the sub-program, "Produce the Student Module List", the compulsory modules are flagged and any attempt to delete them is prevented - if the student is allowed the option of deleting them the programme will fail in its basic objective of listing the modules required to gain the qualification. Because Equivalences tend to specify a complete programme, (see the comments in Chapter 2.7, concerning the lack of flexibility in these courses), it is likely that the required number of modules has been found. However, if the student already had relevant module passes, as identified in the pre-entry list, there may be some space available.
Because all programmes aiming to satisfy the needs of a particular qualification will be very similar, it is doubtful whether the Evaluator will reveal any significant difference between them (see Chapter 4.4.3)

6.14 Selecting modules to gain entry to more advanced courses

Comparison of Figures 6.13 and 6.14 (below) shows that the only difference is in the file used to supply the list of compulsory and optional modules (shown highlighted in Figure 6.14). The same remarks concerning the need to avoid deleting the compulsory modules apply.

Figure 6.14

Gaining Entry to Advanced Education



The student names the course to which he seeks the entrance requirements, by selecting from the list contained in the ADVANCED file. (Once again, the list may be presented on paper.)

The computer will now find the list of modules required, from the ADVANCED file.

6.15 Choice and Negotiation

The computer system described here was intended to increase the amount of choice available to students. To what extent does it do that?

Chapter 5.17.1 listed five factors which limited choice, but which were subject to control during the course design process.

These were breadth, articulation, core modules, negotiation and the quantity of information.

Of these, two, "articulation" and "the quantity of information", are directly addressed by the system described.

Articulation to more advanced education is explicitly described as one of the proposed functions of the system (see 6.14, above). Articulation within the course is facilitated in two ways;

a) the weighting assigned to 'sequences' can to be used to increase the probability that series of modules leading from basic to more advanced skills are included in the programme;

b) the use of the SUPERCLASS system as the 'entry point' to the less specific programmes (the student -centred design process) ensures that all modules related to a job area are identified (4). This will include the more advanced modules.

The problems associated with the quantity of information that has to be processed to produce a viable course are solved simply because of the speed with which the computer can find the information.

Consider the following process :-

> Given a SUPERCLASS job code, find the reference number of a relevant module.
>
> Look up the Module Summary and note the entry requirements to the module.
>
> Check each of the entry requirements to see if they have entry requirements.
>
> Repeat this until no more entry requirements are found.
>
> For each of the modules identified, list the module reference numbers, titles and lengths.

The simple prototype to be described in Chapter 7 can carry out this process in seconds, rather than minutes. When the process is to be repeated for a number of job codes, a series of actions which is simply not practicable when performed manually becomes a workable proposition.

The problem of ensuring breadth is not overcome automatically by the proposed system, but it allows modules from other areas to be added to the programme very easily, simply by specifying a different job code. Contrast that with the difficulty a counsellor might have, trying to recommend modules from outwith his own area of expertise.

Core modules are built into the system, via the COLLEGE file, and if the college declares these to be non-negotiable, then the system cannot alter that constraint on choice. However, if the college is prepared to be more flexible the system, through SUPERCLASS, can suggest alternative core modules which will still satisfy the basic need.

The system does not negotiate, but it facilitates negotiation by supplying information on alternatives very quickly.

This last point is seen as one of the most important features of the system. It allows the student and/or counsellor to experiment, trying out different combinations of modules to see what kind of programme results - and he/they can do this knowing that the basic requirements are being met at each stage. The "unthinkable situation" of a student being offered modules for which he does not have the necessary prerequisite skills cannot occur in the system described, and the equally undesirable possibility of identifying a set of modules only to find that they cannot be completed in the time available is also avoided - the Sequencer will show very quickly which modules, if any, cannot be included in the Course Grid.

6.16 Artificial Intelligence

At an early stage in this project, it was felt that there were certain similarities between designing a good modular course and the game of chess. As the work has developed, some of the original examples which could have been given here have been modified, and are no longer appropriate. Some that are still relevant are;

a)i) At certain points in a game of chess a specific threat has been posed by the opponent, or a particular opportunity is seen. These situations require specific, immediate action.

ii) In course design, certain sequences of modules are recognised as being impossible to fit into one year's work. The 'excess' modules must be immediately removed to avoid giving the student the false impression that he can take these modules.

b)i) At other points in a game, there are no specific moves required. At these times a move which yields the best overall position is selected.

ii) In course design, where no module is seen as directly appropriate or there is a selection of modules to choose from, the general principles of course design should be used as a guide.

c)i) The single best move might not be the best in the long term.

ii) A module which appears relevant might bar the way to better modules, or a 'second best' module might lead to more useful additions.

d)i) Certain moves, (combinations), frequently appear together.

ii) Certain modules are best used together; a sequence of modules, one giving entry to the next, is an example.

e)i) There are certain standard ways of forcing a win from approximately similar positions.

ii) After a certain point concentrate on completing the programme, rather than trying to extend it. Do not continue looking for additional modules when a reasonable number has already been found.

Some differences were noted;

f)i) In chess there are many combinations of moves that can result in winning, but there is only one end result that can be considered a 'solution'; that result is a win.

ii) There are a number of programmes that the student could be offered. One is probably best, but others might be almost as good.

g)i) Once a move has been played it cannot be retracted.

ii) If a selected module is found to be inappropriate it can be removed.

The analogy may now appear somewhat forced, but at the time it prompted the thought that game-playing computers frequently used the ideas of Artificial Intelligence in their design. In simple terms,

> *Artificial Intelligence (A.I) is the study of how to make computers do things at which, at the moment, people are better.*
> (E.Rich, 1983, p1).

It seemed reasonable to follow this line of thought. Texts on Artificial Intelligence frequently mention Lisp and/or Prolog as suitable programming languages (Ibid. p393 and p401). Lisp is a language designed to process lists of related information, and lists feature heavily in course design; the list of modules available, the list of compulsory modules, the list of possible courses, each of which is itself a list of modules, are a few examples. However, for two reasons it was decided to use Prolog rather than Lisp. The reasons were;

a) Prolog, or rather, the smaller version, micro-Prolog, was available for the BBC Master computer, and that was the machine available.

b) *But learning about goal-oriented programming through Lisp programs is like reading Shakespeare in a language other than English.*
(Bratko 1986, foreword p vii)

Prolog was thought to be equally good at processing lists; for example,

> *The Prolog syntax is particularly suited to the succinct expression of rules which process lists.* (P.Hepburn 1987, p60)

It also appeared to offer other desirable features. For example, a Prolog program gives rules governing the relationships between various components, and states the goal to be achieved. It does not have to state how the goal is to be achieved; the language itself contains procedures which allow it to work that out. Accordingly a version which operated on the BBC Master was obtained. (Brough et al 1985). It proved to be very difficult to write programs that worked. The way in which programs were to be written was different from that of more conventional languages. In Prolog the programmer tells the computer what it has to do. He does this by giving relevant facts and the rules governing the manipulation of these facts. In a conventional language he has to tell the computer how to manipulate the facts. Given that the ideas essential to an understanding of the language were new, it was expected that some difficulty would be experienced initially, but that these difficulties would be resolved as experience was gained. After some time, a great deal of difficulty, and very little progress, help was sought from a semi-professional programmer who had a great deal of experience with the BBC computer. He found the same difficulties, but quickly confirmed that these were caused by mistakes in the instruction manual accompanying the package. In some cases an extra set of parentheses had to be used round names or statements, in some cases two extra pairs were required. In other cases, the instruction listed in the manual used the correct words but in the wrong order; changing the order, by trial and error, eventually resulted in a command that worked. Trying to work out the correct syntax for a language by trial and error is an extremely slow process, especially in a language that uses multiple parentheses and the number and position has to be found each time. The error messages were not very informative, some of them were not listed in the manual, and some that were listed seemed to be inappropriate. Eventually it was decided that it would take too long to master the syntax, and that the possibility of being able to write a useful program was very remote. The attempt to use Prolog was abandoned. The only benefit gained from this experience, and the considerable time spent on it, was that it was gradually realised that the problem to be solved did not really require the Artificial Intelligence approach, and that it could be solved using a more conventional language(6). The only part of the problem that might benefit from the use of a language like Prolog is the Sequencer. This attempts to place the selected modules in the Course Grid. Sometimes positioning one module prevents another being placed. The solution lies in 'backtracking', going back to modules fitted earlier in the process and moving them to alternative positions, thereby creating space for the later modules. Prolog can do this sort of thing automatically when it initially fails to achieve the specified goal.

6.17 Notes

1) A file containing a complete set of Module Descriptors will be very large. For that reason it may be decided to keep it separate from the other components - this is quite acceptable, as it is not needed for this system to operate. It will however have to be linked to the other files so that changes in them are reflected in it.
2) Some of these studies were concerned with proof-reading - the results may not be equally valid when applied to reading for understanding. (Shneiderman 1987 p360)
3) Naps(1992, p655), Bell (1992, p118) and Sommerville (1989 p 235) describe slightly different notations for data flow diagrams. In this work the notation given by Sommerville is used.
4) These 'programs' might be more correctly referred to as 'modules', being the components from which a complete program might be constructed.
   ".. a module is a fairly independent piece of program that typically has a name and some instructions and some data of its own. A module is invoked, or called, from some other module and, similarly, uses (calls) other modules." (Bell 1992, p117)
   In the present context the use of the word 'module' is likely to cause even more confusion than the word program (as compared with programme). The word 'subprogram' will be used to describe these components, as that seems to have the most flexible interpretation.
   "You could have a subprogram that prints only a line of data, or you could rewrite an entire program as a subprogram. The main idea is to use a subprogram to perform some specific task." (Naps 1992, p85)
5) This assumes that the JOBS-MODS file, which relates SUPERCLASS codes to module reference numbers is maintained correctly.
6) It was during the period when micro-Prolog was being investigated that the design of system was clarified.

Chapter 7
The Prototype Programs

7.1 Introduction

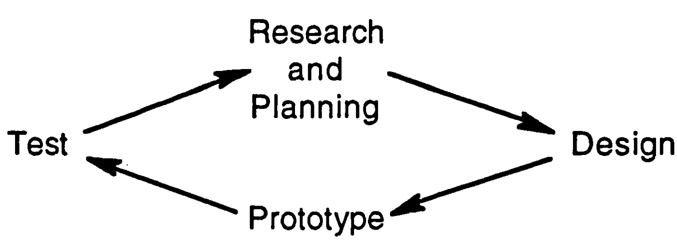
The use of prototypes is an accepted part of the process of designing large pieces of software.

"Prototyping is seen as an integral part of the object-oriented software design and implementation process. Prototyping recognises that in most case the requirements for a system are at best vague or not well understood." (Bell 1992, p143)

IBM, in discussing object-oriented interface design, identify the prototype as an integral part of the "iterative development process" and show the process as in figure 7.1 (1).

Figure 7.1

The Iterative Development Process



(IBM 1992, p100)

7.1.1 Various reasons can be given for prototyping :-

"... it is very difficult for the users...... to know whether the requirements specification actually describes software that will serve their purposes."

and

"It is rather like evaluating the drivability of a car by reading its engineering specification. We need to take it for a test drive." (Marcoty, 1991, p20)

He distinguishes between two types of prototype :-

"1. The specification prototype. This is a rough-and-ready working model of the defined requirements. The model simulates the external interface of the specified software so that the intended users can evaluate the requirements definition and the 'human factors' aspects of its use.

2. *The Technical or feasibility prototype. This is an experimental implementation intended to test the adequacy of a proposed algorithm or software design without, it is hoped, investing the large programming resources required for a properly engineered implementation."*

(ibid.)

In describing 'rapid prototype systems', Shneiderman points out that the customer and user may not have a clear idea of what the system will look like when it is finished, and that users may not realise the implications of design decisions. The prototype is a relatively cheap way of giving the customer the insight he may require.

"Unfortunately it is difficult, costly and time consuming to make major changes to systems once they have been implemented."

(Shneiderman 1987, p394)

Sommerville lists three reasons for prototyping :-

"Software development staff may find incomplete and/or inconsistent requirements as the prototype is developed. "

"A working, albeit limited, system is available very quickly to demonstrate the feasibility and usefulness of the application to management ."

"The prototype serves as a basis for writing the specification for a production quality system." (Sommerville 1989 p114)

7.1.2 Given that there are good reasons for prototyping, these same writers are agreed on one characteristic of the prototype. Marcoty used the phrase "rough-and-ready"; Sommerville, quoted above, uses the phrase "albeit limited". Bell appears to hint that the whole system need not be prototyped :-

"Prototyping is producing - very quickly - a working version of a piece of software. The emphasis is on using whatever methods are available to produce something that looks to the user like the final system."

(Bell 1992, p288)

and Shneiderman states this explicitly. He suggests incomplete simulation versions of the system, e.g. *"The simulation of or prototype of a menu system may only have one or two paths active instead of the thousands of paths in* *the final system."* and goes on to give other examples where only a fraction of the system is actually implemented. (Shneiderman 1987 p394)

In Chapter 4.6 it was concluded that the task of designing a student-centred course was not feasible, if it were to be attempted manually. It was further stated that "a computer appears to be an ideal tool". Faced with a 'difficult' task, it is all too easy for the enthusiastic amateur to say, "Use a computer" - the cynic may immediately reply, "How would a computer do it?". Parts of the proposed system were prototyped to show that the method was feasible - that a computer could do it. Indeed, it was only in the detailed coding of these prototypes that the author clarified, in his own mind, *how* the computer could solve the problem.

7.2 Characteristics of the Prototype

The prototypes were produced on a BBC Master computer and are written in BBC BASIC. Preliminary considerations had shown that, although the machine is limited, it would be satisfactory for the purpose.
Sommerville gives four recommendations concerning the approach to prototyping; for clarity these will be considered separately.

1) *Use a very high-level language for prototype implementation.*

(Sommerville 1989, p 116)

The abortive attempt at using Prolog was described in Chapter 6. The method adopted here departs from this advice in another respect. It was felt that any procedures adopted for the prototype should be as 'portable' as possible; that is, it should be possible to use the same procedures on any other machine with the minimum of alteration. To that end certain features of the BBC Master and BBC BASIC were not used, as it was not known how widespread these features were on other computers. Initially, the prototypes were written using the simplest dialect of BASIC, using only those facilities which could be expected to be available on any machine. While BASIC cannot be considered a 'very high level language', any special characteristics of the machine and language should have been used to facilitate the writing of the programs. It was a mistake not to use these facilities, a mistake caused by the failure to clarify at an early stage the use to be made of the prototype, *i.e. that the prototype itself would not be developed beyond the demonstration phase.

2) *Relax non-functional requirements such as speed and space requirements.* (Ibid.)

To simplify the coding, while some consideration was given to the speed of retrieval of records, so that the prototype was useable, relatively simple indexing systems were used. These are satisfactory when only a sample of the module records is to be considered, but would be inappropriate for a full-size version. For example, record lengths were not optimised, either to give speed or to save space. The limitations of the prototype will be stated in this chapter, improvements resulting from testing will be discussed in Chapter 8, and possible extensions to the system as a whole will be suggested in Chapter 9.

3) *Ignore considerations of error action.* (Ibid.)

Very little error trapping was incorporated into the prototype, so it is possible to give an invalid response which will crash the program.

4) *Reduce reliability and program quality standards.* (Ibid.)

The testing of the prototype will be described later.
While it is hoped that the prototype is not fatally flawed, there is no guarantee that certain combinations of input will not halt the program.

7.2.2 The Throw-away Prototype

For the reasons given above, Sommerville recommends that the prototype be considered a 'throw-away' program; having served its purpose, of clarifying the system needs and demonstrating the feasibility of the proposals, it should be discarded (Ibid. pp 114 and 120). In discussing throw away prototypes Marcoty states that

> *"Building one to throw away is not an instance of premature implementation provided that: this is the announced intention at the beginning and not a post hoc rationalization for discarding a failed implementation and it is in fact thrown away"* (Marcoty 1991, p55)

He goes further; that the prototype should not be designed to make it easily modifiable (as is required by a full implementation). This is because the prototype is not intended for prolonged use, and making it difficult to modify reduces the risk that an attempt might be made to extend the prototype beyond its designed purpose. (ibid.)

7.3 The Objectives of the Prototype

The objectives of the prototyped programs were;

a) to show that the construction of viable modular courses is greatly facilitated by the use of the computer;
b) to develop certain techniques that will be of value in constructing a fully-operational system;
c) to show that much of the work of designing a programme can be done by the student (using the computer) without the assistance of a counsellor.
d) to gain information concerning the time required by the system to produce results.
e) to gain information about the difficulties students might have in operating the system.

7.2.4 The parts of the system that were prototyped

It was felt that the following parts of the system should be prototyped :-

a) Partial construction of the Pre-entry List.
b) Most of the Selector sub-program.
c) The Sequencer.

The reasons for prototyping these sub-programs vary, and, depending on the reason, certain parts were not included. To avoid unnecessary repetition each of these sub-programs will be described as a whole, with the reasons for prototyping and the justification for the omissions.

7.4 Constructing the Pre-entry List

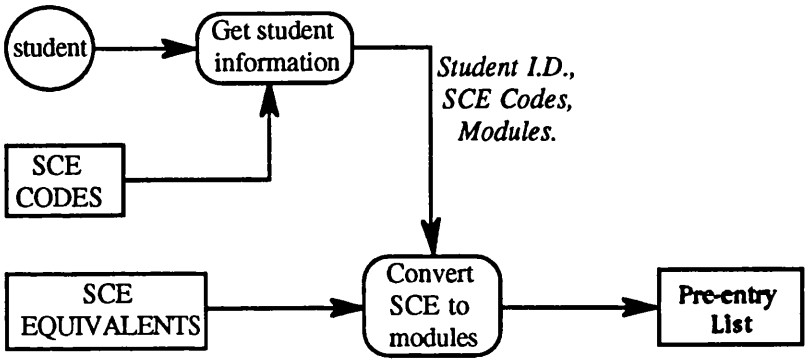
7.4.1 The program

The data-flow diagram for the construction of the full Pre-entry List was shown in Figure 6.10, and is reproduced below for convenience.

Figure 7.4.1.1

(Previously shown as Figure 6.10)

Construct the Pre-entry List.



There are no computing problems in writing this sub-program. It was written so that the students who would test the programs could be introduced to the system via an 'easy' task. Shneiderman (1987, p364) identifies three 'classes' of user :-

a) The user knows the task but not the computer-related concepts e.g. he can write letters but does not know about text editing;
b) The user knows the task and knows about editing, but is not familiar with the particular editor to be used;
c) The user knows all three components and only needs reminding.

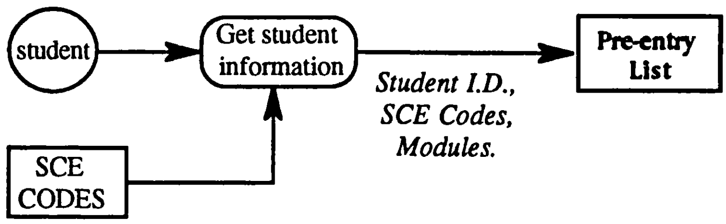
In the case of the system described here, however, the students form a fourth class - they are not familiar with either the task or the system. Students in the target group would have no experience of taking part in course design, and would have no knowledge of the SUPERCLASS system, which is a central feature of it. It was felt that they should be introduced to the system through a task with which they were familiar - which is, essentially, listing the SCE subjects and module passes that they have already gained. It was hoped that this would reduce any anxiety they might feel.

The opportunity was also taken to answer certain questions (which might be thought relatively trivial). These questions, and the answers that were found, will be discussed in Chapter 8.

As this sub-program was not being written to prove its feasibility, but as an introduction to the system, it was not felt necessary to prototype all of it. The conversion of SCE codes into modular equivalents requires a full list of approximately 90 equivalents. Constructing the SCE Equivalents file would, therefore, have taken a considerable amount of time for very little gain. The conversion occurs internally and requires no response on the part of the student. Accordingly, the part that was prototyped is that shown below.

Figure 7.4.1.2

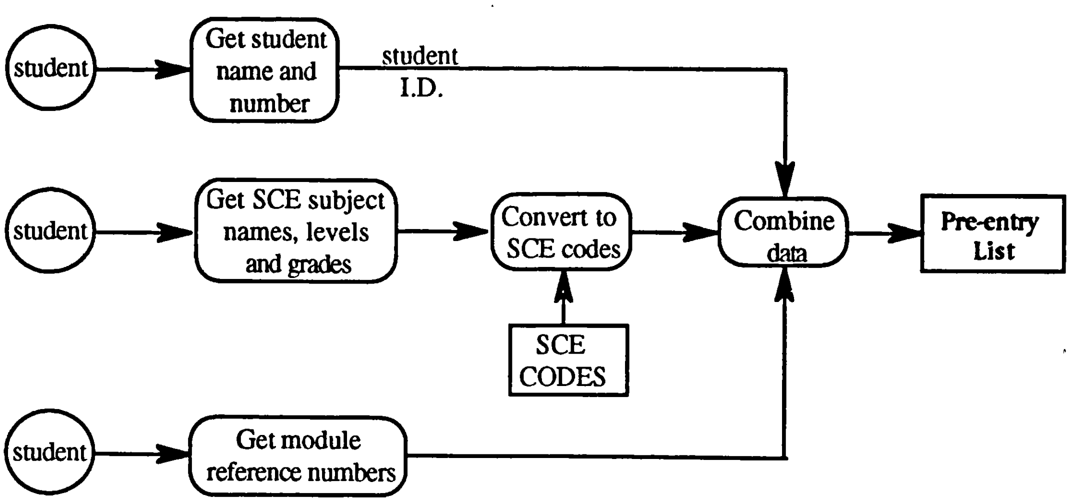
Partial Construction of the Pre-entry List



As this part contains the user interface it allowed some insight to be gained into any problems that students might encounter in using this part of the system. More detail is shown below.

Figure 7.4.1.3

Constructing the Pre-entry List



The student supplies his name, number and module reference numbers in response to simple prompts. The SCE subject names are listed on three screens, the student being prompted to highlight the appropriate name. He is then asked to state the 'level' (Higher grade, Ordinary grade or Standard Grade), and finally asked for the grade, e.g. grade 3 (S-grade) or grade C (H-grade. The combined data is then recorded as the Pre-entry List. The performance of students in this part of the system is discussed in Chapter 8.

7.4.2 The SCE-CODES File

Information will be given about this and the other files used in the prototype programs to give an indication of the way the files are used and what information they contain.

7.4.2.1 Use

This file provides a reference number for each SCE subject and grade, the possible grades being Higher, Ordinary or Standard (Standard grade being phased in to replace Ordinary). Each subject and grade is given a code number on being entered into the file. This number does not change, although the position of the subject in the file may change as subject names change and alternatives are inserted into the file.

Because Standard grades are being phased in over a number of years, some subjects are not available at that grade yet. In addition, some subjects do not appear at Standard grade because, in converting from Ordinary to Standard grade, the name has been changed. Similarly, some subject names do not appear at Ordinary grade for the same reason, or because they were changed to Standard grade some years ago and the Ordinary grade version is no longer offered (2).

The subjects and code numbers are recorded in the file in the following format;

Accounting 1 2 0 Art and Design 3 0 4 Arithmetic 0 5 0 Biology 6 7 8

that is, subject name followed by code numbers for Higher grade, Ordinary Grade and Standard grade. Where the subject was not offered at a particular grade this is shown by a '0'. Thus, Art and Design is not offered at Ordinary grade, whereas that was the only grade at which Arithmetic was offered. No separators other than spaces are used.

7.4.2.2 The data

Preliminary

number_of_records%	the number of subject names recorded
longest_name%	the length of name, used to format the display
highest_code%	the highest code reached

For each record

name$(record_number%)	the subject name
grade%(record_number%,3)	the three code numbers.

The subject names are stored in alphabetical order, in a form suitable for sequential access.

Once a code number is allocated for a subject at a grade it is never changed. If a subject or grade is deleted the code number is lost.

7.4.2.3 File Management

For the purposes of the prototype system, the file was constructed from information contained in the Scottish Examination Board document "Conditions and Arrangements" (SEB 1991, p22). This lists the subjects and grades examinable in the 1991 examination diet.

The file is managed, in the prototype system, by the program "SCEMANAGER". This is a reasonably 'developed' program. The information already entered is broken into a number of screens, or pages, each screen holding up to twenty records. The current record is shown by the name and grade code numbers being highlighted. The user moves from one record to another using the up and down arrow keys, and from one screen to another using the left and right arrow keys.

A small 'help' window at the top of the screen gives the key presses required to delete, add, or edit a record. The updated file can be saved at any time. On 'quitting' the program the file is automatically saved if any alterations have been made. The user is then given the option of printing the contents of the file.

7.5 The Selector

7.5.1 The program

Parts of the Selector were prototyped to answer the following questions :-

a) Can the students use the system to generate a Module List?

b) What difficulties do they find in using the system?

c) How long does the process take?

As the answers to these questions were the result of the evaluation, they are discussed in Chapter 8.

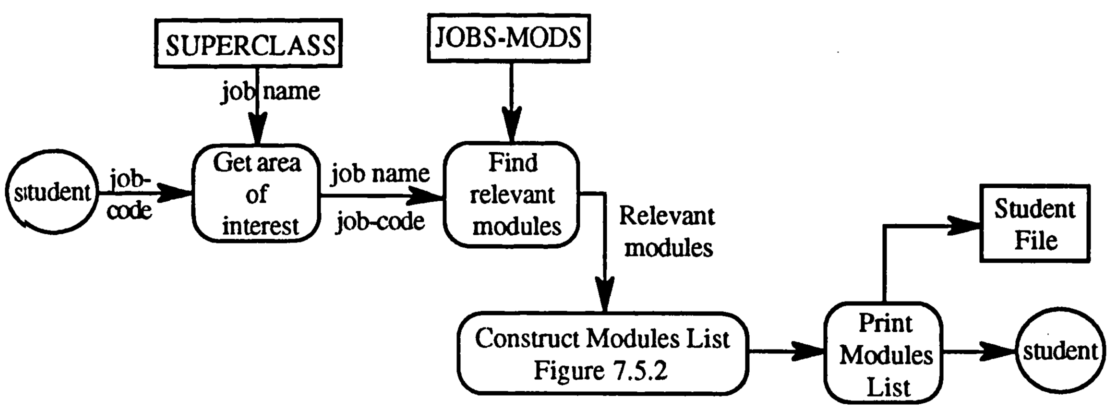
In addition, the Selector provides a Module List so that the operation of the Sequencer can be examined.

The prototyped parts of the Selector are shown in Figure 7.5.1.1, with additional detail in Figure 7.5.1.2.

Figure 7.5.1.1

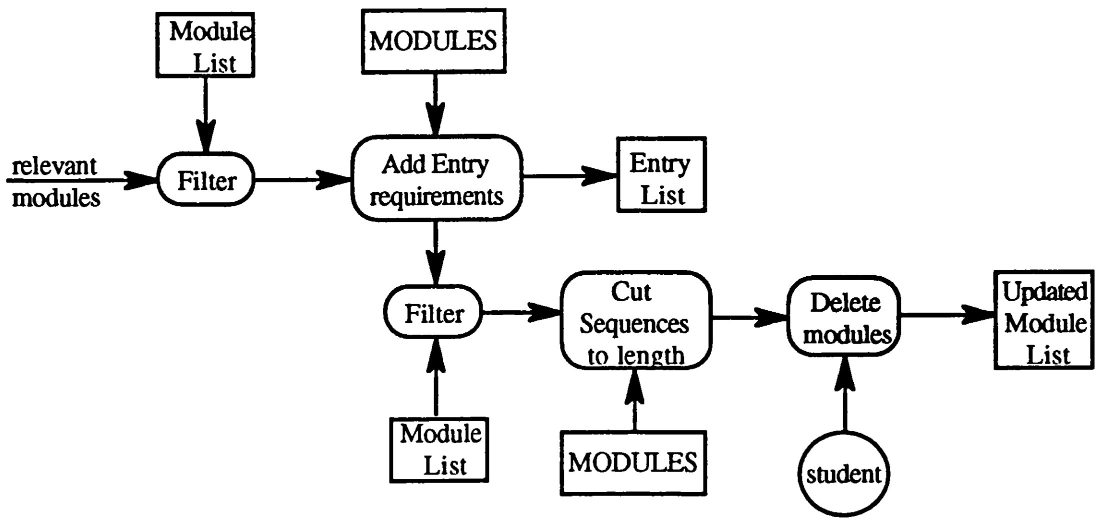
(based on Figure 6.10.3)

Produce the Student Module File.



The student is prompted to give a SUPERCLASS job-code. How he chooses the job-codes is described in Chapter 8. The SUPERCLASS file supplies the name of the job area, confirming (a) the student's choice and (b) that a valid code has been given. Given a valid job-code, the JOBS-MODS file supplies a list of relevant modules (3). These are checked for entry requirements and duplicates are removed in the "Construct Modules List" sub-program, as shown below.

Figure 7.5.1.2
(based on Figure 6.10.2.2)
Construct Modules List



The sub-programs shown above were described in the text following Figure 6.10.2.2.

It will be noted that the prototype does not check 'relevant modules' against the Pre-entry List, as this was felt to be a trivial exercise, similar to that of the check against the Module List which is performed to avoid duplication of modules. Any entry requirements to the modules are supplied by the MODULES file, which also gives the length of each module. Any sequences which are too long are cut by repeatedly removing the last module until the overall length of the sequence is the equivalent of three modules long or less. The student is given the opportunity to delete modules. If he attempts to delete an entry requirement he is warned that the later modules in the sequence must also be deleted. The processes shown in Figure 7.5.1.1 are repeated until the student stops supplying job-codes and proceeds to the Sequencer.

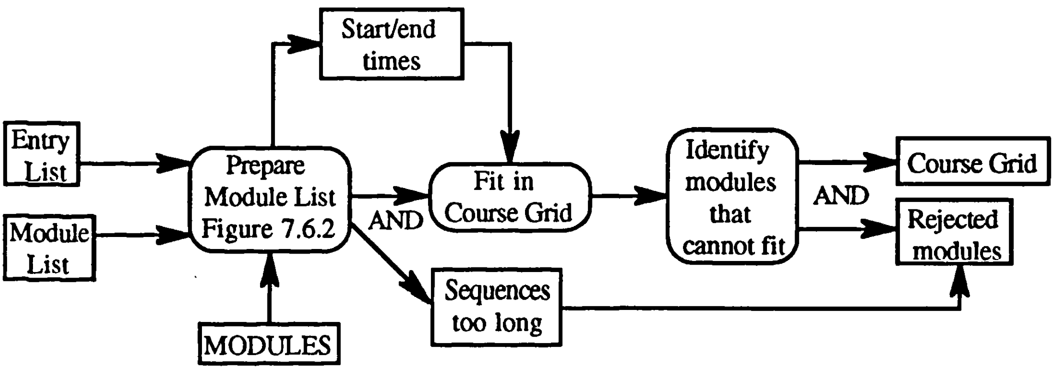
Following the pattern of Chapter 7.4 a description of the data files used by the Selector should now be given. However, it is felt that this would interrupt the description of the programs. For this reason the descriptions of the data files have been deferred to the end of the chapter, where all the remaining data files will be described together.

7.6 The Sequencer

7.6.1 A simple data flow diagram for the Sequencer was shown in Figure 6.11, but insufficient detail was given there to allow it to be implemented. A slightly modified form of that diagram is given below, the modification allowing the main part of the program to be considered as two sub-programs. As stated in 6.11, data flow diagrams should show what has to be done, not how it is to be done. Figure 7.6.1 contravenes this principle, in that the data generated is dependent on how the program works. In particular, the Start/end times are required by the implementation in the prototype - a different way of implementing the design may not require them.

Figure 7.6.1

The Sequencer Program



The sub-program "Prepare Module List" is described below.

A module can be rejected for two reasons;

a) it comes at the end of a sequence that is too long to fit into the Course Grid;

b) there is insufficient space to fit it into the Course Grid at the time it is considered.

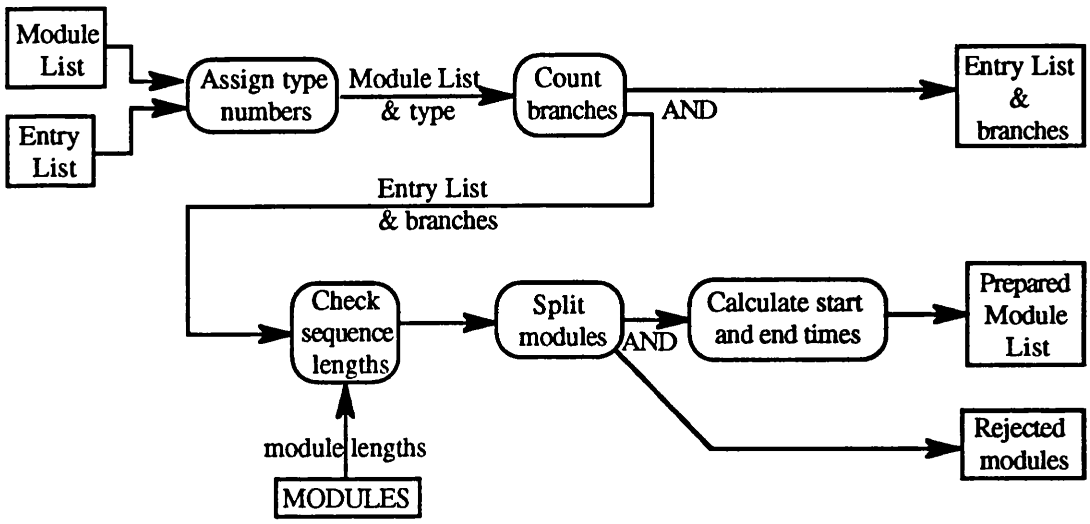
Hence the two processes that result in "Rejected modules".

7.6.2 Prepare Module List

The diagram for this sub-program is shown in Figure 7.6.2 (next page).

Figure 7.6.2

Prepare Module List



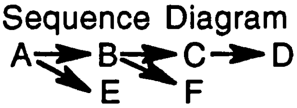
7.6.3 Assign type numbers

Unfortunately, while diagrams can give a clearer idea of a process than can a verbal description, the latter is required to explain why certain processes are carried out. For the purposes of this explanation, consider the situation when a set of modules, shown below, has been identified.

Figure 7.6.3.1

Module and Entry Lists

Sequence Diagram



Module List
A
B
C
D
E
F

Entry List	
A	B
B	C
C	D
A	E
B	F

The Module List shows the modules selected and the Entry List shows which modules are entry requirements, and to which modules they refer. (The module in the first column is an entry requirement to the module in the second column.)

When the Sequencer attempts to sort the modules into a course, it is essential that it places entry requirements before the module(s) to which they refer. The Entry List is used to assign a **type** number to modules, as will now be described;

a) if a module (which appears in the module list) does not appear in either column of the entry list, then the module is neither an entry requirement, nor does it have any entry requirements - it is an independent module. So far in this example there are no independent modules.
b) if a module appears in column 1 of the entry list but not in column 2, then the module is an entry requirement to at least one other, but does not itself have any entry requirements. It is the start of a sequence of modules. In this example, module A is the start of a sequence.
c) if a module appears in column 2 of the entry list but not in column 1, then it has entry requirements but is not an entry requirement to any other - it is the end of a sequence. In this example, D, E and F are the ends of sequences.
d) if a module appears in both columns 1 and 2 it is part of a sequence, but is neither the start nor the end of a sequence. B and C are of this type.

By considering each module in the entry list and applying the rules given above, the computer assigns a 'type' number to each module. Because of the algorithm used, the type numbers have the following meanings;

0 - an independent module
1 - the start of a sequence or sequences
2 - the end of a sequence
3 - part of a sequence (other than the start or end)

The type numbers are added to the Module List, giving the "Module List & Type" referred to in Figure 7.6.3.2. For this example the amended Module List is shown below.

Figure 7.6.3.2

Module Types

Module List & Type		Entry List	
A	1	A	B
B	3	B	C
C	3	C	D
D	2	A	E
E	2	B	F
F	2		

Simple sequences can be constructed from the Entry List and this expanded

Module List as follows :-
Read down the Module List until a type 1 module is found (A).
Find the first occurrence of this module in column 1 of the entry list (see row 1). Read the module in column 2 of the same row (B). This is the second in the sequence. If this module is type 2 the sequence is ended. As B is a type 3, the sequence continues. Find it in column 1 of the Entry List (row 2). Read the module that must follow it (C). That is a type 3, so continue the process. At row 3, column 2, module D is found, and is a type 2. The sequence is complete. The sequence thus generated is therefore ABCD, as required by the Sequence diagram.

7.6.4 Count branches

The whole process can now be repeated beginning with any other type 1 modules. However, there is a complication; a module may be part of more than one sequence. In this example, A and B are each in two sequences. A is a type 1, the start of a sequence, and can be dealt with quite easily. After following the sequence from the **first** occurrence of the module in the Entry List, a second sequence can be generated from the **second** occurrence (in row 4). Applying the process gives the sequence AE.
The difficulty lies in the fact that <u>any</u> module can be part of more than one sequence, but only those at the start can be dealt with by simply looking for subsequent occurrences in column 1 of the Entry List.

Following the procedure described above, having generated the sequence ABCD the computer has to be 'told' to go back to module B and, instead of following the first occurrence in the Entry List, go to the second occurrence of B (in the last row). It also has to remember that this is part of a sequence beginning AB ---.
The fact that module B forms a 'branch' in the sequence can be recognised from its two occurrences in column 1 of the Entry List. The number of branches for a module can be found by looking down column 2 of the entry list until a type 3 module is found. Then look down column 1 (from the top) and count all the occurrences of the module - this is the number of branches

for that module. The number of branches is recorded as a third column in the entry list. To simplify the program coding, type 2 modules are assumed to have one branch (logically they do not, as they are the ends of sequences). The number of branches is associated with the module in column 2 of the entry list. As type 1 modules do not occur in that column they are not associated with a number of branches - as explained above, and repeated below, this is not a problem.

In this example, the complete module and entry lists would appear as shown in Figure 7.6.4.

Figure 7.6.4

Sequence Branches

Module List & Type	
A	1
B	3
C	3
D	2
E	2
F	2

Entry List & Branches		
A	B	2
B	C	1
C	D	1
A	E	1
B	F	1

When the first sequence is generated the computer follows the branch number for each module, these being read straight from the third column of the entry list. In this example, the first sequence would be

	A	B	F
Branch followed	2	1	

F was reached by using the <u>second</u> occurrence of B in column 1 of the entry list.

The program then backtracks through the sequence until it finds a branch which is not 1. That branch number is then decremented and the branch indicated is followed.

This will generate the sequence

	A	B	C	D
Branch followed	1	1	1	

The program goes on backtracking until it reaches the type 1 module at the start of the sequence. It then checks to see if that module is the start of another sequence, by reading further down column 1 of the entry list for more occurrences of the module. (This is why the fact that type 1 modules are not associated with a number of branches is not a problem.) If another occurrence is found, the process described above is repeated.

7.6.5 Check sequence lengths

The total length of a sequence is found by constructing the sequence, finding the length of each module, from the MODULES file, and summing them. If the total length is greater than 6 the module at the end is repeatedly removed until the length is 6 or less (4).

7.6.6 Splitting modules

In terms of the standard module length the longest sequence contains only three modules. After some consideration it was decided to introduce an idea that may be somewhat contentious. Normally, in one week a student would study eight modules, each module being studied for three hours. This would continue until the end of the thirteen week block. However, it is possible to study a module for six hours per week for half of the block, with a different module being studied for the remainder of the block, also for six hours per week. Is this educationally desirable? Anecdotal evidence appears to suggest that opinions vary depending on the subject matter being taught. Where the skills being taught involve manual dexterity, such as learning typewriting, some teachers feel that a short, intensive period of teaching is preferable to a longer period amounting to the same number of hours. The same opinion is frequently expressed by teachers of foreign languages. Frequent exposure to the subject allows the student to learn a basic vocabulary quickly; this can be followed by a less intensive regime, in which the finer points of the language are studied and vocabulary is extended. It might be argued that shorthand is a combination of the two 'skills'; learning the code (the vocabulary) and acquiring the dexterity to write the code quickly. On the other hand, teachers of some subjects have argued for less exposure per week, but continue for more weeks. The argument here appears to be that there is a limit to the number of new ideas that can be assimilated in a given period of time, that the student needs time to 'mull it over' between lessons. Where the subject includes elements of both theoretical and practical work, teachers have been known to argue both cases; the theoretical part should be spread out, but the practical should be concentrated so that less overall time is wasted setting up and dismantling equipment. (Strictly speaking, this is a different situation as the student would still spend three hours per week on the practical element, but it would be concentrated into one morning instead of being spread throughout the week.)

It was decided that the program should include the possibility of 'splitting' modules. Two modules are shown below. Module A is as it would normally appear, module B has been split.

```
: A : A :
: B :
: B :
```

The consequences of splitting modules are;

a) fewer subjects are studied in any given week (in the part of column 1 shown, only two module names appear instead of the usual three);
b) the total number of half-module equivalents studied in the course remains unchanged (the course grid contains forty-eight half-modules, no matter how they are filled);
c) more modules can be added to sequences.

This final point is worth considering in more detail. Consider this sequence;

```
: A : A : B : C : C : D :
```

If modules A and C were split, space would be created for other modules to be added (these extra modules may themselves be split);

```
: A : B : C : D : E : F :
: A :     : C :     : E : F :
```

Modules B and D cannot be split because they are only one half-module long. This prompted the question, "Which modules can be split, and which cannot?"

Consider a module lasting one and a half of the normal lengths (i.e. it is three half-modules long). This cannot be split into equal halves - there can be no unit smaller than the half module. If it is split it must be either

```
: A : A :   or   : A : A :
: A :            :   : A :
```

It was decided that this would not be allowed in the prototype program, as it might be confusing to students and lecturers alike. If it is debatable whether modules should be split at all, then it would be even less satisfactory to split them in such a way that at one time a module is studied for six hours per week and then suddenly changes to three hours per week (or vice versa). This effectively means that a module should not be split if its length is an odd number of half-modules.

A module lasting the equivalent of four half-modules could be split twice. For example, the module : A : A : A : A : could be split once;

```
: A : A :
: A : A :
```

and then split again

```
: A :
: A :
: A :
: A :
```

It was decided that this should not be allowed. The normal week (without split modules) contains eight modules; splitting a module as shown above would mean half the week being devoted to one module.

Summarising the rules applying to splitting modules;

a) a module can only be split if its length is an even number of half-modules;

b) a module cannot be split more than once.

Allowing for splitting modules, deciding whether a sequence can be fitted into the course grid becomes a little more complicated.

If the sequence length is less than or equal to six half-modules, then the sequence can be fitted.

Because of the rule that a module can only be split once, a sequence length cannot be reduced to less than half of its original length, so if the sequence

length is greater than twelve, it cannot be fitted.
If the sequence length is greater than six and less than or equal to twelve, the decision as to whether it can be fitted or not depends on the length of the sequence and the lengths of the individual modules.

When a module is split, the sequence length 'loses' a length equal to half the length of the module, e.g. : A : A : loses one half-module length when it is split;

: A :lost :
: A :

and : A : A : A : A : loses two half-module lengths when split;

: A : A :lost :lost :
: A : A :

To make a sequence fit, it must be possible to reduce the length to six by splitting. Whether this can be done or not can be calculated as follows;
Calculate the length of the sequence (call it sequence_length).
Calculate the excess, from

excess = sequence_length - 6

For each module in the sequence, if the module length is even calculate the length that can be lost by splitting the module;

length_lost = module_length / 2

For the sequence length to be capable of being reduced to six, 'excess' must be less than or equal to length_lost.

Some examples will illustrate this.
Example 1

: A : A : B : B : C : C : D : D : E : E :
sequence_length = 10
excess = 10 - 6 = 4
length_lost = 1 + 1 + 1 + 1 + 1 = 5 (all modules can be split)
4 < 5 (excess is less than length_lost)
therefore the sequence can be made to fit.

Example 2

: A : A : B : B : C : C : D : D : D : E :

sequence_length = 10

excess = 10 - 6 = 4

length_lost = 1 + 1 + 1 = 3   (D and E cannot be split)

4 > 3 (excess is greater than length_lost)

therefore the sequence cannot be fitted.

Split modules can be dealt with in two ways;

a) Treat the two halves of the module as two separate modules whose only difference is the reference number (one half has the suffix 'A', the other 'B'). Because the parts are identical they will treated equally by later parts of the program.
b) Treat them as one module, flagged to indicate it is to be split, and use appropriate coding in the Sequencer to fit the two parts into the Course Grid. It is this method that is used in the prototype.

7.6.7 Calculate start and end times

Four 'times' can be calculated for each module; these are referred to as

a) the earliest start time
b) the earliest end time
c) the latest start time
d) the latest end time.

All of these times are expressed as column numbers in the Course Grid. These figures allow the Sequencer to place modules correctly in the Course Grid, taking account of any entry requirements.

7.6.7.1 Calculating the earliest start and end times

Initially, in the prototype, all modules are given an earliest start time of 1. If a module is independent of all others (type 0) or comes at the start of a sequence (type 1), this will not change.

For all types, the corresponding earliest end time can be calculated from earliest end = earliest start + length - 1

For example,

a) a module of length 1 starting in column 2 finishes in column 2 (it only occupies 1 column, which is the one it starts in).

b) a module of length 3 starting in column 2 finishes in column 4 (occupying columns 2, 3 and 4).

The instructions for setting these initial conditions are :-

```
REPEAT for all modules
        earliest start = 1
        earliest end = earliest start + length - 1
END REPEAT
```

Other modules cannot start until their entry requirements have ended. These modules (types 2 and 3) start in the column after the end of the entry requirement :-

earliest start = earliest end of entry module + 1

The instructions for this process are given below. (In the same way that an entry requirement can be referred to briefly as an entry module, a module which has an entry requirement can be referred to as an 'exit' module.)

```
REPEAT
        IF module is an exit module, THEN :-
            earliest start = earliest end of entry + 1
            earliest end = earliest start + length - 1,
        ELSE continue
UNTIL all modules have been checked (5)
```

There are two complications which must be taken into account by the program. To avoid undue repetition of the instructions given above, the two complications will be described and then the amended instructions will given.

If modules appear in the Module List in the correct order, with entry requirements being listed first, the rules given above need only be applied once. Unfortunately, many modules will be listed in reverse order. When a module is recognised as relevant it is placed in the Module List. If it is then found that the module has an entry requirement, this is added to the List **after** the module to which it refers. If that is the case, then the position of the exit module is calculated on the basis of the end time of the entry module, which might, subsequently, be changed. The position of the exit module will then be incorrect. It is necessary to recalculate the start and end times until no changes are noted. (Hence the test condition 'change' in the instructions given below.)

The other difficulty is that a module can belong to more than one sequence. During the calculation of the start and end times, it is possible for a module to be given a correct starting time and then have this reset to an earlier time because it appears earlier in another sequence. On the assumption that a module is given a late start time for a reason, it should never be decreased. This can be accomplished by calculating a 'possible new start time', comparing this with the existing start time, and only changing the start time if the possible time is later than the existing start time.

The full set of instructions for calculating the earliest start and end times can now be given :-

Figure 7.6.7.1.1
Caluating Earliest Start and End Times

```
REPEAT
        IF module is an exit module, THEN :-
             earliest start = earliest end of entry + 1
             earliest end = earliest start + length - 1,
        ELSE continue
UNTIL all modules have been checked
REPEAT
        change = 'no'
        REPEAT
             IF module is an exit module, THEN :-
                  possible earliest start = earliest end of entry + 1
                  IF possible earliest start > earliest start THEN
                       earliest start = possible earliest start
                  ELSE continue
             earliest end = earliest start + length - 1,
             change = 'yes'.
             ELSE continue
        UNTIL all modules have been checked
UNTIL change ='no'
```

7.6.7.2 Calculating the latest start and end times

The process is similar to that described above, but begins at column 6 and works forward to column towards column 1. All modules are, initially, given latest end times of 6. For independent modules or those at the end of sequences (types 0 and 2) this will remain unchanged.

Types 1 and 3 must end one column before their entry requirements start.

latest end = latest start of entry requirement - 1

The latest that a module can start is determined by its length and its latest end time.

latest start = latest end - length + 1

For example,

A module of length 1 ending in column 5 starts in column 5.

A module of length 3 ending in column 4 starts in column 2, occupying columns 2, 3, and 4.

For the reason given above these calculations must be repeated until no further changes are made.

The process of calculating the latest start and end times can be described by the following instructions.

Figure 7.6.7.1.2
Calculating the Latest Start and End times

```
REPEAT for all modules
        latest end = 6
        latest start = latest end - length + 1
END REPEAT
REPEAT
        change ='no'
        REPEAT
             IF module is an entry, THEN
                  possible latest end = latest start of exit - 1
                  IF possible latest end < latest end THEN :-
                       latest end = possible latest end
                  ELSE continue
                  latest start = latest end - length + 1,
                  change = 'yes'.
             ELSE continue
        UNTIL all modules have been checked
UNTIL change ='no'
```

7.6.8 Flexibility

Consider the following cases :-

i) module A, length 1, type 0 (independent)

ii) module B, length 3, type 0

iii) module C, length 1, type 1, at the start of a sequence of length 5

Compare the possible ways they can be fitted into the Course Grid.

1	2	3	4	5	6	
A						
					A	
B	B	B				
			B	B	B	
C	X	X	X	X		('X' is the rest of
	C	X	X	X	X	the sequence.)

Module A can fitted anywhere between columns 1 and 6 (inclusive).

Module B can only be fitted between columns 1 and 4.

Module C can only be fitted between columns 1 and 2.

As modules are fitted into the Course Grid the number of vacant cells remaining decreases. A module which can be placed in a number of different columns will be easier to place, or, at least, it is more likely that a vacant cell will be found if a number of columns can be considered, rather than being limited to only one.

This introduces what will be referred to as the 'Flexibility' of a module. This is calculated from the equation

flexibility = latest start time - earliest start time

The greater the flexibility, the more likely it is that we will be able to place it in the Course Grid. Flexibility can vary from 0, for a sequence which fills one row of the Course Grid, to 5 for an independent half-length module. (There are other variations; for example, a single module of length six half-lengths occupies a complete row and therefore has a flexibility of 0).

This definition of flexibility is not entirely satisfactory, but it allows a 'first attempt' to be made at fitting modules into the Course Grid.

Split modules are identical apart from the suffix; because they occur at the same point in the sequence they will have identical start and end times, the same flexibility (0), and will therefore be placed in the same column in the Course Grid by the Sequencer.

7.6.9 Fitting the modules

Associated with each column of the Course grid is a counter indicating the number of vacant cells in the column; initially these counters are all set at eight. As explained above, modules with low flexibility are more difficult to fit. Sequences tend to have low flexibility, although this is not always the case. In addition, sequences form a valuable part of the student's course as they allow study to advance to more specialised work. Because of this, sequences are fitted to the grid before independent modules, which tend to be more flexible and less valuable (although this is not always the case). When the sequences are being fitted, and again when the independent modules are fitted, they are entered according to their flexibility, the least flexible being entered first. The procedure for fitting any module is the same - only the order in which they are entered depends on sequence membership and flexibility. Figure 7.6.9.1 below shows the 'description' of a module obtained by the methods already described.

Figure 7.6.9.1

Description of a module

Module	L	es	ee	ls	le	f
A	2	2	3	4	5	2

L = module length. es = earliest start, ee = earliest end,
ls = latest start, le = latest end, f = flexibility
In this description, module lengths are given in terms of the half-module unit.

The module is about to be fitted into the Course Grid shown below.

1	2	3	4	5	6
X	X	X	X		
		X	X	X	X
X	X	X		X	X

The earliest start column is 2, so the first column to be considered will be 2. The length of the module is 2, so two spaces in adjacent columns starting at column 2 must be found.

There is space in column 2, but not in column 3, so the module cannot be fitted in columns 2 and 3.

The search now moves along the Grid by one column. Look for two adjacent spaces starting in column 3. Column 3 has no space.

Move along the Grid again. Look for two adjacent spaces starting in column 4. Columns 4 And 5 both have a space available, so the module can be fitted.

Note that, if there had been no space in column 4 the module could not have been fitted - the latest start time is 4.

The process can be summarised in Figure 7.6.9.2

Figure 7.6.9.2

Fitting a Module

```
startcolumn = earliest start for the module
REPEAT
      length = 0
      column = startcolumn
      REPEAT
          IF space in column THEN count = count + 1
          column = column + 1
          length = length + 1
      UNTIL length = module length
      IF count = module length THEN fit module
      ELSE startcolumn = startcolumn + 1
UNTIL module is fitted OR startcolumn > latest start
```

The inner REPEAT counts the spaces from startcolumn to the end of the module. If the number of spaces found (count) is equal to the length of the module, then the module can be fitted. If not, the starting point for the search (startcolumn) moves along by one column and the process repeats. The process stops when the latest start time for the module is exceeded (or the module has been fitted).

In general, the length of the module indicates the number of columns that must be considered together, the earliest start gives the first column that should be tried, and the latest start gives the last column that can be used. If space is found, the module reference number is written into the appropriate cells and the program moves to the next module to be fitted. If space is not found the module is flagged as 'not fitted', and attention moves to the next module.

It may be observed that this procedure makes an assumption that has not been explicitly stated; if the module length is greater than 1, all parts of the module must fit in consecutive columns. In Figure 7.6.9.3, below, row 1 is allowed, row 2 is not, because the two parts of module D do not come in adjacent columns.

Figure 7.6.9.3

The parts of a module must be in adjacent columns

	1	2	3	4	5	6
1	A	A	B	C	C	
2	D		D	E	E	

The procedure for filling in the Course Grid can now be described.

a) For each module in the list identified as relevant to the student's needs by the Module Selector, calculate the earliest and latest starting positions and the flexibility factor.

b) The least flexible modules are the most difficult to fit, because there are fewer places that they can be fitted. By placing them early in the process, when there is most empty space, it will be easier to fit them.

c) Continue by adding the 'optional' modules, taking them in order of flexibility, starting with the least flexible.

7.6.10 The requirements specification

The requirements specification can now be extended :-

If modules are to be split

a) only modules with an even length can be split

b) a module cannot be split twice

c) all parts of the module must appear in adjacent columns of the Course Grid.

The latter is a special case of the more general rule, that all parts of any module must appear in adjacent columns of the Course Grid.

7.7 The data files

7.7.1 The SUPERCLASS File

The 'real' SUPERCLASS file is maintained by the Department of Employment. The file discussed here was developed to simulate the real file, for the purposes of the demonstration programs.

7.7.1.1 Use

The SUPERCLASS file is used for two purposes;

a) It provides a printed list of job names, from which the student can select those areas of interest to him; this will be discussed in Chapter 8.

b) It is used to relate job names to job codes (6).

The JOBS-MODS file, described in 7.5, does not contain the names of jobs associated with modules; to save memory, it only contains the SUPERCLASS code. By referring to the SUPERCLASS file the name of the job can be found, for display or print-out.

7.7.1.2 Structure of the SUPERCLASS File

While this file is essential for the purposes of demonstrating the prototype system, its structure is of secondary interest. The file structure was therefore chosen to simplify the coding, rather than to maximise its speed of operation.

7.7.1.3 The data

The first entry in the prototype file is the date when the file was last updated. This allows up-dated versions to be distinguished from older versions. Following the date is the number of SUPERCLASS records in the file, and the number of characters in the longest class name. This was included so that, if the file is to be converted to random access (7), an estimate of the appropriate record length can be obtained.

The rest of the file consists of a number of records, each record containing two fields. The first field is the class identifier, e.g. A311, the second field is the class name, e.g. Self Management.

Preliminary

date$ (8)	the date and time when the file was last updated

number_of_records%   the number of records in the file
longest_name%   the length of the longest class name

For each record
class$ the class identifier, e.g. A2 (the job code)
name$ the class name, e.g. Small Businesses

7.7.1.4 File Management

For the prototype the SUPERCLASS file was constructed using a simple program, STARTSUPER. This consisted mainly of records written into DATA statements, read by the few lines of program and written to the file. Initially only the top two levels of SUPERCLASS were included, i.e. codes of the type A1, B3, etc. Originally it had been intended that this file would be up-dated automatically as new entries were made in the MODULES file. However, it was decided that, rather than complicate still further the coding of the file-manager program for the MODULES file, it would be just as convenient to edit the SUPERCLASS file directly, by adding to the DATA statements and re-running the STARTSUPER program.

7.7.2 The JOBS-MODS File

7.7.2.1 Use

When the student is asked to state the area in which he is interested, he responds by giving a SUPERCLASS job-code. The computer then has to find all modules which are considered relevant to that job. The JOBS-MODS file relates the job code directly to module reference numbers, so that any relevant module, no matter which category it belongs to, can be identified immediately. There is thus no need to read every module record to see if it is relevant to the job - this would be a lengthy process in complete file.

7.7.2.2 The data

The date on which the file was last updated is stored so that redundant versions of the list can be identified and destroyed.

To prevent over-running the end of the file, the number of entries appears next.

The file continues with records containing;

a SUPERCLASS job code;

the number of modules considered relevant to the implied job;

a list of the reference numbers of modules relevant to that job.

Preliminary

date$	the date when the file was last updated
number_of_records%	

For each record

job_code$	the job code	
number_of_modules%	the number of modules relevant to the job code	
reference%	}	the list of reference numbers
:	}	of relevant modules
reference%(number_of_modules%)	}	

7.7.2.3 File Management

The file was initially created using the short program, "STARTJOBS". This consisted mainly of DATA statements containing the information to be stored for each record. All job codes at the first and second levels were included. Each job code was followed by the number '0', indicating that no relevant modules were listed.

Thereafter the file was updated by hand when modules were added to the MODULES file. This was done by adding the module reference number to the list associated with the job code, and incrementing the "number_of_ modules%". (Initially, when the latter was '0', there was no list.)

As this file is an integral part of the process of locating modules relevant to a specified job, it was thought desirable that this file should be designed for rapid retrieval of records. It was considered advantageous to be able to jump straight to any given job code. The file was therefore constructed as an indexed sequential file, i.e. the file is accessed through an index file (JOBSINDEX, described below). Given the size of the sample data, this was probably an unnecessarily sophisticated approach, for a prototype (9).

By entering new job codes at an appropriate point in the list of DATA statements the job codes are kept in SUPERCLASS order. That is, A11 follows A1, A111 follows A11 etc. (See Figure 7.7.2.3.1, below.) This figure shows, for the first three records, sample data that would make up a complete record. This ensured that any list printed from the file would have the records in the correct order; because this is an indexed sequential file, the only other benefit to be derived from this order is that, if a record is missing this will be discovered at an early stage in the search, as soon as a record further down the list than the target record should have been is encountered (10).

Figure 7.7.2.3.1

The JOBS-MODS File

(1)	(2)	(3)		
A1	2	1234567	1234568	
A11	1	1234569		
A111	3	5123454	4321567	8765432
A123				
A2				
A213				
A3				
A3234				

(1) = the job-code
(2) = the number of relevant modules
(3) = the reference numbers of the relevant modules

7.7.3 The JOBSINDEX File

7.7.3.1 Use

The JOBSINDEX file is the index file to the JOBS-MODS file. When a job code is given, the computer reads from the start of this file until the target job code is found. The position of the relevant job code record in the JOBS-MODS file is then read.

Each record has two fields; the first contains the job code, the second the pointer to the position of the job code in the JOBS-MODS file. Note that this is the 'position', not the record number; no calculation is required to locate the record in the JOBS-MODS file.

job_code$	a job code
position%	the position of the job code record in the JOBS-MODS file

7.7.3.2 File Management

The file was created and updated automatically by the program STARTJOBS, which manages the JOBS-MODS file. Before each record is written into the JOBS-MODS file, the pointer to this file is read. The value obtained is then written to the JOBSINDEX file after the job code. The JOBSINDEX file is thus updated so that it always matches the JOBS-MODS file.

7.7.4 The MODULES File

7.7.4.1 Use

This file contains the information, listed below, about each module. It is at the heart of the computer system described here; it is used by the Selector program when choosing modules, it is, or should be, linked to other files to keep them updated and, in a fully implemented system, would be used by the student reading about the purpose and content of the module when deciding to choose one module rather than another.
The file and its file manager will be described in some detail.

7.7.4.2 The data fields within a record.

Each record contains the following information about each module.

Category
Reference number
Title
Length
The number of entry requirements
List of the entry requirements
The number of job codes
List of job codes
The number of hobby/interest codes
List of hobby/interest codes
The number of exit modules
List of exit modules

The module **Category** is the sub-section of the catalogue in which the module can be found. This is a SUPERCLASS code at the second level, e.g. A2, B3.

The **Reference Number** may be a five or seven digit number(11). Eventually all reference numbers will be converted to the seven digit system.

The **Title** needs no explanation.
The **Length** of the module is expressed in terms of the normal forty-hour basic unit. When the computer reads this number it multiplies it by two. This allows the 'invisible' working of the machine to be confined to manipulating integers, which can be processed more quickly than can floating point numbers. For any screen display or printed copy, the computer shows the 'normal' length as a user would expect.

The **Entry Requirements** for a module, more correctly called the Preferred Entry Level, are expressed as a list of qualifications deemed to be prerequisites to studying the module under discussion. This will normally be stated in terms of module reference numbers, or SCE subject code numbers.
Subject names and module titles are not required as part of this field; they are not needed by the computer, which works with the code numbers and module references. If the full information about the module has to be displayed these codes and references can be expanded by looking up the list of subject names and grades, for SCE subjects,(12) or by referring elsewhere in the Module File, for module titles.
The information concerning Preferred Entry Level is given in two parts. The first part, the number of entry requirements, is needed by the computer in order to make sense of the list that follows.
Both the SCE subject codes and module reference numbers are expressed as integers and can be read using the same instruction; they can be distinguished later by their size. Module reference numbers have a

minimum of five digits, the smallest being 11111. The SCE codes, as used in the prototype, are numbered from 1; with approximately ninety subject names, each being offered at three grades, the highest number used at present is two hundred and seventy. If seven different codes are allowed for Standard grade subjects each subject will have nine code numbers associated with it (13), giving a biggest code number of eight hundred and ten. Even allowing for expansion, and the use of larger numbers as Ordinary grades are replaced by Standard grades, it will take some time before an SCE code number reaches eleven thousand, when it could be confused with a module reference number.

The **Number of job codes**. The size of the list of job codes must precede the list.
The **list of job codes** is the list of SUPERCLASS categories identified as being associated with the module; they are the vocational areas to which the module is relevant. Names are not given in this file; the job names can be obtained from the SUPERCLASS file when they are required for display or printed output.

The **Hobby codes** were intended to indicate the hobbies or interest/recreational areas to which the module was considered relevant. These codes can be obtained from inspection of the SUPERCLASS catalogue. However, when the sample data was being input to the prototype system it was found that the hobby codes were invariably the same as the job codes; for some people a job can be considered as a hobby, and vice versa. It was decided to abandon the idea of a separate hobby code, the job code being taken to include job, hobby, interest and recreational areas. The prototype file manager for the MODULES file has not been changed, however, and still prompts for hobbies; the user simply responds '0'.

Exit modules are modules to which the current module is an entry requirement. Their use was abandoned for the prototype; they will be considered in Chapter 9.

7.7.4.3 The data

The only preliminary data would have been the number of records in the file. However, as the file is only ever accessed via an index file (described below), the number of records is stored at the beginning of that file.

For each record:-

cat$	the sub-category, within the SCOTVEC catalogue, containing the module, e.g. A2.
ref%	the SCOTVEC reference for the module.
title$	the module title.
modlength	the length of the module, in terms of the nominal 'standard' length of forty hours.
nentry%	the number of entry requirements that follow.
entry%(nentry%)	the references for the entry requirements. These can be SCE codes (from the SCE-CODES file).
njob%	the number of job codes that follow.
job%(njob%)	the codes for any jobs associated with the module. The job codes are taken from the SUPERCLASS system.
nhobby%	the number of hobby codes that follow.
hobby%(nhobby%)	the code for each hobby associated with the module.
nexit%	the number of exit module references that follow.
exit%(nexit%)	the references to exit modules. An exit module is a module to which the current module is an entry requirement.

The file is organised as an indexed sequential file.
With the exception of the reference number and length, the quantity of information in, or more importantly, the amount of space occupied by, each field can vary considerably, with a corresponding variation in the space occupied by different records. Some modules have no preferred entry level,

others have a number of pre-requisite modules. Some highly specific modules will relate to only one vocational or interest area, while more general modules will have a number of entries. It would be very wasteful of disc space to store these records in a form suitable for random access - all records would occupy the same space as the longest record, and this could be considerably longer than the shortest or even the average length of record.

The 'indexed sequential' method of storing the records was therefore adopted for the Module file, with a Module Index file, called MODINDEX being created to allow the speedy location of records. This relates the reference number of a module to its position in the MODULES file.

7.7.4.4 File Management

The file is managed by the program, "MODULEEDIT", described later.

In the MODULES file, the module records are stored in the order in which the modules are entered, i.e. a new module is simply added to the end of the file. The MODINDEX file allows the module information to be retrieved in the correct order, which, as will be explained, approximates catalogue order.

7.7.5 The MODINDEX file

7.7.5.1 Use

This is an index to the MODULES file. The MODULES file may be described as 'fully indexed'; every record there has a corresponding entry in this index file.

7.7.5.2 The data

Preliminary

nrecs% the number of records in the index file; as explained immediately above, this is the same as the number of modules in the MODULES file.

lastptr% the position, in the MODULES file, of the first space where a new module can be entered.

For each record :-

ref% the reference number of the module.

locn% the position, in the MODULES file, where the module information begins.

As all the fields are integers and all records have two fields (including the Preliminary data, if that is regarded as one record), this file lends itself to direct access. All the high speed methods for locating a record in this file can be used, although for the prototype these methods were not justified.

7.7.5.3 File management : the MODULEEDIT program

Entries in the MODINDEX file are controlled by the "MODULEEDIT" program and are entered automatically. The user does not access the file directly, and does not need to know of its existence.

The construction and ongoing maintenance of the MODULES file are of fundamental importance to the operation of the system. The prototype program that manages the MODULES and MODINDEX files is therefore described here in some detail.

To simplify the coding of the prototype, adding a new record was regarded as a special case of editing a record. When the user indicates that a new record is to be added, the computer displays a 'dummy' record, which the user then alters as appropriate(14). The following information must be supplied by the user;

a) the position of the module in the catalogue;
b) the category in which the module is placed in the catalogue;
c) the reference number of the module;
d) the title of the module;
e) the length of the module;
f) the preferred entry level, by giving the code numbers of SCE subjects or reference numbers of other modules;
g) the SUPERCLASS identifiers for jobs associated with the module;

The program then supplies the following data;

1) the number of entry qualifications given in (f), above;
2) the number of jobs identified in (g), above;

The user is prompted to supply the data by the computer working through all the fields in sequence, the user responding as required.

a) The user locates a record adjacent to the position for the new record, and indicates whether the latter is to be placed before or after the former.
b) The user types the category, at the second level, into which the module should be placed.
c) The reference number is the five or seven digit number assigned by SCOTVEC to the module.
d) The title is only required by a human user; the computer makes no use of it.
e) The length of the module, expressed in terms of the standard forty hour unit, is entered. '0.5' and '.5' are equally acceptable.
f) In the prototype sample of data, only module references were given; SCE subject codes were not used. A separate routine in the program is required to handle the SCE codes.

The computer displays a list of references already entered, each reference being numbered. In addition, one extra number, with no entry, is shown. Figure 7.12.1, below, will clarify what is meant.

Figure 7.7.5.3

Adding the preferred entry level

(A)		(B)	
1	1234567	1	1234567
2	1234568	2	1234568
3		3	1234569
		4	

In this example, part A, two modules have already been specified as entry requirements. The user is then asked which of these he wishes to change.

If he wanted to add an entry requirement, he would type '3', and then type the reference number. The display would then change to that shown in part B of the figure. He can then go on changing or adding to the list.
For a new record the list would be empty; initially only the number '1' would appear.
When the user has finished altering the list he types '0'; the computer then moves to the next field.
Although this procedure may appear clumsy (and a better method is to be recommended for a fully implemented system) it allows the computer to 'work out' the number of entries in the list as it goes, rather than have the user supply the number before starting the list. This was thought desirable, in case any modules had a lengthy list of entry requirements.

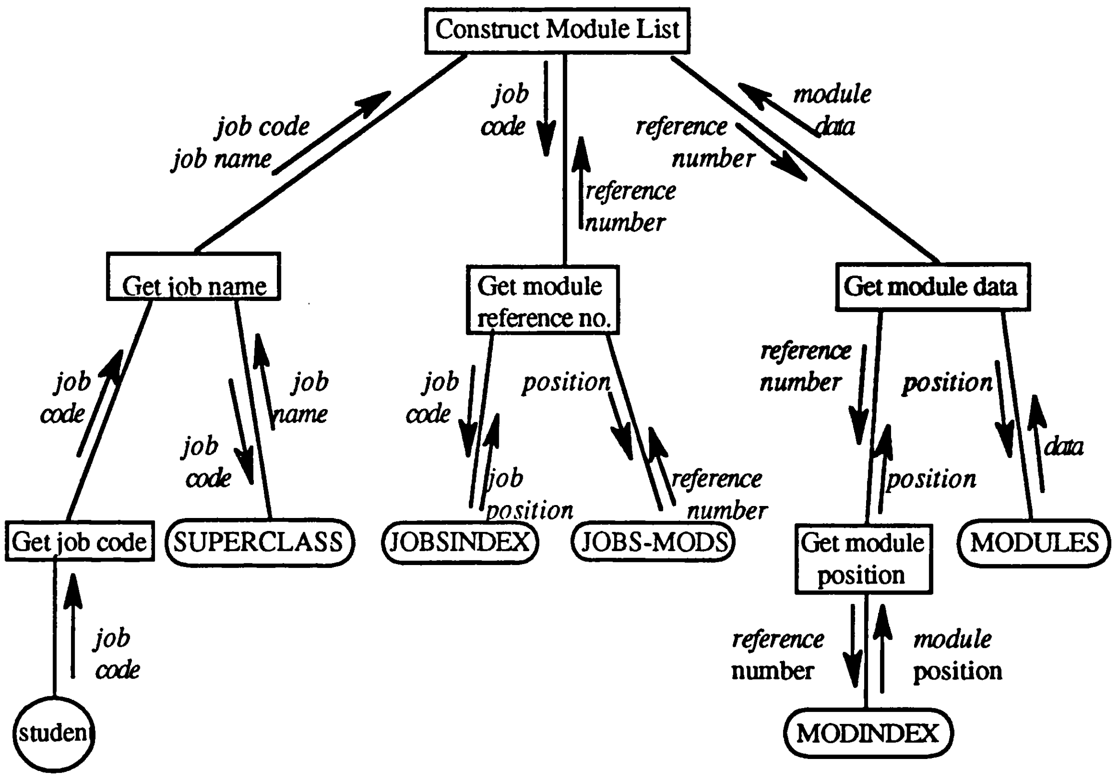
g) The job codes are entered using the same procedure as described above in (f).

This constitutes the data recorded for each module, as listed for the MODULES file, above.
New and amended records are added to the end of the MODULES file. When such a record has been filed the file pointer records the position of the next byte, at which any new record must be added; this is the number contained in the variable 'lastptr%'. The reference number of the module and the position in the MODULES file where it was recorded are written into the index file. The entry is inserted in the position specified by the user in (a), above. The records in the MODULES are therefore in a random order, while in the MODINDEX file they are kept in the order required. Inserting a new record in the index file involves moving all later records further along the file to create space. If the MODINDEX file is enlarged substantially, as it will be when a larger sample of modules are added, this could slow down the operation of adding modules. Ways of overcoming this will be suggested later, when the recommendations for full implementation of the system are given.

7.7.6 The use of the files is summarised in the chart below [15].

Figure 7.7.6

Use of Data files in constructing Module List



7.8 Notes

1) Although the use of prototypes is mentioned at a number of points, the process of prototyping was considered beyond the scope of the book. (IBM 1992, p100)
2) When a subject is changed to Standard grade from Ordinary grade, both versions are offered for a few years to allow pupils who had already started Ordinary grade courses to complete them. This also allows staff time to prepare course material for the new subject.
3) The compulsory modules contained in the COLLEGE file are not added - this simply involves inserting two or three modules into the Modules List at the start.
4) Each block in the Course Grid is sub-divided into two columns, so that half-modules can be shown correctly.
5) It will observed that the condition which terminates the REPEAT loop has been placed at the end of the loop. In these instructions it is assumed

that the REPEAT loop operates as a 'post-test' loop - the condition that terminates the loop is tested after the sequence of instructions within the loop has been performed. The sequence is always executed once. This is in accordance with Pascal practice :-

"The loop must be entered at least once because the Boolean expression is not evaluated until after the loop body has been executed."

(Naps 1992, p214)

6) The entries in the SUPERCLASS file can be thought of variously as courses, subjects, jobs or hobbies. The phrase 'job code' will be used throughout to refer to any of these meanings.
7) In a sequential file records take only only as much space as they need. The records have to be read from the start. In a random access file the records all have the same length. The computer can jump into the file at the start of any record.
8) In BBC Basic the suffix % indicates that the variable is an integer, and $ indicates a string variable (a string of alphanumeric characters). A variable name without a suffix is a floating point number.
9) Sommerville suggests that certain requirements, such as speed of operation, can be relaxed for a prototype (Sommerville 1989 p116).
10) This facility was not implemented in the prototype.
11) At present reference numbers are given in two forms; the old system used five digit numbers, but these are being replaced by the new seven digit references. This was discussed in Chapter 4.
12) SCE subject names are contained in the SCE-CODES file, described earlier in this chapter.
13) One code for Higher, one for Ordinary and seven for Standard grade.
14) The 'user' of this file manager will be the person responsible for maintaining the MODULES file. This should not be confused with the ultimate user of the overall system, who will be a student.
15) In structure charts the square rectangles indicate processes, while he round edged rectangles represent data sources (Sommerville 1989, p237).

Chapter 8
Evaluating the Prototype

8.1 Validation and Verification

The evaluation of a program must be considered as two distinct processes, validation and verification. Sommerville gives the following quotation from Boem (1979);

Validation: Are we building the right product?
Verification: Are we building the product right?

(Sommerville 1989 p406)

Verification implies checking that the program does what the programmer intended it to, whereas validation involves checking that the programmer knew what the 'customer' wanted.

In this project, the 'customer' is ultimately the authors of the Action Plan, insofar as it related to choice and negotiation. The higher objectives of the Action Plan were that more students would involve themselves in further education, and that further education would itself be more flexible, would provide better opportunities for students to fit themselves for the more changeable demands of the future, and in so doing would help students develop adult responsibility. Providing choice and therefore allowing more opportunities for negotiation, the factors with which this work is concerned, were seen as tools in the process of achieving these higher objectives.
In this context validation is therefore seen as the process of finding out whether the prototype does provide choice and allow negotiation, while still conforming to the principles of course design; principles which were also laid down in the Action Plan.

8.2 Program Faults

Faults in a program vary considerably in cause and effect. The lowest level of fault may be a mistake in syntax. For example, in BASIC the command to print a word may be abbreviated to 'P.'. Omitting the '.', or using the lower case 'p' would both result in program failure; this kind of syntactical mistake is easily detected because the program fails when an attempt is made to execute the command. On the BBC Master, the error message "Mistake at line ..." appears on the screen, the line number indicating precisely where the mistake can be found.

Other typographical errors can be more difficult to identify; for example, "number=4" and "number%=3" are both acceptable statements(1), but will produce incorrect results if the programmer forgets which one is which; as far as the computer is concerned, number and number% are two quite different names.

At a higher level, logical errors can sometimes produce results which are obviously wrong, but are more difficult to correct. As an example, consider a routine which sorts names alphabetically. The programmer's first attempt may well produce the sequence, "Mabbot, MacGregor, Madox, McGregor". This is alphabetically correct, but may not be what the programmer intended; using the definitions given above in 8.1, this constitutes a failure to verify the program. Alternatively, it may not have been the customer's intention, but he failed to communicate that to the programmer, in which case it would be a validation fault.

The most difficult errors to deal with are those that do not cause the program to crash, and do not produce obviously incorrect results.(2)

The word 'fault' is thus used in a very general sense; anything which causes the program to produce invalid results is a fault. This raises the question, "Can a program contain a fault which does not produce an invalid result?". If it can be said that an invalid result will <u>never</u> result from the fault, then it is difficult to imagine what kind of fault this could be; it is even more difficult to imagine why it should be called a fault. However, there are three ways in which the statement, "It contains a fault but the results are correct.", can be true.

a) The fault only gives invalid results under certain conditions. The name-sorting routine referred to above only gives incorrect results if the list of names includes Mac's and/or Mc's. Identifying the fault relies on testing the program with the full range of possible data; the program must be tested under all of the conditions under which it is supposed to operate.

b) The results may only be 'correct' in that the tester did not recognise that they were incorrect. In some such cases the user will eventually recognise that the results are incorrect and refer the program back to the programmer. In other cases this may be unlikely. For example, when modules are fitted into the Course Grid space may not be found for some; this is inevitable if the sum of the module lengths exceeds forty-eight, and is possible even with smaller numbers. The tester, seeing a module labelled 'not fitted', might not realise that it should have been possible to fit it. In 8.5, two errors in this part of the program will be described. These errors were only noticed because the result was very noticeably wrong. It is quite possible that incorrect results had been obtained in earlier tests, but had escaped the tester's attention; and if he, while specifically testing the program, did not notice them, is it likely that a student using the program would? The only consolation is that such a fault does not render the program useless; the courses it generates are all correct, but some additional possibilities are rejected unnecessarily.

c) The third type of fault is not a fault of the program, but of the data it manipulates. If a module has an entry requirement, but that is not entered in the module record, the program may recommend the module without the entry requirement. Although this is a data fault not a program fault, as far as the user is concerned the program gave an invalid result, and therefore, the example has been included here.

The conclusions which can be drawn from these examples are;

a) The program should be tested under all likely conditions;

"Testing involves exercising the program using data which is similar to the real data on which the program is designed to execute."

(Sommerville 1989, p406)

The possibility of testing using all possible cases is dismissed as impracticable. Marcoty, for example, discussing exhaustive testing cites an example of testing a relatively simple program using all possible input values, and concludes that it would take 500 000 years to complete. (Marcoty 1991 p150)

b) The tester must be extremely careful to check the output from tests;

Testing is a dynamic technique. It involves running the program, observing its inputs and outputs, and looking for unexpected behaviour.

(Sommerville 1989, p406)

"6. There is no point in testing if the output is not examined."

(Marcoty 1991, p291

This implies that the tester not only knows what behaviour may be termed unexpected; he must also be able to recognise it when it occurs. This is particularly difficult when incorrect behaviour is a sub-set of correct behaviour. In the example discussed later, it is normal for the program to fail to fit some modules into the Course Grid; it is only when it should have been possible to fit a particular module that this becomes a symptom of a fault.

c) The output can only be as accurate as the data supplied.

d) Testing is concerned with both the verification and the validation of the program. Because an error is anything that causes the program to produce an invalid result, failure to meet the customer's needs, even when the program is correctly coded, is nevertheless a fault, just as much as when it fails to do what the programmer intended.

8.3 Verifying the Prototype : Testing

Sommerville distinguishes between 'testing' a program, the process of establishing the presence of faults, and 'debugging' the program, in which the fault is found and removed. (Sommerville 1989, p405)

On first reading of this distinction, it was felt to be unnecessary; what would be the point in finding a fault and not doing anything about it? It is now realised that the processes are distinct; it is known that the prototype contains faults, but these faults have not been removed. In general, however, both processes were carried out as a sequence, with debugging routinely following the discovery of an inappropriate result.

Initial testing was of the 'benevolent' variety (3), i.e. it was carried out by the author. Testing of the Selector was carried out by operating the program as a user would, using sample data; hence the need to prepare relevant data files. The sample data is described below, in 8.10. When a procedure was added to the program, it was tested. In general, this was done in the context of the whole program (4). Where the procedure did not itself produce a recognisable output, additional procedures or lines of code were inserted to give an indication of the performance of the procedure under test. The approach initially adopted for the Sequencer was slightly different. It was written separately from the Selector. The output from the Selector was simulated by writing a fixed block of module data as a 'front end' to the Sequencer (5). The operation of the procedures forming the Sequencer were then tested using that data until the program operated correctly. This was done for three reasons;

a) It avoided the need to work through the Selector process before reaching the Sequencer procedures.
b) It allowed the Selector and Sequencer programs to be developed 'simultaneously'. This proved to be particularly useful, as faults were found, not in the coding, but in the specification for the output from the Selector. It was found that this output did not contain enough information for the Sequencer to work; to be more accurate, it was found to be impossible to design the Sequencer to operate from the original Selector output. This was a consequence of not having a detailed plan of the former before starting to code the latter. As the output from the Selector was being simulated by the block of data, it was relatively easy to change the data until it contained the relevant information. When that was found to give satisfactory results the Selector was then changed to give the required input to the Sequencer.
c) Because a fixed block of data was being used it was possible to carry out the laborious task of sequencing it by hand. The difficulty of recognising

faulty performance has been explained, in 8.2(b). One way of allowing incorrect output to be recognised is to produce correct output and compare it with the program output; in this case, the only way to produce correct output was to do it manually. This was a tedious task which nevertheless required concentration to avoid making mistakes. By using fixed data as an input, the task had to be carried out only once.

In the final stages of program development, the Selector and Sequencer were used together, the former producing the input to the latter. This allowed the operation of both parts to be checked using different sets of modules. This part of the testing process identified some unwanted interactions between the two; these were caused by inappropriate naming of variables. For example, in the Selector a variable might have been called 'nmods', whereas in the Sequencer the same variable was called 'nmod'; as the computer does not recognise the programmer's intentions, values would not be passed on correctly. A similar problem arose when two different variables were inadvertently given the same name. This kind of problem could have been avoided by better planning, and by the use of 'local variables' in procedures; defining a variable as local to a procedure allows the same name to be used for different purposes, with each being treated separately. Similarly, 'parameter passing' allows the same variable to have different names in different procedures, the value of one being passed to the other automatically. This was one of the areas where full use was not made of the facilities available, and has already been acknowledged as a mistake.(6)

8.4 The faults

8.4.1 Fatal errors in coding, those that caused the program to crash, were corrected immediately; testing the program could not proceed until the program could operate.

8.4.2 Two identified faults in the Selector, and the way in which they were corrected, will be described in 8.9.2, below. One of these, being concerned with the way modules are split, has educational significance. The other imposes a limit on the fitting of modules into the Course Grid.

8.4.3 Another fault in the program is imposed by the limitations of the computer memory; if an attempt is made to add more than forty-eight modules to the module list, the program will crash.(7) The size of the array holding the module information was limited by the amount of memory available. No protection is written into the program to prevent the user 'overloading' the Module List, but it is not difficult to do so. Forty-eight was chosen as the maximum number because this allowed for the worst case, where forty-eight half modules were identified. If that principle is accepted, the module list must be capable of holding more than forty-eight modules. This is to allow for the forty-eighth module having an entry requirement, which must also be included.

8.4.4 Having modified the Selector program to allow student responses to be recorded (to be described later), it was found that certain combinations of modules over-ran the computer memory. These combinations all involve deleting modules from sequences which are considerably too long to fit in the Course Grid. It is thought that the fault lies in the coding of the 'count branches' routine, the most probable cause being a 'non-terminating loop', which if called repeatedly uses up large amounts of memory. Fortunately, the program can cope with some examples of long sequences, as will be demonstrated by the sample output to be described in 8.11.

8.5 Validating the Prototype

Because the program was tested using real data, validation was carried out at the same time as verification; this is referred to as acceptance testing (Sommerville 1989 p409). In checking the output for mistakes, no distinction was made between program errors and 'specification' errors, and, within the limitations of the prototype objectives, where it was realised that the principles of course design were being violated, corrections were made. 'Alpha testing' is similar to acceptance testing, except that it is carried out in the presence of the 'customer', who can verify that the output is appropriate or point out where it is not (Ibid.).

8.6 The User Interface

One part of the system which has received only passing consideration in the text is the interface between the user and the computer, i.e. the way in which the computer conveys information and instructions to the user, and the way in which the user communicates his responses to the computer. On the one hand, the interface is a vital component of a system; a badly designed interface can render a program unusable, or can deter the user from using it to its full extent. Thus;

> *It is now generally recognised that the user interface of a system is the yardstick by which that system is judged. An interface which is difficult to use will at best result in a high level of user errors. At worst, it will cause the software system to be discarded, irrespective of the functionality that it offers.* (Sommerville 1989, p258)

On the other hand, designing a good interface requires specialist knowledge, and cannot be done in isolation.

> *The design of a user interface should not be undertaken by the software engineer alone. It is essential to consult with system users and discuss their background and needs.* (ibid.)

It is unlikely that a full implementation would use a computer as restricted as the BBC Master.

> *It is now possible to assume that the user's terminal will include a significant amount of processing power and it is increasingly likely to have a bit-mapped high-resolution display supporting multiple text fonts and mixed text and graphic displays.* (Ibid.)

Sommerville expresses the opinion that text-only interfaces will be completely supplanted by the pictorial, user-friendly environment offered by the use of windows, icons, menus and pointing (the WIMP environment). This being the case, it seems unlikely that an interface designed on the BBC Master would offer anything of value to a designer of the proposed system.

However, in order to test the system with users, it was necessary to design a working interface. In addition, when considering the tasks that would be performed by the student in using the system certain questions had arisen. The main entry point to the Selector is through the SUPERCLASS codes, referred to in the text as job 'codes'. One question is, "Can students use the SUPERCLASS system to find the required job codes?" This might be extended to, "How much training must students be given in the SUPERCLASS system?" If one of the objectives of the computer system is to reduce the amount of staff time required to generate modular courses (8), then the value of the system will decrease as the amount of training time increases.

Thus, reasons for testing the system with users were

a) to check that it operated correctly under 'real life' conditions with real users
b) to find out more about the target group
c) to answer such questions as, "How long does it take to generate a course?" (9)

In designing the interface, the principles given in Shneiderman (1987, pp312 - 325) were followed.

a) Instructions are given on the screen
b) when responses are required of the student the valid responses are shown
c) when errors are detected the valid responses are shown again.
d) The students, as novice users, are required to make minimal responses, i.e. responses are, as far as possible, single key presses. (ibid. p72)

Shneiderman gives eight "golden rules" of dialogue design: (ibid. p 61)

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialogues to yield closure - sequences should have a beginning, middle and end.
5. Offer simple error handling.

6. Permit easy reversal of actions.
7. Support internal locus of control (avoid surprising system actions or difficulty in obtaining information. The user should feel in control of the system.)
8. Reduce short-term memory load.

In the first program "Construct the Pre-entry List" the rules were applied as follows:-

1) Where a response is required, the question or instruction demanding the response is shown in inverse video. However, some important pieces of information are also shown in the same way. For maximum consistency these items should be highlighted some other way, e.g. different colours
2) It is not anticipated that students will become frequent users. Within the program it might be useful to bypass the confirmation of subject, and the sequence of questions/responses that detail the SCE subject, exam, grade (described later).
3) Feedback is somewhat minimal. Brief messages are given when the disc drive is operating, to tell the student that the computer is working. Some of the messages are 'unfriendly', e.g. "That level is not possible. Press any key to continue."
4) A number of sequences are required in the program; for example, the student has to supply the name of a subject, the examination, and the grade awarded (or expected). Having supplied one of these components he is lead to the next. He cannot bypass any, which ensures that all data supplied is complete.
5) In this program some errors are easy to identify. In highlighting an SCE subject name, only the arrow keys and RETURN are 'entries'. For the examination level, only H, O and S are valid, and for the grade of award only numbers 1 to 7 (for O- and S-grade) or letters A to E (for H-grade) are valid. In confirming the choice of subject, only the letters Y and N can be used. Correct key presses are signalled by a relatively quiet 'bleep', whereas invalid responses are noted by a lower, longer noise. The question is then repeated, with the valid responses shown.

However, other errors are more difficult to identify. The computer cannot check the student's name or number (10). Module reference numbers are not checked, so errors will pass unnoticed (11).

6) This has not been implemented. It had been assumed that, as the student cannot omit parts of a sequence and cannot leave an invalid response uncorrected, that the information he supplied could not be incorrect when he exited from the system. This assumption was incorrect, as will be explained later.
7) Some actions of the program are probably unexpected, but they are explained; for example, when the program pauses to open a file to hold the student information, the student is told that that is what is happening. As will be described later, some of the procedures built in as part of the testing process caused unpredictable behaviour from the computer - these were cured before the program was tested. Some procedures were built in to give the student control over certain features of the program.
8) In the initial version the student had to keep note of the SCE subjects he had already entered. This was changed to allow the student to see this on screen.

8.7 The student testers

The prototypes were tested by a group of fifteen students chosen from the target group. They used the system just after they had completed their Standard grade examinations, in S4. They were at the stage when they might have been considering going to college rather than staying on at school. They did not know the results of their examinations. This, of course, is precisely the situation when students are interviewed prior to attending a college. They all 'knew about' the National Certificate, as they had all taken some (typically four) SCOTVEC modules. To say that they were 'familiar' with the National Certificate would be an optimistic assumption - they did not, for example, have any idea just how many modules were currently available. Most of them were more familiar with the phrase 'SCOTVEC modules' than with 'the National Certificate'.

The policy adopted up to this point has been to describe the system as it is, leaving discussion of possible improvements to Chapter 9. However, in this section the faults or observations will be reported and suggested improvements described immediately. This will reduce the need to repeat, in Chapter 9, the description of the fault in order to explain the suggested improvement.

8.8 Testing the program to Construct the Pre-entry List

As was stated in Chapter 7.4.1, this part of the system was constructed to introduce the students to the system via an easy task.

8.8.1 The first test

For the first test a student was asked to 'try out' the system. She was told that she had to tell the computer which SCE examinations and modules she had taken, and that she should do this by following the instructions on the screen. No written instructions were given. A number of alterations were made to the interface after this first test. T.K. Landauer (quoted in Carroll 1991, p69) gives the following information :-

> *"So far, practical experience, as well as some experimental evidence, suggests that tests with 2 to 10 users usually reveal most glaring flaws, and sometimes offer strong guidance for positive improvements."*

Useful information was gained from the first student to test the system. The faults, and the author's response to them, are listed below.

1) The student was not sure whether to type her name and number in response to the first prompt (Name?), and did not realise she had to press RETURN at the end of **each**, i.e. name and number were to be typed separately. (This is not surprising - she had not been told to!)

The confusion was caused by the first line of text, which said "Please type your name and student number.", and then prompted, "Name?". This was rewritten as shown below :-

$\boxed{\text{Please type your name.}}$
Type your whole name on one line.
Press the RETURN key when you have finished.

Name?

followed by

$\boxed{\text{Please type your student number.}}$

Number?

The normal screen is printed with white text on a blue background. 'Inverse video', i.e. blue text on a white background will be shown $\boxed{\text{in a solid box}}$ (12). To avoid any confusion, screen displays will be surrounded by a dotted box. Shneiderman (1987 p71) recommends a maximum of four colours, with additional colours occasionally used to attract attention. The BBC Master can be operated in various 'modes', which determine the size of the characters and the number of colours which can be shown on screen. One mode allows up to sixteen colours (13) on screen simultaneously, but the characters are large, with the result that very little information can be shown at any time. The mode used for all the prototypes allows two colours at a time (the pair to be selected can be chosen from the sixteen available.) Fortunately, this is not seen as a disadvantage.

> *"Monochrome displays should be seriously considered as the primary format because approximately 8 per cent of males in European and North American communities have some form of color blindness."*
>
> (Shneiderman 1987 p339)

He further recommends that the initial design of the interface should be in monochrome (ibid.). His recommendation on the choice of colours for the monochrome display was not followed.

"Black on blue and blue on white were two colors with low error rates....."

(Shneiderman 1987 p341)

This author uses white on blue for all programming, because he finds a predominantly white screen too bright for sustained use. (Turning the brilliance down reduces the readability of text. This is probably a personal matter, but see later.)

This student, in common with all of the students tested, did not have to be told to press the RETURN key after such an entry had been typed (14). This does not contradict the statement above; in that case she did not know she had to **separate** the name and number with a RETURN. However, the instruction to use the RETURN key was included at all appropriate places so the student would be in no doubt that that was what was required.

2) The keyboard repeat rate had been increased for programming (it allows the user to move through a screenful of text quickly when editing programs). In responding to one question, the student held the key down for too long, with the result that she answered the next question as well. As the answer she gave was invalid she was immediately given an error message, but did not understand what she had done wrong. The second message had appeared and been incorrectly answered before she had seen the question on screen.

Choosing a successful value for the keyboard repeat rate was thought to be problematic. For some students it would be too slow, causing frustration, while for others it would be too fast, causing the kind of error just described. It was felt to be more useful to prevent the keys repeating at all. Accordingly, an instruction was written into the program to prevent the keyboard auto-repeating when a key is held down.

3) After correctly entering a number of SCE subjects she forgot how many she had entered. The list was displayed, but not when the SCE subject list was on screen.

The screen display was changed so that the subjects already entered are displayed on the right side, while the list of SCE subjects is displayed down the left. While selecting the next subject to enter, the student can see which have already been entered.

4) In response to the question, "How many modules have you passed?", she

gave the reference number of the first module - implying that she had passed 12,345 modules! The program would have quite happily prompted for the reference numbers of these until the array size was exceeded. The only way out of that situation was to press the ESCAPE key, which stopped the program running (and lost all the data which had been entered.).

By changing the program slightly it was possible to remove the question. Instead of stating the number of modules to be entered, and then being prompted for each one, the student is told to enter module reference numbers or 'Q' when there are no more to be entered.

5) The student complained that she did not like the white text on blue background, finding this more difficult to read that white on black. (See the comment on personal preference for white text on a blue background.)

In line with Shneiderman's recommendation that the choice of colour should be under user control (1987, p339), a procedure was written into the program to allow the student to change the colours used for display. The procedure was extended to allow the student to control the screen 'interlace'. Without going into the technicalities, on some BBC systems the output from the computer does not synchronise correctly with the VDU. This results in blurring or 'wobbling' of the display. Commands were included in the program to allow the student to obtain the clearest picture.

6) Although not commented on by the student, the author, perhaps beginning to see the display through the student's eyes, felt that the SCE names and, later, the module list accumulating at the side of the screen were not very clear.

The display was altered so that this information was moved away from the edge of the screen by one character.

8.8.2 Testing with other students

Having made the alterations described above the program was tested by another student. After a few minutes it was observed that she had listed a surprisingly large number of SCE subjects. It transpired that she had ignored or not read the information that SCE subjects would be dealt with first, and was indicating both SCE subjects and modules using the SCE subject list. Some modules have the same name as an SCE subject, and where there was no direct match she was choosing the nearest name, on the assumption that she did not know the 'official' name of the module. Recognising that a module is not a Higher or Standard grade subject she was marking all the modules as Ordinary grades. Modules are not graded 1 to 7 or A to E, but she was entering grades on the same basis as for Standard grades.

As a result of this observation, it was decided that the distinction between SCE subjects and modules must be made much more clear. A set of introductory notes was written (on paper). These give information about the purpose of the system and how to operate it, and are included in Appendix 9. The notes instruct the student to complete a 'Qualifications Form'. An abbreviated copy of this is given below.

Figure 8.8.2

The Qualifications Form

Qualifications Form

Name :- ______________________ Number :- ________

	SCE subject name	**Exam**	**Grade**
1	______________________	___	___
2	______________________	___	___
etc.			

Give the name and reference number of any SCOTVEC modules you have completed.

	SCOTVEC module name	**Reference**
1	______________________	________
2	______________________	________
etc.		

8.8.3 Before further testing was performed, it was necessary to modify the program to record the student responses and the times taken to respond. Three methods of achieving this were tried, the third being the method used in the program supplied on the disc.

Method 1 had the computer printing the time at which the student responded and the keys that the student used. Because anything sent to the printer is also displayed on the screen, some of the printed text caused confusion by overwriting the correct prompts. It is possible to overcome this problem, but only with additional programming. In addition, it was felt that the constant clattering of the printer was off-putting to the student. Before improving on this method, by additional programming, it was decided to try a different approach.

Method 2 was similar to Method 1 except that the times and responses were written to the disc in a file created for the purpose. Writing to the disc does not interfere with the screen display. However, when an instruction is given to write information to the disc, the information is stored in a buffer. Repeated instructions to write are fed into the buffer until it is full. Only at this point is the information actually recorded on the disc. The point at which the buffer became full depended on the previous responses of the student, but always followed immediately on a response. The operation of the disc drive stops the program for a short time. It was found that the program would pause, unpredictably, while the disc drive ran, and this caused problems. For example, in one test the program paused just after the student had typed a response. As the computer did not appear to respond to this input, the student typed it again. This gave a similar result to that described in (2) above, the computer indicating an invalid response which the student felt was correct.

Method 3 had the student's responses and response times written into an array in memory. This does not spoil the screen display and is virtually instantaneous, so there are no pauses. There are no drawbacks from the student's point of view - he is not aware of it happening. In terms of programming, on the BBC it is necessary to state the maximum size of an array before it can be used. The array has to be big enough to hold a record of every response the student is likely to make, but small enough to fit in the computer's memory (along with the program and all the variables etc.). The number of responses made by the student depends on a number of factors, e.g. how many subjects he names, how many mistakes he makes. It is not therefore possible to state the actual size of the array in advance. By this time however, Methods 1 and 2 had been used. These showed that it was unlikely that the number of messages to be recorded would exceed 200. (The recorded messages include student responses and certain operations of the program as a result of these responses - the student does not make anywhere near 200 responses of his own.) The array was therefore dimensioned to hold 200 possible messages. To cope with the (unlikely) case of more being required, the procedure for recording the messages was designed to print the first 200 messages and then start again filling up the array. While printing took place the student would be prevented from continuing, but it was felt that this was not unreasonable. If more than 200 messages were considered unlikely, more than 400 were considered impossible. Printing during the student's session would therefore only occur once, the rest of the messages being printed after the student had finished. This method proved to be much more satisfactory than the other two, although they served a purpose in establishing that 200 messages was unlikely.

As they were completing the Qualifications Form the students were given a five digit number, to simulate a student number.
They were then asked, in turn, to enter the information into the computer by following the on-screen instructions. While they were waiting they were asked to continue reading the introductory notes, which gives information about the SUPERCLASS system and how to use it.

8.8.4 Observations during the tests

8.8.4.1 Student's listings of subjects and modules

The SCE subjects are listed in alphabetical order. One of the minor questions in the author's mind was, "Is there a more useful way of sorting the subjects?" If pupils are asked to list their subjects very few will give them in alphabetical order - can the subjects be listed in an order that matches more closely that used by the pupils?

The evidence appears to be unclear. Shneiderman gives three examples of work in this area :-

a) In a comparison of semantic versus alphabetic organization of menus, there was no significant difference. (Shneiderman 1987, p102)

b) When locating words in lists, alphabetic is faster than functional groupings. (ibid. p106)

c) Quoting McDonald (1983)

> *"These results demonstrate the superiority of a categorical menu organization over a purely alphabetical organization ...."*
>
> (Shneiderman 1987 p88)

Inspection of the Qualifications Forms returned by the test group showed no common criteria used for grouping the SCE subjects, although students tended to begin with Mathematics and English and then form the following groups :-

Maths + English
Physics Chemistry Biology
French German
Modern Studies Geography History
OIS Computing Accountancy and Finance

However, the groupings were not rigid, all students did not use the same groupings and the groups were in variable order. It was concluded that it was unlikely that any significant gain would result from arranging the SCE subjects in an order other than alphabetical.

One obvious improvement to the screen display would be to type the subject names in lower case, as it is well known that this is easier to read.

There was no discernible pattern to the order in which modules were listed. Very few students knew the reference numbers of the modules they had taken, which is not surprising as these numbers are used infrequently, if at all, by pupils. This would suggest that students would have to be given advance notice of the need to use reference numbers, so that they could find out what they were (15).

8.8.4.2 Screen colours

It was found that very few pupils tried to change the screen colours. Those who did experiment returned them to the original. It would appear that students are happy to use the white text on blue. With more modern machines black text on white background should be quite satisfactory (16). If a colour monitor is used the option to change colour should be given, in keeping with the principle of 'putting the user in control'.

8.8.4.3 Screen interlace

A single key is used to 'toggle' the interlace on and off. Some students used the left arrow (correctly) to toggle on, but then used the right arrow (incorrectly) to toggle it off. This is probably more natural and, if the system were to be implemented on a computer with a similar requirement (unlikely?), should be allowed.

Most students did not use this option. In fact, on the machine used for the tests, the best picture was already on screen when the pupils were presented with the possibility of changing it.

8.8.4.4 Selecting the SCE subject and grade

The student uses the up/down arrow keys to highlight the required subject name, and selects it by pressing RETURN. He is then presented with the screen display shown below, in Figure 8.8.4.4.1.

Figure 8.8.4.4.1

Screen Display for confirming the Subject Name

Only the grades shown below are possible.

If the grade you passed is not shown, you may
have chosen the wrong name for the subject.
This part of the program does not deal with modules.

Subject	Grades	
ACCOUNTING	Higher	Ordinary

Is this the correct subject?

Tap Y (for Yes) or N for No.

Some subjects have different names at Higher grade and Standard grade; for example, Art is only offered at Standard grade, while Art and Design is offered at Higher grade. Accounting, shown in the display, is not offered at Standard grade. To allow for a student being unsure as to the correct name, he is shown the subject name and the possible levels at which it could have been taken, and then asked to confirm that this is the subject he intended to choose.

It was observed that a number of students ignored the warning and responded 'Yes' even though the required level was not shown. They then found that they could not indicate the required level because that was not possible. This mistake was more likely to occur after the student had entered a few subjects and grades. Inspection of the times taken to complete the transactions showed that the first one was slow, but the speed increased thereafter. The students rapidly learned the sequence of questions and automatically responded "Yes" without actually checking the grades available. Given that the subjects likely to cause confusion can be anticipated, a better dialogue, following the selection of the subject name, would be as shown below in Figure 8.8.4.4.2.

Figure 8.8.4.4.2

Improved Screen Display for Confirming the Subject Name

Subject :- Art and Design

What exam did you sit :- Higher, Ordinary or Standard grade?

Type H, O or S

Student response > *S*

Art and Design was not offered at Standard grade.

Do you mean Standard grade Art?

Type Y (for Yes) or N (for No).

Perhaps even better would be to ask the student to group SCE subjects under Standard grade etc and then only display the possible subjects.

8.8.5 Error rates and response times

Figures relating to the mistakes made and the time taken taken to reach various points in the program are given in Table 8.8.5, below.

Table 8.8.5

Errors and Response times for Constructing the Pre-entry List

Student	Colour	Interlace	Invalid screen	Invalid elsewhere	SCE	Modules	Time 1 GetSCE	Time 2 Stopped SCE	Time 3 Finished	Time 3 (mins)
1	N	Y	2	3	8	5	128	398	505	8.42
2	N	Y	0	0	8	5	62	165	251	4.18
3	N	Y	1	1	8	3	140	426	556	9.27
4	N	Y	1	4	8	5	110	497	593	9.88
5	Y	Y	0	1	0	6	182	187	312	5.20
6	N	N	0	0	8	1	99	379	482	8.03
7	N	Y	0	0	8	6	99	207	300	5.00
8	N	N	0	5	8	5	90	489	628	10.47
9	N	N	0	5	8	5	87	326	425	7.08
10	N	N	0	1	8	5	103	316	434	7.23
11	N	N	0	4	8	5	34	187	253	4.22
12	N	Y	0	1	0	6	134	150	284	4.73
13	N	N	0	6	8	3	43	268	361	6.02
14	N	N	0	10	8	2	103	331	418	6.97
15	N	N	0	5	8	3	51	226	297	4.95
Av	1	7	0.3	3.1	6.9	4.3	97.7	303.5	406.6	6.8

Time 1 = the time taken to enter name etc. up to the start of entering SCE names.
Time 2 = the total time taken up to finish entering SCE names.
Time 3 = the total time taken to use the program.

Only one student changed the colour of the display (and he put it back to white text on blue). About half changed the interlace, and of those about half changed it back again incorrectly. This was referred to in 8.8.4.3. These errors have been separated from those occurring elsewhere in the procedure. Other mistakes have been detailed above. On average the students made 3 errors, although this figure is inflated by the 10 errors made by one student, who frequently failde to confirm that the correct subject had been chosen. One of his errors was in typing 'T' instead of 'Y'. Given that these are adjacent on the keyboard it is probable that he was trying to type too quickly - given that he made these errors and still managed to complete the process in the 'average' time, this is a reasonably assumption.
The mean time to operate the program was about seven minutes, with the longest taking nearly eleven.

8.9 Testing the Selector program

8.9.1 Author testing

The program had been tested as it was being coded (see earlier comments). A number of tests were run on what was thought to be the final version. These tests consisted of supplying job codes which were known to have particular effects, i.e. job codes which

a) yielded no modules
b) yielded a large number of modules, either directly or because of the need for entry requirements
c) generated a long sequence of modules.

A number of errors were found in the data files. These included :-

a) a job code in the module record with no SUPERCLASS name
b) an incorrect module reference number in the MODULES file; the correct number was used in the JOBS-MODS file, so the computer searched for a record which, apparently, did not exist.
c) a module reference had been updated, but another module used the original as an entry requirement.

These faults are all likely to happen when a number of files must contain matching data, but each file can be edited separately from the others. These 'operator' errors had been anticipated and error-handling routines included, but in case (b) it was found that the error was not handled correctly. It was decided that rather than to improve the error-handling it would be preferable to remove the errors.,

Checking programs were therefore written to ensure that :-

a) any job code used in the MODULES file and JOBS-MODS file had a corresponding name in the SUPERCLASS file.
b) any module reference number in the JOBS-MODS file had a corresponding record in the MODULES file.
c) any module reference number given as an entry requirement in a module record had a corresponding record.

The only error that can still occur happens when a student enters a job-code for which there is no entry in the JOBS-MODS file. The error-handling routine, which simply points out that the code cannot be identified and asks the student to re-type it, works successfully.

8.9.2 Before leaving this discussion of 'author' testing, it is worth looking at part of the output from an earlier version of the program; this output cannot now be generated from the program, as the program was improved following these 'discoveries'.

The output described below was obtained during the evaluation of the program. The program had already been used on numerous occasions and had performed satisfactorily; this was intended to be the last trial before testing the program with students.

8.9.2.1 Five job codes had been supplied and the program had identified ten suitable modules, totalling twenty-two units in length. This being approximately half-way to a full course the results were printed, so that an intermediate stage in the process could be included in this text.

The Course Grid obtained was as shown in Figure 8.9.2.1, below.

Figure 8.9.2.1

Part of the Output from an Early Version of the Program (A)

	1	2	3	4	5	6
1	7010020	7010020	7010030	7010030	6170021A	6170031A
2	6171011	6171011	7110031A	6171021	6170021B	6170031B
3	7110330	7110330	7110031B	7110041A	6171021	7110051A
4	-------	-------	-------	7110041B	7110051A	7110051B
5	-------	-------	-------	-------	7110051B	-------
6	-------	-------	-------	-------	-------	-------
7	-------	-------	-------	-------	-------	-------
8	-------	-------	-------	-------	-------	-------

The suffixes A and B indicate that a module has been split - instead of occupying two columns (for a normal length module) as for 701002, a module which has been split occupies two rows. 7110051 (columns 5 and 6) represents a double length module, which has been split to occupy two rows and two columns. It will be observed that where modules have been split, this tends to occur towards the end of the sequence. It had been decided that it would be better to split modules starting at the end of a sequence because;

a) this would allow students to 'settle in' at the college before exposing them to the more concentrated and less varied study implied in splitting modules.

b) concentrating on a few modules, a consequence of splitting modules, 'puts all their eggs in one basket'; any module becomes a larger proportion of the work done when fewer modules are studied, and failure to complete one would affect a larger part of the course to follow. It was considered preferable to delay this risk until the student had had the opportunity to complete a larger number of modules.

8.9.2.2 After printing the output (of which the Course Grid is a part), another job code was supplied. The five previous codes had only yielded ten modules, an unusually small number because the jobs concerned were all closely related (one job code, for example, had added nothing to the course as all the relevant modules were already in the module list). It was decided to deliberately use a job code that would add a relatively large number of modules to the course, and, being less closely related to those already used, was unlikely to suggest modules already in the list. The extra job code resulted in eight modules being added to the list. Figure 8.9.2.2 shows the Course Grid that resulted.

Figure 8.9.2.2

Part of the Output from an Early Version of the Program (B)

```
        1         2        3         4         5         6
1 7010020A  7010030  7010030   6170021A  6170031A  -------
2 7010020B  7110031A 6171021   6170021B  6170031B  -------
3 6171011   7110031B 7110041A  6171021   7110051A  -------
4 7110330   6180011A 7110041B  7110051A  7110051B  -------
5 6170061   6180011B 6170011A  7110051B  -------   -------
6 -------   6171011  6170011B  6170011A  -------   -------
7 -------   7110330  6180021   6170011B  -------   -------
8 -------   6170061  6171031   6171031   -------   -------

These modules could not be fitted into the grid.
6180031  Text Processing 2
6180041  Text Processing 3
6180051  Text Processing 4
```

In previous tests the program had been unable to fit some modules into the grid, so that part was not considered unusual. What was startling was that column six was not used at all. After considering the Entry List and the sequences now involved, it was realised that the drastic change in the Course Grid was caused by the addition of a long sequence of modules to the Module List. In order to fit this sequence the program had split almost every module in it; and the first module in the sequence was 7010020, which was also the first module in a number of other sequences which had been included before. A simplified example will help to make this clear. One of the earlier sequences may be represented as AABBCCDD (four modules, all of the normal length, therefore occupying two columns each). To fit this sequence into six columns, modules C and D must be split, giving

```
AABBCD .
    CD
```

The sequence added later may be represented as AAEEFFGGHHJJ. When

these modules are split they give AEFGHJ .
AEFGHJ

Because module A has now been split, the earlier sequence is changed to

ABBCD
A   CD

This sequence now occupies only five columns; C or D has been split unnecessarily. One disadvantage of this is that column six is underused. Because earlier columns are filled more quickly, there is less chance of being able to fit sequences into the Course Grid. A single isolated column can only be used for half modules or split, normal length modules, which reduces the usefulness of the column. The other disadvantage is that split modules themselves may be educationally 'suspect; creating split modules unnecessarily cannot be justified at all.

It was realised that this was a consequence of the decision to split modules from the end of the sequence rather than from the start. This was not an error; the program had operated correctly as it was supposed to, but had had an unforeseen and undesirable consequence.

The second problem was that by the time the new longer sequence was being fitted all the cells in columns 2, 3 and 4 were already occupied and most of the sequence could not be fitted - with the result that the sequence that had caused the trouble was not used! This was partly a result of the unnecessary splitting of modules filling up columns too quickly (as explained above), but inspection of the flexibilities of the modules showed that the least flexible sequence had been added to the Course Grid last instead of first. Re-examination of the program revealed that it had been implicitly assumed that all members of a sequence would have the same flexibility; in particular, if the first member had zero flexibility then all members would also have zero flexibility. This is not necessarily so. Again, an example may clarify this.

Consider the two sequences described above :-

ABBCD
A   CD
AEFGH J
AEFGH J

Module A has zero flexibility because of its membership of the AEFGHJ sequence, but modules B, C and D each have a flexibility of one, because they can be moved one column to the right.

8.9.2.3 As a result of these considerations, two alterations were made to the program.

a) When sequences were being added to the Course Grid each member of the sequence was checked to see if it was of the same flexibility as the first member. If it was not, the sequence was abandoned, temporarily, and the next sequence starting with the same module was considered.

b) It was decided that where modules had to be split they should be split starting from the beginning of the sequence. This was despite the educational advantages of splitting modules from the end of the sequence.

Alteration (a) was made first and the program run. The same job codes as had been used before were used again so that the same module list was generated. The resulting Course Grid is shown below in Figure 8.9.2.3.

Figure 8.9.2.3

The effect of considering the flexibilities of modules within sequences

	1	2	3	4	5	6
1	7010020A	6180011A	6180021	6180031A	6180041A	6180051
2	7010020B	6180011B	7010030	6180031B	6180041B	6170011A
3	6171011	7010030	7110041A	6170021A	6170031A	6170011B
4	7110330	7110031A	7110041B	6170021B	6170031B	-------
5	6170061	7110031B	6171021	7110051A	7110051A	-------
6	-------	6171011	6171031	7110051B	7110051B	-------
7	-------	7110330	-------	6171021	6170011A	-------
8	-------	6170061	-------	6171031	6170011B	-------

All the modules in the list have been fitted.

This is an improvement on Figure 8.9.2.2, in that all the modules have been fitted, whereas earlier three were not. This has been achieved by fitting the most difficult sequence first and then using the flexibility of some of the remaining modules to find space for them in column 6. It can be seen, however, that modules are still being split unnecessarily. For example, 6180041 and 6170031, in column 5, clearly need not have been split as there is space in column 6. Column 5 contains only four different modules instead of the ideal eight, all modules in this column having been split.

8.9.2.4 The program was then altered so that modules would be split from the start of a sequence rather then from the end.
The resulting Course Grid is as shown below in Figure 8.9.2.4.

Figure 8.9.2.4
The effect of splitting modules
from the beginning of a sequence

1	7010020A	7010030A	6170021	6170021	6170031	6170031
2	7010020B	7010030B	6170011	6170011	6170011	6170011
3	6171011	7110031A	6180021	6180031A	6180041A	6180051
4	7110330	7110031B	7110041B	6180031B	6180041B	6171031
5	-------	6180011A	7110041B	7110051A	7110051A	-------
6	-------	6180011B	6171021	7110051B	7110051B	-------
7	-------	6171011	6170061	6171021	6171031	-------
8	-------	7110330	-------	6170061	-------	-------

All the modules in the list have been fitted.

As would be expected, all the modules have again been fitted.
In the last figure ten modules were split whereas in this figure only eight have been split. The split modules have been distributed differently, with more appearing earlier in the program; this is to be expected when split modules come at the beginnings of sequences.
A space has been created in column 5; this is potentially useful, as most modules are two units long and require space in two consecutive columns if they are to be fitted.

8.9.2.5 The two alterations described above have made the procedure for fitting modules into the Course Grid more effective, the most significant improvement being that the improved version can fit more modules into the Grid. There is still considerable scope for improvement, however. These improvements will be described in Chapter 9.

8.9.3 Testing by students

The students were asked to read 'Stage 2' of the student information booklet. They were given an abbreviated list of the SUPERCLASS names and codes. The shortest list would have contained only the codes and names for which relevant modules were available (the MODULES file contains only a sample of the total number of modules available). However, to show the students how the codes are built up - in effect, to explain the SUPERCLASS system - all the higher level codes leading to the initial set were also given. In addition, whether required or not, all first and second level codes and names were given. Figure 8.9.3.1 shows part of the list (the complete list, which shows which areas were included in the sample data, is given in Appendix 9).

Figure 8.9.3.1

Part of the SUPERCLASS list given to students

A		Business & Management
- A1		Business & Management (general)
- A2		Small Businesses
- A3		Management Skills, Systems & Techniques
A31		Management Skills
A312		Business Communication Skills
A3122	1	Report Writing
A318	1	Meetings and Committees

Codes marked '-' are second level headings.
Codes A1 and A2 are given because they are second level headings - there are no modules in these categories.
Codes A3, A31 and A312 are given because they lead to the code A3122, for which one module is available.
The list therefore contains more codes and names that it need, but this is a closer approximation to the full size list that would have to be given in a full implementation.
Students were asked to complete a 'SUPERCLASS Form', to show the job codes in which they were interested. The relevant part of the form is shown below.

Figure 8.9.3.2
Part of the SUPERCLASS Form

	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
1						
2						
.....	....	....	....	....	....	....
.....	....	....	....	....	....	....
22						

8.9.4 The students were told to write down all the code numbers leading to the one they had selected (hence the space for the six levels) - this was unnecessary, and caused some confusion. They should have been told to write down only the job code in which they were interested. (They were told in the written instructions and again verbally to write the most specific code they could.)

8.9.5 It was stated earlier that one of the questions to be answered was, "Can students use the SUPERCLASS system to find the required job codes?". Some of the names may not be familiar to students. If the student does not recognise the names, how can he tell if that is an area of interest. Given that only a restricted sample of the catalogue was available during the testing phase, the students did not appear to have any problems identifying the codes they wanted. There was one interesting example, however. One student asked, "What is cryophysics?". (The behaviour of materials at low temperatures.) He asked this question because he was looking for Science and Mathematics modules (category L), and was looking under L2, Physics. The point is, that although he did not know what 'Cryophysics' was, the position of the name within the catalogue indicated that it might be of interest to him. If the positioning of names in the catalogue is an aid to student use of the system, then this suggests that, whatever method is used to show the SUPERCLASS system to students, it must show enough of the names in the area being investigated that the student can recognise at least some (preferably most) of the names in that area. This would then encourage the student to find out more about the 'unknown but possibly relevant' name (17). Another hypothesis (untested) is that students will recognise names at the level at which they are working. For example, the student may not recognise cryophysics initially, but having taken a few Physics modules he will have either encountered the word or, having seen it in the catalogue (possibly when planning his next block of modules) will have asked a Physics lecturer. In other words, within the area of interest, more names will become familiar to him as he gains experience. It must be stressed that this is a possibility only - it has not been investigated.

8.9.6 The students expressed considerable interest in the SUPERCLASS system, particularly the range of subjects covered - this almost vindicated the decision to list so many superfluous codes, as described above in Figure 8.9.3.1. They were equally interested to know that modules corresponding to many of these codes were available (but not in the sample data file).

8.9.7 Following from the experiences with the "Construct Pre-entry List" program, the Selector program was modified before use by students. Code was added to record the responses of the students as they worked through the construction of their course. The program 'pauses' at various points while information is read from the data files. It was decided that these would be suitable points at which to print the responses already recorded. This allowed a smaller amount of memory to be allocated to recording the responses, without interfering with the normal 'flow' of operation. The printing operation always preceded an instruction to clear the screen, so the messages being printed did not interfere with the screen display.
The original screen display was entirely white text on blue, with no highlighting of prompts. In addition, the prompts were printed at the end of the list of modules selected. As this list grew the instructions moved further down the screen, until the screen scrolled and 'lost' the top few modules. The options to add or delete modules or fit modules to the Course Grid were repositioned to a fixed block at the right hand side of the screen and shown in inverse video, which separated from them the list of modules being

displayed. One option was reworded - "Fit to grid" was changed to "Check Course Plan" (this having been explained in the written student instructions). Originally the user was required to type a number to select an option, e.g. 0 = "Fit to grid", 1 = "Add more modules", 2 = "Delete some modules". These were changed to letters; C = Check Course Plan, A = Add more modules, D = Delete some modules.

A typical display, in this case after giving the job code A822, 'Business Correspondence', is shown below.

Figure 8.9.7

A Typical Display after entering a Job Code

```
Module List
Number of Modules 10    Course length 19   (TARGET =48)
A822  -------  -------  -------  -------  -------  -------  -------  -------  -------
1   6170011  Clerical Procedures
2   6170061  Handling Business Documents
3   6171031  Processing Business Transactions
4   6180021  Text Processing 1
5   6180031  Text Processing 1
6   6180041  Text Processing 1
7   6180051  Text Processing 1
8   7110031  Communication 2                              C  Check Course Plan
9   6180011  Producing Text                               A  Add more modules
10  7110021  Communication 1 : Written Communication      D  Delete some modules
                                                           What do you want to do?
                                                           (Type C, A or D)
```

The display has been reduced to fit on this page.

The row beginning A822 shows the job codes already entered. The nine sets of dashes represent the spaces to be filled by subsequent job codes, from which it can be seen that the prototype could only accept 10 such codes. This limitation was imposed by the memory of the computer, but it did allow all ten to be written on one line at the top of the screen

8.9.8 The instructions on how to delete modules (should that option have been chosen) contained the lines,

```
To remove a module type the number in the left hand column
To accept the list as it is ........................type 0
Press RETURN after you type a number.
```

The difficulty in repositioning this block is that the student has to be able to see the whole of the module title, and this can occupy most of the width of the screen. Reducing the width of the prompt to the necessary size would have rendered it unintelligible. This dialogue box had to be left at the end of the module list, although, as explained above, this is not the ideal place for it [18]. The box was printed in inverse video.

8.9.9 Examination of the responses showed that the errors were :-

a) giving a job code for which no modules were available [19].

b) using the RETURN key to mean 'continue' [20]

c) putting a space between the letter and the number in a job code, e.g. A 822 instead of A822. (1 student)

d) Starting to enter the next job code without first selecting 'Add more modules'.

However, as all of these errors were dealt with by the error handling routines the students were able to move through the system successfully. It was noted that many were selective about printing the Course Grid, having chosen to see it (using the Check Course Plan option). It had been anticipated that they might print it just see what it was like, but they tended to wait until a reasonable number of modules had been selected before printing results.

It was noted that those who chose to delete modules which were entry requirements to others responded to the warning that those others would have to be deleted as well. For example, a student went to delete module Archery 1, was informed that this would also delete Archery 2, cancelled his first choice and instead deleted Archery 2. He wanted to have Archery in the course, but not two modules of it. He had not realised the significance of the suffix, but, following the instructions on screen, correctly deleted the one that would still leave the other in the course.

More comments on the desirable features of a full implementation will be given in Chapter 9.

8.9.10 Errors and times taken to use the Selector

Table 8.9.10

Errors and Response Times for the Selector

Pupil	Job codes	No modules	No. of modules	Course length	Passes	Time (s)	Time (mins)
1	5	0	10	13	2	203	3.38
2	6	0	21	39	5	420	7.00
3	4	0	4	8	1	167	2.78
4	13	3	21	28	2	585	9.75
5	9	0	18	27	2	386	6.43
6	9	6	8	10	2	291	4.85
7	10	0	24	32	2	524	8.73
8	15	5	18	22	2	534	8.90
9	15	6	19	35	2	1006	16.77
10	7	1	15	19	2	403	6.72
11	13	3	20	31	2	592	9.87
12	8	0	19	36	3	495	8.25
13	5	0	22	40	5	320	5.33
Av.	9.2	1.8	16.9	26.2	2.5	456	7.6

Job codes = the number of codes supplied by the student.
No modules = the number of job codes that did not add modules to the programme.
No. of modules = the number of modules in the final programme.
Course Length = the length of the programme in half-module units (maximum = 48)
Passes = the number of passes needed to calculate the start/finish times.
Time = the overall time taken to construct the programme.

The maximum number of job codes that can be used is 10. If the student gives a code for which there are no modules, the code is not added to the list - hence the entries for student 4, 8, 9 and 11. This confusion, resulting in the student using codes for which there are no relevant modules, was referred to earlier.

Some of the students, not wanting modules concerned with Business and Administraton, used a number of job codes to obtain modules in the Sports and Recreation area. As this part of the data is not complete, they tend to have fewer modules in the final programme.

8.10 The sample data

The data used in testing the prototype consisted of information on 184 modules. The list of module reference numbers and titles is given in Appendix 8. All of the modules from category A8, Office and Secretarial

Skills, of the SCOTVEC catalogue were included, along with those quoted as preferred entry levels. This set of modules was chosen as the main sample data because it constituted a self-contained 'block'. For the program to operate satisfactorily, any module quoted as an entry requirement must appear in the Modules file. In a fully implemented version part of the Modules file manager program must deal with the possibility that a module is quoted before its record has been entered. By using a set of modules covering a clearly defined area of work, and then supplying data on all relevant modules, it was possible to avoid the problem of referring to a module for which no record had been made; it was therefore unnecessary to supply the necessary procedures to deal with that situation.
In addition to the 'A8' modules, a sample was taken from Arts and Crafts, Language and Communication, Music, Sports and Games, Science and Mathematics, Health, Woodwork, Computers, and Engineering.
The module information was taken from Module Summaries for the relevant modules (an example is given in Appendix 2.2). Job codes were added by comparing the information in the Module Summary with the SUPERCLASS categories. Where a module was judged to be relevant to a job, the SUPERCLASS code was added to the module record.

Although the sample data constitutes only six percent of the full catalogue, all the typical situations are included. The only problem that might arise in a full implementation that does not arise in the prototype is that of a single job code resulting in a very large number of related modules. The prototype can deal with only forty-eight modules, as explained in 8.4.3. A full implementation must be able to cope with a larger number and must also contain a routine to deal with the possibility of exceeding that number.

8.11 A sample output

If the reader has the relevant hardware available, the last entry in the table can be reconstructed by supplying the following job codes:-
A822, A854, A862, G32, G782

This example demonstrates all the features required of the system :-

A number of modules have entry requirements.

A number of modules have to be split

Two sequences are identified as too long, and modules have to be deleted from the end.

Some modules are common to different job codes, and are not repeated.

The programme contains modules not related to the main job code.

Table 8.11.1

Sample output

Categories from modules have been taken:-

A822	Business Correspondence
A854	Keyboarding
A862	Receptionists (business)
G32	Swimming
G782	Badminton

Module List

Number of modules = 22, Course length = 40

1	6170011	Clerical Procedures
2	6170061	Handling Business Documents
3	6171031	Processing Business Transactions
4	6180021	Text Processing 1
5	6180031	Text Processing 2
6	6180041	Text Processing 3
7	6180051	Text Processing 4
8	7110031	Communication 2
9	6180011	Producing Text
10	7110021	Communication 1 : Written Communication
11	6180090	Introduction to Word Processing
12	6180100	Word Processing Skills
13	6180111	Audio Typewriting
14	6180131	Audio Typewriting - Legal
15	6181011	Using a Keyboard
16	6181022	Producing Text from Audio
17	6170021	Communicating with Callers and Colleagues
18	6170031	Reception
19	6171021	Working in a Reception Area
20	80100	Introduction to Aquatics
21	80144	Badminton 1
22	80244	Badminton 2

(Continued on next page)

Course Plan

1	7110021	7110031A	6170011	6170011	6170011	6170011
2	80144	7110031B	6170021	6170021	6170031	6170031
3	6170061	6180011A	6180021	6180031A	6180041A	6180051
4	6171031	6180011B	6171021	6180031B	6180041B	6180111A
5	6180090	80244	6180100	6171021	--------	6180111B
6	--------	6170061	6181011	6180100	--------	6180131A
7	--------	6171031	6181022	6181022	--------	6180131B
8	--------	6180090	80100	80100	--------	--------

The total time taken to produce this output was 320 seconds.

The modules with the suffix A or B have been split, and are part of long sequences.

The remaining spaces do not allow many modules to be fitted. One full length module could be placed in columns 5 and 6 and the rest are all half module spaces, unless independent modules are split. An improved Sequencer could move module 80244 (Badmniton 2) from column 2 into column 5. This would allow a full length module to be fitted in columns 1 and 2. Moving 80100 to coulmns 4 and 5 would give a better distribution of free time, or allow a sequence to be fitted from columns 1 to 5. This re-arranged Course Plan is shown below.

Table 8.11.2

Improved Course Plan

1	7110021	7110031A	6170011	6170011	6170011	6170011
2	80144	7110031B	6170021	6170021	6170031	6170031
3	6170061	6180011A	6180021	6180031A	6180041A	6180051
4	6171031	6180011B	6171021	6180031B	6180041B	6180111A
5	6180090	--------	6180100	6171021	80244	6180111B
6	--------	6170061	6181011	6180100	--------	6180131A
7	--------	6171031	6181022	6181022	--------	6180131B
8	--------	6180090	--------	--------	80100	80100

8.12 Conclusion

The objectives of the prototyped programs were stated in 7.3 as :-

a) to show that the construction of viable modular courses is greatly facilitated by the use of the computer;

b) to develop certain techniques that will be of value in constructing a fully-operational system;

c) to show that much of the work of designing a programme can be done by the student (using the computer) without the assistance of a counsellor.

d) to gain information concerning the time required by the system to produce results.

e) to gain information about the difficulties students might have in operating the system.

In addition, the following questions were asked (in 8.6) :-

f) Can students use the SUPERCLASS system to find the required job codes?

g) How much training must students be given in the SUPERCLASS system?

a) The sample output, described in 8.11, shows what can be achieved by the simple prototype. A more complete implementation should be able to fit one or two more modules into the programme. The sample was produced in about 5.5 minutes, although this time would increase if the student had chosen to delete modules. One can only guess at the time that would be required to do this manually.

b) The approach used in the prototype has been described in Chapter 7. These techniques work, although the coding is convoluted to the point of being unintelligible.

c) In the test cases all of the work was done by the student working individually. The counsellor need only be involved in negotiating such questions as, "Can the student be given credit for work experience?", or in deciding if the programme is too restrictive. (The sample output is heavily concentrated in one area.)

d) The average time taken to construct the Pre-entry List was 7 minutes, with a maximum of 11. The mean time to construct a course containing 17 modules and fill 26 cells of the Course Grid was about 8 minutes. The longest time recorded was 17 minutes to assemble a programme containing 19 modules in 35 cells (which is a more realistic course).
It should be possible to construct the Pre-entry List in 10 minutes, with about 20 minutes taken to complete a full course. The whole process, for a novice user, should be capable of completion within 30 to 40 minutes.

e) The types of error made by students have been described. Some of these can be blamed on the user interface itself, while others may be attributed to the student failing to read the on-screen intsructions when he becomes familiar with the procedure. This is only to be expected, and a full version should allow students to recover from these errors.

f) and g) The students were quite confident in their use of the SUPERCLASS system with very little training. The use of codes that did not yield modules is not a major error - unless the SUPERCLASS file is restricted to relevant codes only this is likely to occur in practice. On balance, providing these codes is thought to be useful in that it shows the construction of the SUPERCLASS hierarchy more clearly.

In addition, the importance of having accurate data-bases became very evident during the test process. A number of errors were located - one can only assume that there are more. Ensuring that all data is correct and up to date, in particular that data in one file matches the data in other files, will be an essential procedure for a full version.

8.13 Notes

1) The suffix '%' indicates that the variable is an integer.

2) The program crashing can be considered an extreme case of 'producing an incorrect result'.

3) *"Benevolent testing is performed by the author of the software whereas adversary testing is performed by an independent tester .... who has no personal attachment to the program."* (Marcoty 1991, p 290)

4) Marcoty (1991, p 298) describes two approaches to the testing of individual parts of a program :-

"1. Test each integrand in complete isolation."

This requires a driver and possibly 'stubs', simulations of the procedures invoked by the sub-program being tested.

"2. Test each integrand by combining with other previously tested integrands."

This he divides into two types; top-down and bottom-up.

The test procedure used in the prototype was type 2, mainly bottom-up, which allowed the interfaces with previously written sub-programs to be tested at the same time. (Although that can be a disadvantage on occasions.)

5) Bottom-up testing
"The usual technique, known as module or unit testing, is to construct a 'test harness' or 'test bed' for each component. This is another program whose sole function is to invoke the component under test in a way that is consistent with its eventual role in the complete system." (Bell 1992, p282)
6) See Chapter 7.2
7) This is an exception to the statement in 8.4.1.
8) See Chapter 6.2, the Requirements Specification, part (f).
9) Also in Chapter 6.2, two of the requirements relate to time; one is to minimise the time taken to train students, the other is to minmise the time taken to produce courses. While minimising the time can be achieved by ensuring that 'high speed' procedures are always used in the program, it is felt to be useful to know a 'specimen' time that is to be improved on. Two of the prototype objectives are relevant to these points.
10) In a full implementation the computer could, if linked to the administration files, check that the student's name and number are recorded, and that the number matches the name.
11) In a full implementation the computer could check the module reference against the MODULES file and supply the name, for confirmation by the student.
12) While the word processor used to produce this text can show inverse video on screen, the printer cannot print it correctly.
13) Eight of these colours alternate the text and background colours, i.e. appear to flash. These can be used as 'attention-getters', but are not very useful normally.
14) Every school in Strathclyde Region, both Primary and Secondary, uses the BBC computer to some extent. It was the first computer to be used for teaching Standard grade Computer Science, and most secondary schools were equipped with a computer room containing twenty of them.

Most pupils will have had experience of using the BBC, sometimes on an ad hoc basis in subject teaching, and frequently in a 'Keyboarding' course in S1 or S2. Keyboarding involves both manual typewriters and word processors.

15) In the test, hypothetical reference numbers were given, to find out how many mistakes were made in entering them into the computer. A record was kept of the numbers supplied so that any errors in entering them could be identified.
16) The author's preference for white text on blue is limited to the BBC computer. This text is typed using the standard black text on white, without any feelings of strain. This preference is more a function of screen resolution than colour.
17) In a full implementation the student would be able to see the Module Summary for any modules linked to this job name/code.
18) An improved version might have the dialogue box displayed at the request of the student and removed, by the student, if he needed to see a title that was obscured by it, e.g. a Help window as used on many comoputers with pull down menus.
19) On testing this later, it was found that the error handling routine is flawed. Fortunately, the flaw only causes an unnecessary but harmless procedure to be invoked, and then correctly indicates that an input error has been made.
20) This was caused by the presence of the 'superfluous' job codes in the SUPERCLASS list provided to them, the requirement to write down all the codes leading to the code of interest, and a failure to stress that only the codes with a number (of modules) beside them were worth entering.

Chapter 9
Recommendations for a full Implementation

9.1 Introduction

As a result of the testing process, improvements to the prototypes can be suggested, and this chapter will describe these improvements. Some of these are relatively minor and follow directly from the prototype. Others are more radical, and would be desirable in a full-scale implementation.

9.2 Constructing the Pre-entry List

9.2.1 The SCE subject names should be shown in lower case, as this is easier to read.

9.2.2 The student should identify SCE subjects in the sequence, Ordinary grade, Standard grade, Higher grade. Within each of these blocks only the relevant subject names should be shown. This will marginally reduce the time taken to find a given subject (because slightly fewer will be shown) and will eliminate (or at least reduce) the confusion arising from the use of similar, but different, names at different grades.

9.2.3 An editor should be provided so that the student can correct any errors before leaving this part of the system.

9.2.4 When the student is asked to identify the modules already completed, only the reference number of a module is required; this is to reduce the amount of typing required of the student. As it is likely that typing errors will still occur, the program should check the MODULES file and display the module name. The student should then be asked to confirm the name, rather than the reference number. Failure to find the reference number in the file would indicate an error in typing the reference number.

9.3 The Selector program

9.3.1 In the prototype the Module List is displayed as shown below.

Position	Reference	Title
1	7110011	Communication 1 : Spoken Communication
2	7110021	Communication 1 : Written Communication
3	3161261	Experimental Procedures - Biology
4	3161131	Science Practical Skills

The module reference numbers are an irrelevant distraction and should not be shown. This would also eliminate the possibility that a student could misunderstand the instruction for deleting modules, and type the reference number instead of the position number. While no students in the sample group actually made this mistake, it would be sensible to remove the possibility from the outset.
The 'highlight and select' method (as used to indicate SCE subjects) would be preferable to typing the number of the module to be deleted.

9.3.2 Even in the most flexible of courses it is likely that some modules will be compulsory. These must be 'flagged' so that they cannot be deleted. In the more restricted qualification-centred courses this is essential.

9.3.3 As the Module List grows, listing all the modules identified can be problematic, as eventually they cannot all be fitted on one screen. The display format must therefore be designed to accommodate this, either by allowing the student to scroll through the list or by showing them as separate pages. One solution would be to extend the idea of flagging modules (9.3.2, above). The student can be asked to identify those modules he feels are essential, i.e. he would not be happy with the programme if it did not contain them. These can then be kept on a separate screen, along with any compulsory modules, the 'normal' screen showing only new modules, those identified with the latest job code entered, and those which the student might want to delete.

9.3.4 The student should be told, preferably on screen, that "Check the Course Plan" is time consuming and unnecessary when there are only a few modules in the list. On the other hand, as the number of modules increases he should be prompted to use this option, and then prompted for deletions.

9.3.5 In the prototype it is possible to delete all the modules associated with a job code, but the job code is still shown as having been selected. As the number of job codes is limited to ten, this uses up a code unnecessarily. In a full implementation this restriction might be lifted completely, an unlimited number of job codes being allowed, or at least extended, perhaps with twenty codes being allowed. It is still desirable that the rejected job code is displayed, so that the student does not inadvertently waste time checking it again. A further improvement on this would be to allow the student to reject the job code itself, the computer deleting the associated modules for him. This would allow the user to 'experiment' with the system more easily.

> *"Learners apparently prefer trying things out on the computer rather than reading lengthy manuals. They want to perform meaningful, familiar tasks immediately and see the results for themselves."*
>
> (Shneiderman 1987 p370)

If this is the case, the procedures for erasing or reversing actions must be easy to operate.

9.3.6 In the sequence of modules, A->B, 'A' has been called an entry requirement or an 'entry module'. 'B' may be referred to as an 'exit module'. An exit module is any module to which the given module is an entry requirement. It is useful to know these exit modules as they are modules that follow from the given module; i.e. they are in the same field of study but at a more advanced level. These are the modules that allow a module to lead to greater depth of study. The mechanism by which the exit modules can be identified and entered into the module record will be explained later.

9.3.7 In some cases the student may have used all the job codes that he identified as relevant but there is still space in the Course Grid. A number of different approaches may be used to find additional modules :-

9.3.7.1 The student goes back to the SUPERCLASS system and identifies more job codes. This is the most flexible approach as he has the freedom to chose from any category. However, he may want advice or simply not know where to start - he has, after all, already named the areas that interest him.

9.3.7.2 Exit modules are mentioned immediately above (9.3.6). In the prototype these were not used, although the module records have a field for the purpose. This is because it was found that, given such a small sample, it was possible to link every module to the appropriate job code. For example, consider the sequence A->B. 'A' is related to job code 'T', so when the student supplies the code 'T' module 'A' is found. But 'B' is also related to 'T' and will be found as well. However, as the system expands it is possible that 'B' (being more specialised) may be linked to 'T1234' by the person completing the module record. Code 'T' will not then locate 'B'. If exit modules are used 'B' will be found through its association with 'A'. It can be argued that this is an operator [1] fault; the module record for 'B' should contain 'T', 'T1', 'T12', 'T123' as well as 'T1234'. Exit modules are a way of reducing the consequences of incomplete data in the module record. An alternative to entering all levels of job code will be discussed below.
One drawback to the use of exit modules is that some modules, for example basic Arithmetic or basic English, may be entry requirements to a large number of others - to exaggerate slightly, the computer would be saying, "Now that you can count you can take every module that involves counting".

9.3.7.3 If the student has identified a reasonable number of modules, perhaps two thirds of the Course Grid, the computer may be able to suggest additional modules by comparing the modules already selected with those required for specific 'qualifications'. The computer would then list the modules that appear in the Qualification list, but not in the student's Module List. In this way the student would not only take the modules directly related to the area of interest, but has the option of gaining a recognised qualification.
This is the sort of advice that an experienced careers officer would give, and it would be highly desirable that a computer program designed to select the best group of modules for a student should be able to do the same thing. In this respect it would begin to behave like an 'expert system', a computer

program which can simulate the behaviour of a human expert. The computer has the advantage over the human of having an infallible memory (based on the data stored in it). A human may immediately recognise a group of modules as being similar to that required for a different purpose, but only if he is familiar with that group. The computer does not 'recognise' a group in the same way, but can search for similarities without boredom or fatigue limiting the effectiveness of its search.
The limitation on this approach is that, with qualification-centred courses being so prescriptive, the student may have to delete many of the modules he chose in order to take the recognised course. That, however, is his decision - he is not being forced to take the qualification, the possibility is simply being pointed out to him.
The same principle can be applied to a search of the Advanced courses list, i.e. check the entry requirements to higher education courses in related areas to see if, by adding a few appropriate modules, the student could satisfy the requirements.
This process might apply when a student has chosen a qualification-centred course. By checking the Qualifications lists in related areas it may be possible to make the final programme satisfy more than one qualification, so that if the student fails one module he may still have the necessary modules for the second qualification. This, however, depends on both qualifications being flexible enough. It is possibly more likely that a search of the Advanced lists would identify a course to which the student could gain entry by adding appropriate modules to his programme. Finally, a search of the Qualifications file might be useful to the student who has chosen a higher education-centred course. The intention here would be to provide the student with an option in case he is unsuccessful in his application for admission to the advanced course.

9.3.8 The SUPERCLASS system is a vital part of the procedures described here. A job code can be used to locate relevant modules, as shown in the prototype. It can also be used as a pointer to a Qualification, or to an Advanced course. The ease with which the system can be used depends on the ease with which the student can search the SUPERCLASS categories to locate the areas of interest to him. It was suggested in Chapter 8 that the recognition of a relevant area may depend on the student seeing and recognising job names in surrounding areas. The way the SUPERCLASS system is displayed for the student is therefore crucial.

9.3.8.1 SUPERCLASS as a series of menus

Initially the student should be presented with the seventeen main classes. When he selects from this list he is shown a list of the next more specific level within the area selected.
For example if the student selects class A, Business & management, he would then be shown the list given in Figure 9.3.8.1.

Figure 9.3.8.1

SUPERCLASS Display at Second Level

A Business & management
A.1 business & management (general)
A.2 small businesses
A.3 management skills systems & techniques
A.4 human resources management
A.5 financial management & accounting
A.6 marketing sales & distribution
A.7 information & library management
A.8 office & secretarial skills
A.9 public administration

Ideally, this process would be repeated, with the student selecting from a list at one level, the computer then showing the next more specific level for the class. For example, he would be presented with the following lists, in order;

a) the list of classes at the first level, e.g. A
b) second level, e.g. A1
c) third level, e.g. A11
d) fourth level, e.g. A111
e) fifth level, e.g. A1111
f) sixth level, e.g. A11111

One difficulty with this approach is that the number of sub-divisions at any given level varies from one class and level to another. For example, in class A there are nine records at the second level (A1-9), but if the student chose A3, this has only two sub-divisions at the third level (A31 and A32). If all of the third and fourth level classes are shown, the list would contain nineteen entries (A31, A311 - A318 and A32, A321 - A329). Neither of these is ideal. The quote given earlier, in Chapter 6.8.3(b) is again relevant :-

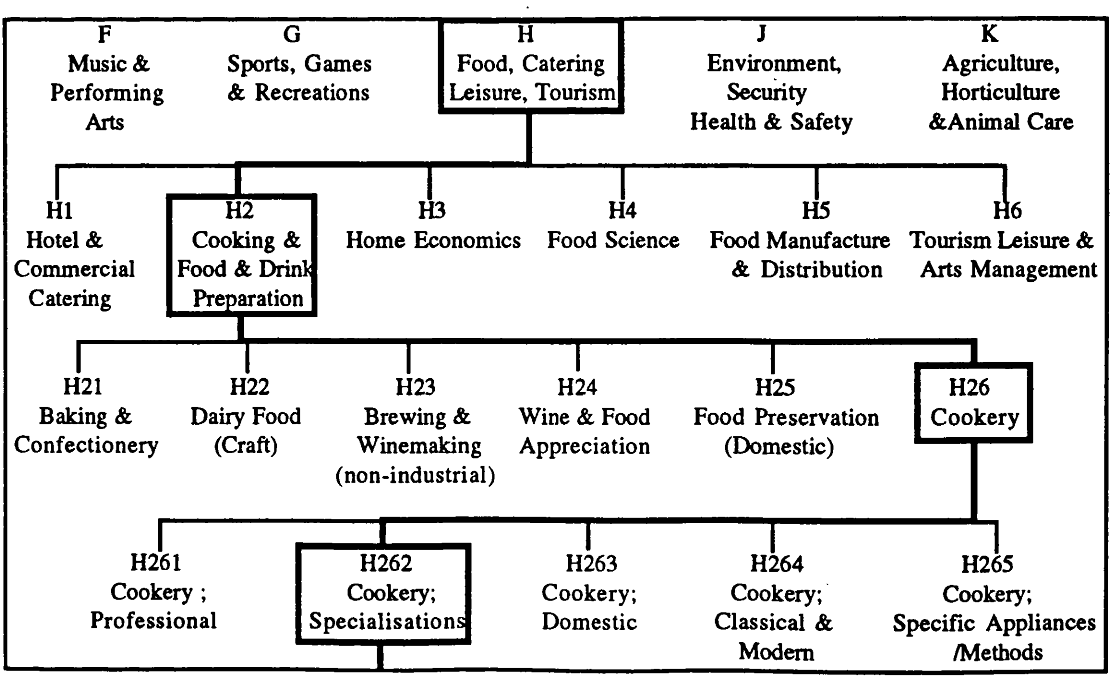
> *"Several authors have urged four to eight items per menu, but at the same time they urge no more than three to four levels. With large menu applications one or both of these guidelines must be compromised."*
>
> (Shneiderman 1987 p98)

9.3.8.2 A SUPERCLASS map

An alternative approach is to use the VDU as a 'window' on a map of the SUPERCLASS system, and makes use of the more modern 'pointing' methods for indicating selections. The amount of information that can be shown on screen at one time depends on the hardware in use. The examples given assume a screen as used on the Macintosh Classic. The initial appearance of the screen is as shown below.

Figure 9.3.8.2.1

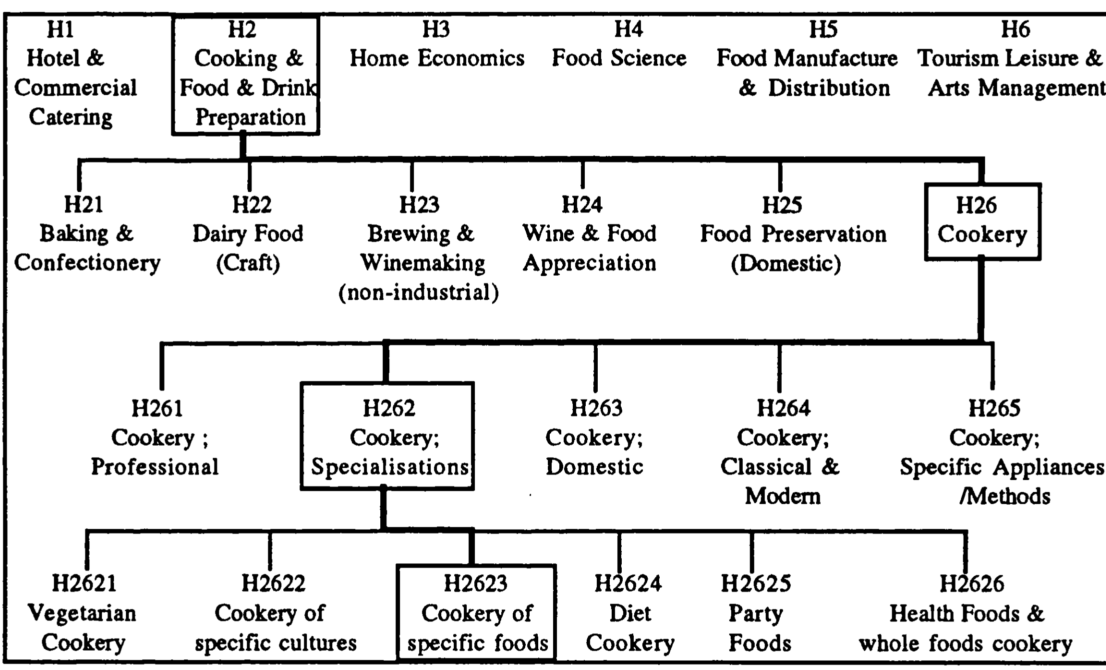
Map of the SUPERCLASS System (A)



The outer box represents the usable part of the screen - the space above this would be used for the 'menus', giving the options available to the student. (The screen has been reduced in size slightly to meet the binding requirements of this work.) In the case shown above the student has pointed and clicked on category H, which has been centred on the screen. The student has then pointed and clicked on H2, and all the subsets of H2 are shown. This is followed by pointing and clicking on H26 and H262. At some point, probably immediately after selecting category H, the screen would scroll to reveal the lowest level, as shown below.

Figure 9.3.8.2.2

Map of the SUPERCLASS System (B)



By pointing and 'double-clicking' on a category (at any level) the student would indicate he wanted to see the modules available for that category - the equivalent of typing the job code in the prototype.

Although this example was chosen at random, it happens to show all the categories available; selecting another category, it might not be possible to show all the subsets on screen simultaneously. For example, category H12 descends to the level H12246, which would require another scroll to show the bottom level. However, it would appear that enough of the surrounding area can be shown to let the student see what is available. The number of categories that can be shown simultaneously will depend on the usable size of the VDU and the font and font size chosen for the display.

The main advantages of this method are that

a) the student can see surrounding areas, so that he can interpret unfamiliar names because of the context in which they appear, and
b) the student is not required to type anything - he merely points and clicks. This eliminates the problem of typing errors.
c) He can change his mind easily. For example, if the subsets of H262 are not what he was looking for he can simply point and click on, say H263. Again, this encourages 'browsing' or experimentation.

9.3.9 Examination of the SUPERCLASS system suggests yet another way of extending a programme, but without imposing the limitations of the qualification- or higher education-centred courses. It is possible to move around the SUPERCLASS system by manipulating the digits in the job code. For example, suppose the student has given the job code H2623 and this has not yielded enough modules. By changing the last digit other codes at the same level are generated, e.g. H2624 or H2622. (In this particular example, inspection of Figure 9.3.8.2.2 shows that an amateur cook may well be interested in modules from any of the fifth level codes.)

By removing the last digit the student is taken to a higher level, H262. Changing the last digit there, and then adding a digit, the student is taken to codes such as H2611 or H2631. These codes are all in the same general area from which the original code was taken and might contain modules of interest to the student.

If the map system described above is being used this is probably obvious to the student and would not require additional coding - he simply points and clicks without asking for suggestions from the computer. In the more conventional system used in the prototype an extra option would be added to the main display, i.e.

C - Check Course Plan
A - Add more modules
D - Delete some modules
S - Suggest more modules

9.4 The Sequencer

9.4.1 The splitting of modules was introduced in order to allow long sequences to be fitted into the Course Grid. The splitting of independent modules, those that are not in sequences, was not even considered. However, there are situations where it would be useful. Consider the simplified Course Grid :-

1	2	3	4	5	6	
X	X	X	X	X		Assume that all other rows are full.
X	X	X	X	X		X = a module

An independent module, 'B', still has to be fitted, but it is of the normal length, i.e. it occupies two columns. It cannot be fitted unless it is split. This situation is likely to arise, because independent modules are the most flexible and are therefore fitted into the Course Grid last. It seems almost paradoxical that because we delay fitting the most flexible modules until the end, they may be impossible to fit. This is because flexibility is defined in relation to any empty Course Grid.

The reader may remember that when splitting modules was first suggested, in Chapter 7, it was described as 'somewhat contentious'. There are educational reasons for allowing it (it allows longer sequences to be used in the student's programme), but there equally valid arguments against it (it concentrates too much time on a smaller number of modules, emphasising depth at the expense of breadth). A decision must be made as to whether splitting modules should be allowed at all, and this decision must take into account, at least, the abilities of the student. If it is decided that splitting should be allowed, a full implementation should include it as an option, rather than as an automatic reaction to a given situation, as in the prototype. It may be desirable to limit the number of modules that can be split in any one programme, and that number must be at the discretion of the student's counsellor (it should probably be a subject for negotiation).

9.4.2 Flexibility was defined from

flexibility = latest start time - earliest start time (see Chapter 6.10.2)

Consider the group of modules in this simplified Course Grid (Figure 9.4.2a).

Figure 9.4.2

Differences in Flexibility

(a)

1	2	3	4	5	6
A	A	B	B		
C	C	C	C		

(b)

1	2	3	4	5	6
		A	A	B	B
		C	C	C	C

(c)

1	2	3	4	5	6
A	A			B	B
C	C	C	C		

All of the modules have a flexibility of 2 (=3-1). The sequence AB can be moved as one complete block, and therefore has the same number of options as C. But the sequence AB can also be separated, as shown in (c). Module 'C' cannot be separated in this way, because of the rule that all parts of a module must be in adjacent columns (see Chapter 7.6.9.2). Thus, although the modules have the same flexibility, 'A' and 'B' are easier to fit than C (in the sense that, as there are more ways of fitting them there is a greater probability of their being fitted). As a general rule, the modules in a sequence are easier to fit than a single module of the same flexibility. (Hence the comment in Chapter 7.6.8, that 'this definition of flexibility is not entirely satisfactory'). It might be possible to define flexibility in a more useful way, but it was felt that another approach might be more satisfactory. The problem with the idea of flexibility is the way it is used. It is assumed that more flexible modules will be easier to fit in the Course Grid, but, as explained above, this is not necessarily the case.

9.4.3 It can be seen, from the considerations given above and earlier, that there are problems with the process of fitting modules into the Course Grid. The present definition of flexibility is less useful than it could be, it may not be possible to fit some of the most flexible modules because of the state of the Grid when the attempt is made to fit them, and, unless modules are split at the beginning of the sequence, there is the danger of splitting modules unnecessarily. The author feels that the main fault with the method used in the prototype is that it is based on assumptions that may not be valid in every case, and given the large number of possible permutations of sequence lengths, sequence structure, module lengths, branches etc., there is a very large number of cases.

Two methods of improving the Sequencer are outlined below. Neither has been developed, as it was felt that the prototype had demonstrated the point that modules can be fitted into the Course Grid.

9.4.3.1 Consider two modules which have not been fitted into the Course Grid.

	l	es	ls	f
A	3	1	4	3
B	2	3	5	2

l = length
es = earliest start
ls = latest start
f = flexibility

1	2	3	4	5	6

The Course Grid is almost full, with only one row empty. If a module is fitted into column 1, then there is no space left in that column. 'A' cannot now start in column 1 - its earliest start is now 2, which changes its flexibility.

	l	es	ls	f
A	3	**2**	4	**2**
B	2	3	5	2

1	2	3	4	5	6
X					

If another module is placed in column 2, A's flexibility changes again.

	l	es	ls	f
A	3	**3**	4	**1**
B	2	3	5	2

1	2	3	4	5	6
X	X				

A is now less flexible than B, and should be fitted in preference to B if there are no less flexible modules to be fitted. Recalculating the start and finish times in this way will ensure that the most 'urgent' cases are dealt with first. One more example, however, shows another factor which must be taken into account in recalculating flexibility. Starting from the empty row, suppose a single half module had been fitted in column 3.

	l	es	ls	f
A	3	4	4	0
B	2	4	5	1

1	2	3	4	5	6
		X			

Neither A nor B can now be fitted in columns 1, 2 or 3, because of the rule that all parts of a module must be fitted in adjacent columns. 'A' now has a flexibility of 0. Again 'A' should be fitted in preference to 'B'. 'A', it can be argued, is of more value to the student because it is longer than 'B' - it makes better use of the time available. Note that, if modules had been fitted according to their original flexibilities, 'B' would have been fitted first.

9.4.3.2 A further improvement on the Sequencer would be to maximise the number of adjacent spaces in the Grid, rather than fitting all modules at their earliest start times, as this tends to allow a wider range of modules to be fitted. (See 8.11.2, improving the Course Grid from the sample output.)

9.4.3.3 Both of these improvements need only be made when the Grid is nearing completion, or when all split modules have been fitted (split modules need two spaces in a column). As a module is only split because it, or the sequence to which it belongs, is too long to fit normally, split modules have zero flexibility and are fitted first. The second condition is therefore likely to occur before the first.

The improved Sequencer can be summarised as follows :-

a) Use the prototyped method, with a new definition of flexibility to distinguish between single modules and sequences.

b) When all split modules have been fitted, if any column is filled move modules to create space in that column (if possible).

c) When only one row is left unfilled, recalculate flexibilities between fitting modules.

9.4.4 The problems associated with splitting modules unnecessarily were described in Chapter 8.9.2. If splitting modules is accepted as valid and desirable, they must only be split when this is required to make long sequences possible. The problem can be overcome using the procedure described here.

When the first attempt to fill the Course Grid has been completed, a second pass through the sequences should be made. Where a sequence contains split modules, but is shorter than six columns, i.e. modules may have been split unnecessarily, some of the modules can be 'un-split', i.e. the two parts can be re-combined to give the original module, to lengthen the sequence to six columns. Before any particular module is un-split a check should be made on whether it belongs to any other sequences or not; a module may have been split to allow another sequence to fit, and cannot now be re-combined without preventing that other sequence being used. Using the same example as in 8.9.2, two sequences were ABCD and AEFGHJ (all modules are of normal length i.e. occupy 2 columns). These were fitted into the Grid as shown below.

```
A B B C D
A     C D
  E F G H J
  E F G H J
```

The first sequence was too short; as module 'A' is the first in the sequence, it might be the first considered for 'un-splitting'; this would be inappropriate, as it must be split to allow the second sequence to be included. Either of the two later modules, 'C' or 'D' could be re-combined to reduce the number of split modules.

It will be noted that 'un-splitting' modules has the effect of reducing the number of rows occupied, which is in agreement with the principle of maximising adjacent spaces, provide it does not fill the next column.

9.4.5 If modules have been identified as essential (see 9.3.2 and 9.3.3) either they can be fitted earlier than their flexibility would indicate, or they can be fitted before other modules of the same flexibility - the decision could be based on how essential they are considered to be. Compulsory modules **must** be fitted, whereas although it is highly desirable that modules in which the student has expressed a particular interest are fitted, the course is still possible without them.

9.5 The Evaluator

The Sequencer passes groups of 'possible programmes' of modules to the Evaluator, i.e., each programme has been checked to ensure that it can be fitted into the Course Grid. The purpose of the Evaluator is to assign a score to each programme; the programme with the highest score is the best programme for the student, and becomes the 'actual programme'. A variation on this, allowing the student to make one more choice, will be discussed towards the end of this section.

9.5.1 Characteristics of a good course

The best course for a student will have the following characteristics;

a) it will make the best use of the student's time; this means that there should be as many modules as possible in the course. For a full-time student this means that there should be the equivalent of forty-eight half-modules in the programme. This does not imply that he should spend all of this time in class; the study period may form a valuable part of his work. For the purposes described here, the study period can be treated as a module. In the rest of this discussion the word 'module' should be taken to include the study module or modules.

   Depending on the particular combination of modules, it is sometimes not possible to achieve this ideal, of having all the cells in the Course Grid filled, but a desirable course may still result.

b) The programme must conform to the college policies. Certain modules may be compulsory, and this includes the possibility of a specified amount of study time. These compulsory modules must be in the programme.

   Modules may be compulsory because of a criterion specified by the student; for example, if he wants a specified qualification, he must study the compulsory modules.

c) The programme must contain all the modules identified by the student as 'essential', in so far as they can be fitted (if the student specified twenty-five full length modules as essential, one cannot be fitted as there are only twenty-four spaces).

d) The programme must involve some degree of progression from basic to more advanced skills.

e) The programme should have some degree of breadth; it should not be entirely concerned with one area of work or recreation.

9.5.2 The scoring system

The Evaluator should assign scores to modules and, as will be explained later, the whole course. The scoring system must be such as to value the factors described above. The Evaluator assigns scores to courses which are offered to it; it evaluates something which has already been done. However, in the same way that 'pay' is a reward for work done, and can also be seen as an encouragement to do more, the scoring system can be spoken of as if it encourages the desirable features of the program. To be strictly accurate of course it does not encourage anything; the Sequencer does not 'know' about the scoring system and cannot learn to pass only desirable courses to the Evaluator.

(If it could, the functions of the Evaluator and the Sequencer would have been combined; it is no doubt possible to build such a system, but it has not been the approach adopted here. For simplicity, and for no other reason, the two are considered as separate units.)

In the context of this chapter, the sentence "The Evaluator encourages the use of a module" can be interpreted as meaning "The Evaluator assigns a score to the module in such a way that the probability that the module will appear in the final course is increased".

Following from consideration of the factors (a) to (e), in 9.5.1, the implications for the scoring system are as follows.

a) The scoring system must take account of the number of modules in the course.
b) All compulsory modules will be in all the 'possible' programmes; the Sequencer is designed to reject any course which does not include all of the compulsory modules.
c) The scoring system must give higher scores to programmes that contain sequences of modules, as it is these sequences that give the desired progression. A large number of independent modules, those that are not related to any others, should result in a lower score.
d) The scoring system must give higher scores to programmes that contain a wide variety of modules, in the sense that they are drawn from different areas of the module catalogue. This implies that it will give low scores to programmes that involve undue specialisation, where all modules are taken from the same area.

Some of these principles require additional consideration.

a) In two extreme cases, one course might contain twenty-four full length modules, while another might contain forty-eight half modules. It can be argued that a full module is intrinsically better than two half modules; because the full module lasts longer it allows skills to be developed to a higher degree than the two separate half-modules. This assumes however, that any module is a linear sequence of skills, each related to, and a progression from, the previous skill. In fact, while any module is likely to contain related skills, it is unlikely that they can be described as one skill, developed from the beginning to the end of the module. Such modules may exist, but they are in the minority. Most modules contain clusters of skills; within a cluster the skills are developed, but between clusters the skills are unrelated (except in so far as they are all designed to satisfy the learning outcomes of the module). On this basis one full module is no better than two half-modules. Apart from this consideration, the two half-modules may be part of a sequence, in which case the skills are developed throughout both. For these reasons it is felt that one full module should be given the same score as two half-modules. On that basis, it is not the number of modules, but the number of half-module equivalents that should be counted.

b) If all compulsory modules appear in all the courses being evaluated, need they be given scores? All courses will have the same score for these modules; any differences will lie with the other modules. It appears to be a waste of time to evaluate these compulsory modules as they do not help to discriminate between courses. This argument only considers the possibility of comparing one course with another where all the courses are designed for one student. It may be desirable to compare the course finally selected with some external criterion score. This might be done by an advisor, for example, as a check that the student had used the system correctly. A copy of the output from the system might be automatically directed to the advisor. If the course score is high enough he would accept it without comment, as it satisfies the principles of course construction. If it falls below the criterion score, he might choose to interview the student with a view to improving the selection of modules.

It is debatable whether this approach would be valid; given the variation in possible values it might not be possible to decide on a criterion score that could be widely applied. Even having different criterion scores related to the different criteria used to select modules may not be effective.

The decision as to whether compulsory modules are scored or not hinges on three facts;

i) The comparisons described above might be valuable; it would be undesirable to exclude the possibility of such comparisons being made.

ii) The only objection to scoring the compulsory modules is that it is a 'waste of time'. However, because these modules appear in every course the scores have only to be calculated once; the value ascribed to them is memorised and simply included in the score of every course. It does not, therefore, waste very much time.

iii) The true value of a module is not known until the whole course is considered. A compulsory module may acquire a larger value in one course because it is used as an entry requirement to another, non-compulsory, module that is omitted from other courses.

It is therefore felt that compulsory modules should be evaluated in the same way as all the others.

c) Attaching a higher value to modules which are entry requirements to others encourages the use of sequences and awards a score related to the sequence length. However, this raises a question; if module A is the entry requirement to module B, does it matter whether B is included in the course or not? If module B is not present, and A was only included because it was an entry requiiement to B, then A will contribute nothing to the course if B is not present. According to this argument it does matter; module A should only be given the additional score if B is present.

Ideally, module A should be removed from the course if B cannot be included as well. This approach, however, assumes that the course is an end in itself; that the student will not be studying modules after this course is completed. The student may intend staying at college for two years; the Course Grid only deals with one year at a time. The simple solution in this case is to extend the Course Grid so that it includes both years. Even if the Course Grid were designed to cover two years, it would still treat the (two year) course as a fixed unit, isolated from any later studies. If the idea of 'continuing' education is to be encouraged, where young (or old) adults return to college as and when required by changing circumstances or changing skills requirements, then attention should be paid to possible future attendance. An example may help to put this in context. During the evaluation of the program it was found that the module "Shorthand Transcription 6 - Medical" had so many sequential entry requirements that it could never be included in any course lasting only one year. On that basis some of the entry requirements could be discarded. But discarding these modules would significantly inhibit any chance that the student could, at a later date, return to complete the module, because the student would then have to start by studying the entry requirements. At a later stage in the student's career, he may be more interested in part-time study; having to acquire the entry requirements would lengthen the period allocated to study, and might deter the student from ever attempting to up-grade his qualification. It is therefore felt that modules should be given additional points if they are entry requirements, even if the module to which they refer is not in the course. It is possible to do this in two stages; award points if the module is an entry requirement, and award additional points if the module to which it refers is in the course. This distinguishes between the two types and prevents too much emphasis being placed on the *possibility* of modules being studied rather than the *actuality*.

It may be remembered that the idea of promoting sequences came from the Action Plan's use of the word 'articulation'. This has been interpreted as articulation within the course, from one module to the next, i.e.

sequences. The checks, made by the Selector, on the possibility of more advanced study promote articulation in its wider sense; articulation with later studies. Valuing modules that are entry requirements to modules that are not in the course, but might be studied at a later date, also promotes articulation in its wider sense.
If a module is an entry requirement to another which is in the course, this fact can be recognised because it is part of a sequence (but not the last module in the sequence). If the module is an entry requirement to one which is not in the course, this can be recognised because the module to which it refers will be quoted in the module record as an exit module (see Chapter 9.3.6).

d) The breadth of a course cannot be measured by looking at an individual module; it is the course that has breadth, not the modules within it. Some courses do not allow for very much breadth. For example, the criterion, 'Qualifications', tends to result in specialised courses. In comparing two courses for a student all we can do is ensure that full use is made of whatever breadth is possible. Breadth can be defined as "the variety of modules present in the possible course as a proportion of the total variety possible". In this context, variety can be measured by counting the number of different areas of the module catalogue from which modules have been taken. One way of doing that is to look at the category code which forms part of the reference number for each module. The full reference number for a module might be "B3 7100200". The "B3" is a SUPERCLASS code describing the job area to which the module is related. In a highly specialised course, all modules might be taken from the same area, yielding the same code for all. A better choice is to use the job code recorded for each module. This is more specific; a typical job code might be B312. Using the whole code might equate to counting all the modules selected, because it is highly specific and every module might have a different code (while still being in category B3.) Codes should probably be read to a depth of three; i.e. at the level B31.

9.5.3 Summary of the scoring system

a)i) Every filled cell in the Course Grid is given a score.

ii) If study periods are to be encouraged they can either be treated as modules in their own right, or, for preference, the following approach can be used.

In this context, a study 'period' means one cell in the Course Grid, equivalent to a half-module, or three hours per week. The Evaluator, looking down each column of the Grid, awards a score for the first empty cell it finds, and 0 points for any others.

b) No distinction is made between compulsory modules and any others.

c)i) Cells containing modules which are entry requirements are given an additional score.

ii) Cells containing modules which are entry requirements to modules which are in the course are given additional points.

d) The number of different SUPERCLASS job-codes for the complete set of modules identified by the Selector is counted. This is called X.
The number of different job-codes found in the sub-set being evaluated is counted. This is called Y.
The ratio Y/X is calculated. This ratio is multiplied by a factor to yield a score.

The score for the course is the sum of all these individual scores.

The actual number of points awarded for each characteristic can be varied to allow different weightings to be given.
For example, if the college decides that it is very beneficial to a student to have study periods, perhaps one in every column, a higher score can be assigned, making it more likely that a course containing the specified distribution of study periods is ultimately selected as the best for the student. (This kind of decision can be incorporated into the College data-file.)
It is also possible to emphasise, for example, breadth rather than depth, by increasing the factor used in (d), above, relative to the scores awarded to sequences. This kind of variability adds another dimension to tailoring the

programme to suit the needs of the student.

9.5.4 The best programme

We have now considered the operation of the three programs, the Module Selector, the Sequencer and the Programme Evaluator. It has been stated that control should pass from one to the other in the order given, that is;

a) The Selector identifies relevant modules;
b) Using some of these modules the Sequencer constructs a programme that can be fitted into the time available;
c) The Evaluator calculates a score for the programme.

The next stage is to consider a different combination of modules, from all those selected, and construct a workable course from that. When all possible combinations have been evaluated, the course which gives the highest score is selected as being the best for the student. However, this simple approach requires some thought.

While computers are fast (and speeds are increasing as better machines are developed) they are not 'infinitely fast' - any computer operation takes a certain amount of time. When a very large number of operations has to be performed the overall time taken may become appreciable.

Using the Selector, the time taken to identify the relevant modules can be minimised by adequate indexing, so that one module 'points to' other related modules. It should never be necessary to read the whole module file to find a module of a given type.

In the Evaluator the computer has to assign scores to each module according to various characteristics which have already been identified, and then add all the scores together to give the module score. In simplistic terms this can be described as "write down forty-eight numbers and add them up" (forty-eight is the maximum number of modules that can be included in a programme, and that only occurs when they are all half-modules). In itself this is not a time-consuming process, but it is a process that has to be repeated a number of times.

The key factor is the number of groups that have to be checked, to ensure they fit the Course Grid, and then evaluated.

Consider the situation when twenty-four normal-length modules have been identified. Only one programme is possible; re-arranging the modules within the programme does not constitute a different programme.

If one more module is added to the list, that extra module could replace each of the twenty-four other modules. Thus there are twenty-five possible courses; the course resulting from using the first twenty-four modules in the list, plus the twenty-four other courses that result from substituting the extra module. A simpler way of looking at this is to consider which of the twenty-five modules should be omitted; there are twenty-five modules which could be omitted so that twenty-four are left.

If twenty-six modules have been selected for consideration the number of candidate programmes rises to six hundred and fifty. Effectively, this is the number of ways of discarding two modules from twenty-six. There are twenty-six choices for the first module to be discarded, and for each of these there are twenty-five choices for the second module. The number of possibilities is 26 x 25.

With twenty-seven to choose from the number becomes 27 x 26 x 25, and so on. Every time a module is added to the list, the number of possibilities rises by a factor equal to the number of modules in the list.

The number of ways of selecting twenty-four modules from a number, 'N', of modules is given by the equation

Number of ways = N!/24!

(The symbol ! means 'factorial'. 3! = 3x2x1, 4! = 4x3x2x1, etc.)

Table 9.5.4 (on the next page) shows how quickly this number rises.

Table 9.5.4
Number of candidate programmes as a function of the number of modules selected.

Number of modules selected	Number of candidate programmes
24	1
25	25
26	650
27	17,550
28	491,400
29	14,250,600
30	427,518,000

In most cases these numbers will not occur, because the analysis assumes that all modules are independent of each other and can be discarded, if it is so desired.

a) Most programmes will contain groups of modules which cannot be separated; that is, where a module is an entry requirement to another. The programme that contains the second module without its entry requirement is not a possible programme and would not be tested.

b) If compulsory modules are listed, they cannot be discarded. A programme which omitted any compulsory module is not a possible program and would not be tested,

Nevertheless, if the number of modules identified exceeds the number required there is a real danger that the time taken to investigate every candidate programme would render the process unworkable.
To give an idea of the scale of the problem, allow one tenth of a second for the computer to select the candidate programme, attempt to fit it into the Course Grid, and evaluate it. With thirty modules to select from the process would take 42,751,800 seconds, or 71,253 minutes, or 11,875.5 hours, or 49.5 days of continuous computation!

9.5.5 The solution

The solution would appear to lie in a compromise. Up to this point we have

been looking for the 'best' course; we should now consider finding a 'satisfactory' course.
Before considering how that should be done, consider the first course that has been evaluated. It contains only modules relevant to the student's needs, because only such modules were chosen by the Selector. When the Sequencer operates, it tries to add modules to the Course Grid. These modules are not simply taken in turn from the list available; that might result in only one area of interest being included. The order in which the modules are selected depends on their flexibility; this in itself implies that modules from more than one area will probably be included. In addition, the most difficult modules to fit are sequences. Although they are considered early in the process, because they tend to have low flexibility, nevertheless some sequences may not be fitted. This leaves space for the more flexible modules, those that are independent or belong to short sequences. The probability is that these come from the other areas of the catalogue, when the student was looking for hobbies or recreational modules. It is probable, therefore, that even this first course will already contain modules involving progression (the sequences), and modules giving breadth (those from the hobby/recreational areas).
In short, any course that has been evaluated will satisfy most, if not all, of the needs of the student, and is therefore a 'satisfactory' course. There are two reasons for continuing to form different courses from the modules selected.

a) Another combination of modules may be better than already found.
b) To give the student a final choice between courses.

For the reasons stated above, the search for better courses can be stopped at any time after the first course has been evaluated. The single factor which will determine the number of courses examined is, therefore, time.

9.5.6 Is evaluation worth the trouble?

In Chapter 4.3.3, 4.4.3 and 4.5.3 reservations were expressed about the value of evaluating the courses. To summarise those arguments :-

a) In the Qualification-centred course the primary objective is that the student gains the qualification. Any course that achieves that is satisfactory, and the differences between such courses will be minor.

b) In the Higher education-centred course the primary objective is to gain entry to an advanced course. Any course that achieves that is satisfactory. While there may be more scope for variation than in the Qualification-centred course, the use that is made of that choice will be heavily dependent on the student's needs. A student who has been absent from full-time education for some time may need a more intensive course, to sharpen his skills in preparation for the advanced course. Another student may benefit from a broader course.

c) In the student-centred course it is the interests of the student, as expressed by his choice of job codes, that decides the merit of the course. A course containing very few modules related to the areas the student has specified is clearly of little value, but adding other modules from areas which the student has not specified will not make it any more valuable to the student.

In short, during the course of this project the author has become less convinced that the Evaluator is necessary. The principles behind the design of one, and the part it would play in selecting a notionally best course have been described. The decision to include it, or not, in a full scale version of the system must be the subject of further discussion.

9.6 The data files

9.6.1 SCE File

The file as described indicates each subject at the three grades, Higher, Ordinary and Standard. This was satisfactory when entry requirements were expressed as, for example, "a pass at Ordinary grade". With the increasing use of Standard grades some module descriptors now specify the grades within Standard grade(2). For example, a descriptor might include under Preferred Entry Level, " a pass in Standard grade English at grade 3 or 4". To allow for this, the SCE-CODES file should be extended, at least where Standard grade is concerned, to allow the student to indicate the actual grade awarded.

To allow for students with qualifications obtained in earlier years a more complete list, including those subjects or grades which have been phased out, would have to be provided.

9.6.2 The COLLEGE File

The COLLEGE file will be used by the college to indicate its policies in respect of compulsory modules and study time. The larger part of the file will contain the reference numbers of modules available within the college.

The COLLEGE file should contain

a) reference numbers of any modules deemed to be compulsory by the college. Given the reference number the system can find any other information it requires from the MODULES file.

b) the reference numbers of those modules taught in the college, or which could be taught if there was sufficient demand.

c) the number and distribution of any study time that should be included in the student's course.

It should discriminate, where possible, between compulsory study time, where the college feels that is essential that the student have some time away from classes, recommended study time, and maximum allowed study time. The latter will be used by the system to determine the maximum number of vacant cells that can be left in the Course Grid.

9.6.3 The QUALIFICATIONS File

If the student states that he wants to gain a recognised qualification, there has to be a way of identifying all the modules which, taken together, have been recognised as equivalent to the qualification; this is the file that enables this to be done. As was explained in chapter 2.10, these 'equivalences' can be considered as a series of lists;

List (a); the compulsory modules
List (b); a list of 'elective' modules.
The student must choose a specified number of modules from this list.
Possibly other lists similar to (b)

To allow the computer to interpret these lists correctly the following data will be required.

a) The number of compulsory modules.
b) The list of reference numbers for compulsory modules.
c) The number of elective lists.
d) For each elective list, the number of modules in the list
e) " " " " the number of modules to be selected.
f) " " " " the reference numbers for the elective modules

The following example will clarify this.
The student must take modules 1234567, 1234568 and 1234569. He must choose one module from 2234567, 2234568 and 2234569, two modules from 3234567, 3234568, 3234569 and 3234560, and one module from 4234567, 4234568 and 4234569
The corresponding data would be;
3 . . . . 3 compulsory modules
1234567, 1234568, 1234569 . . . . list of module reference numbers
3 . . . . 3 elective lists
3 . . . . .3 modules in the first elective list
1 . . . . 1 module to be selected
2234567, 2234568, 2234569....................list from which selection is made
4 . . . . .4 modules in the second list
2 . . . . 2 modules to be selected
3234567, 3234568, 3234569, 3234560
3
1
4234567, 4234568, 4234569

Using this data the computer knows the number of times it has to repeat the operation of offering lists of elective modules, how many modules to list each time and how many modules must be selected. Before presenting a list of electives the computer must check that it does not contain modules that have already been chosen; this happens in some equivalences. If a list did contain modules already chosen, these would not be re-presented.

Finding the correct qualification name would be facilitated by grouping them according to some criterion; in SCOTVEC 1986, eight groups were used, although these did not appear to match the categories of their module catalogue. Either the eight sector boards or the seventeen module categories would be suitable; the latter would have the advantage of allowing easy reference to the SUPERCLASS file and the identification of related modules, if required to complete a programme.

9.6.4 The ADVANCED file

The ADVANCED file holds lists of modules deemed to satisfy the entry requirements to more advanced courses. Little information is available from which to estimate the size of the file. In principle it has the same structure as the QUALIFICATIONS file.

9.6.5 The SCE-EQUIV File

When a student gives SCE subjects as previous qualifications, it is necessary to translate these into modular equivalents. The SCE-EQUIV file will be used for this purpose. It has to relate a subject name at one of three grades to the list of modules.
Each record will contain the name of the subject, the grade and the list of modules which, together, are deemed equivalent.

9.6.6 The Module File Manager

Decisions concerning the type of file organisation will depend on the hardware on which the system will be run, and therefore no specific recommendations can be given. The organisation chosen, however, should take into account the frequency of alteration to the MODULES file, and the rate of expansion, which appears to be continuing unabated.

9.6.7 The MODULES file

The MODULES file should include at least the Module Summary, the plain

language description of the module. This has to be made available to the student so that any decision to accept or reject an offered module is based on some knowledge of the module contents.
In addition, it appears sensible to link the MODULES file to the file of Module Descriptors, if these are not included as part of the module record. The Module Descriptor will not be used by the suite of programs designed to construct a course for a student, but it would be useful to have the Module Descriptors linked to the other files, or at least, to the MODULES file.
9.6.7.1 The job codes in the module records should be as specific as possible. Because of the structure of the SUPERCLASS system it is easy for the computer to move 'up' a level to a less specific class; this class will contain the more specific level already identified, and may refer to more general modules on the same topic. (See Chapter 9.3.9)

9.6.8 JOBS-MODS

In the prototype, every job code has an entry in the JOBSINDEX file. As the number of job codes included in the system increases, the JOBSINDEX file will also increase in size. The JOBSINDEX file must be read sequentially from the start until the job code is found. To prevent this slowing down the system, the JOBSINDEX file can be abbreviated; Only the first and second level job codes need be given. When a job code is given, for example A862, this is shortened to A8. This entry is then located in the JOBSINDEX file, and a jump is made to the relevant entry in the JOBS-MODS file. From that point, (A8), the JOBS-MODS file is searched, sequentially, until the subdivision, (A862), is located.
To illustrate the procedure, consider Figure 9.6.8.1, below.
(The data associated with each JOBS entry have been omitted for simplicity.)

Figure 9.6.8.1

Indexing the JOBS-MODS file

JOBS-MODS File		JOBSINDEX File		
position		(a)	(b)	
0	A	A	0	In the JOBS-MODS
3	A1	A1	3	file, the category
:	:	A2	103	shown in column (a)
103	A2	A3	432	starts at the location
:	:	A4	700	shown in column (b).
432	A3			
:	:			
700	A4			

There are seventeen main classes. These are divided into a number of more specific classes as shown below.

Main classes	a	b	c	d	e	f	g	h	j	k	l	m	n	p	r	s	t	= 17
number of sub-classes	9	3	9	8	8	9	9	6	7	9	8	9	9	6	8	7	9	= 133

Thus there would only be 150 records in the file. Reading from the start of the JOBSINDEX File to any position is not a lengthy process, as the file is quite short.

This approach, referred to by Hanson as partial indexing (Hanson p260), appears to be standard procedure on mainframe computers, but this is for technical reasons concerning the discs in such systems. It is not clear whether such considerations apply equally to micro-computers. It is almost certainly the case that, provided the index can be read in memory, a full index will locate the record faster than a partial index followed by a small search of a disc.

If it was felt necessary to have an entry in the JOBSINDEX file corresponding to every entry in the JOBS-MODS file, as was done in the prototype, the number of records would increase rapidly. At each level, other than the first, it is possible to have nine sub-divisions. The number of records, and the memory occupied (in the BBC Master) for the first three levels are shown in Table 9.6.8.1, below.

At level 3 the job code contains three characters; for example A12. This field therefore occupies 5 bytes (2+3). The position field occupies another 5 bytes (being an integer), so each record occupies 10 bytes. At level 4 there could be nine times as many records and each would occupy 11 bytes, etc.

Table 9.6.8.1

Memory occupied by JOBSINDEX file

level	records	record length	memory required	Total memory
1	17	8	136	136
2	153	9	1,377	1,513
3	1377	10	13,770	15,283
	1547			

At level 3 it can be seen that there would 1,547 records, if all classes had an entry. This would reduce the speed at which the position of a job code could be located in the index file. In addition, for maximum speed of use, the index file should be kept in the computer memory, where searching is considerably faster than when it is read from a disc (Hanson 1982 p133). At level 3 this would require about 15 kilobytes of memory to be allocated to the JOBSINDEX file. However, there are two methods that could be used to speed up the process without using too much memory.

a) The index file is kept in SUPERCLASS order, as explained above for the JOBS-MODS file. As this is a known sequence (it is in fact alphabetical order), various methods can be used to locate records rapidly while keeping the file on disc rather than reading it into memory.
b) The index file could itself have an index. This second index file might contain only the first two levels of job code, i.e. 170 records, and would occupy only 1.5 kilobytes of memory.

In the prototype the SUPERCLASS and jobs-mods files were kept separate. This was intended to simulate the operation of the 'official' SUPERCLASS file. However, it is now felt that the system should have its own copy of SUPERCLASS, because :-

a) As it is unlikely that modules will be available for every SUPERCLASS code the system version will be shorter than the official version.

b) Printing or displaying the system version will only show names for which modules are available, which avoids the irritation to the student of choosing a job name only to be told there are no modules available for it.

c) If the 'map' display is to be used (see Chapter 9.3.8.2) it is probable that it can be implemented more easily if the SUPERCLASS file is in a format designed for the purpose.

If the system does not use the official SUPERCLASS file, the names can be combined with the JOBS-MODS file, so that in response to a job code it can return not just the reference numbers of relevant modules but the name of the job as well. In the prototype, checking the job name slowed down the operation of the Selector program. Whether this approach is adopted depends on (c) above.

9.6.9 If a modular programme has been partially constructed on the basis of Qualification, and a search is being made for a similar group under Higher Education, the search time can be reduced if

i) all module lists, whether under Qualification or Advanced, contain modules arranged according to the same criterion. The most obvious would be in order of reference number;

and

ii) the qualifications and advanced education lists are coded as to career area. If this is done, then rather than search all advanced education lists, only those lists in the same area as the qualification need be searched. SUPERCLASS would appear to be the most suitable coding system.

9.7 Updating data files

During testing of the prototypes a number of problems were encountered because data files were updated manually. Some examples are :-

a) in one case a module was quoted as an entry requirement, but there was no record of the module in the MODULES file.

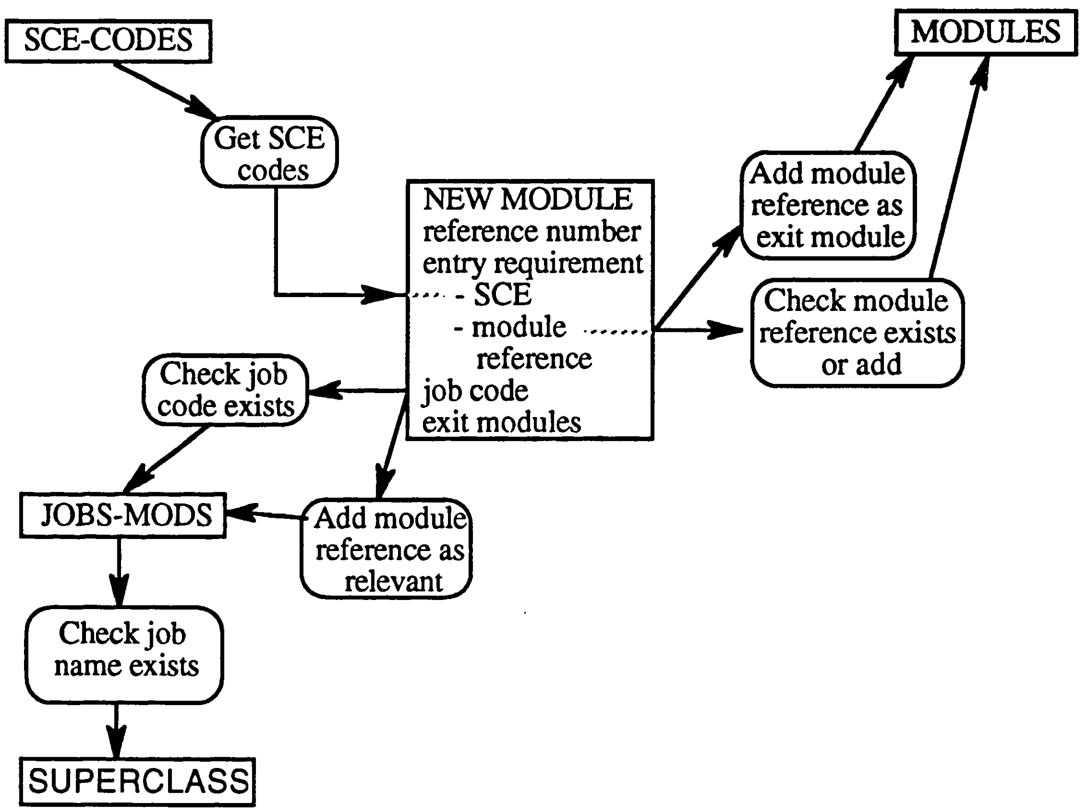
b) A job code might be given for a module, but the code did not exist in the JOB-CODES file, or its name might not appear in the SUPERCLASS file.

9.7.1 Adding a new module

Most (but not all) of these faults can be rectified by the use of a suitably designed MODULE file manager program. The file manager will prompt the operator for the information about a new module :- title, reference number, entry requirements, etc. (See Chapter 7.7.4.2). This program should perform automatic checks on some of these entries, as shown below in Figure 9.7.1, and explained, briefly, below the figure.

Figure 9.7.1

Checks to be made when adding a new module



9.7.1.1 Get SCE code

Entry requirements can be specified in terms of SCE subjects. The codes for these can be obtained from the SCE-CODES file.

9.7.1.2 Check module reference exists

If a module is quoted as an entry requirement the MODULES file must be checked to ensure that the relevant module record exists. Where it does not a record must be constructed. To allow the current new module record to be completed the reference number which must be added can be placed in a temporary file. When the current record is completed the operator would be prompted to supply details of the quoted module, adding it as another new module record (and performing the same checks).

9.7.1.3 Add module reference as exit module

If the record for module 'B' is being added, and it quotes module 'A' as an entry requirement, the record for 'A' should be amended to show 'B' as an exit module.

9.7.1.4 Operations on the JOBS-MODS file

When a job code is quoted for a module the JOBS-MODS file must be checked to ensure that the job code exists; if not, the code should be added to the JOBS-MODS file and the operator prompted to give the job name, which is added to the SUPERCLASS file or directly into the JOBS-MODS file if that is to contain the job names. (See 9.6.8)

9.7.2 Links between other files

The four files, QUALIFICATIONS, ADVANCED, SCE-EQUIVALENTS and COLLEGE all contain module reference numbers. When these files are being constructed or amended each reference number should be checked against the MODULES file to ensure that a valid number has been given. Failure to find the record may indicate a typing error in either file (assuming that the qualification does not refer to modules that are not yet in the catalogue).

9.7.3 All the comments given above apply to adding new records to files. The MODULES file manager must also consider the case of modifying an existing record. In particular, if a module is revised its reference number may be changed (see 4.5.7). Every reference to the original number, in any file that quotes module reference numbers, must be changed accordingly. As some modules (those concerned with basic skills) may be referred to on many occasions, particularly in the QUALIFICATIONS file, this may be a slow process. An alternative approach may be to keep an 'update' file which relates the old reference number to the new. On failing to find the old number in the MODULES file, the update file would be checked to find the new reference (if one exists - as stated above, failure to find a reference number can indicate a typographical error). As the update file grows, a convenient time can be chosen to make the relevant alterations to the other files and remove the reference from the update file.

9.8 The complete system

The interaction of the four components, (Constructing the Pre-entry List, the Selector, Sequencer and Timetabler), were described in 6.6.4 and these comments will not be repeated. In designing a programme for a student the Selector and Sequencer must be used alternately as the Course Grid fills. At the end of a block of work the student should only have to confirm his success in the suggested modules to update the Pre-entry List ready to design the next stage of the programme.

9.9 Notes

1) In this case the 'operator' is the person who decides which job codes should be linked to the module - probably the person who wrote the Module Descriptor.
2) The use of 'grade' to mean both the examination, e.g. Standard grade, and the level of award within that grade, e.g. grade 1, is unfortunate.
3) The space requirements are for the BBC Master computer.

Chapter 10
Summary, Conclusions and Recommendations

10.1 Summary and conclusions

10.1.1 The National Certificate, and the system of further education to deliver it, were established by the 'Action Plan' in 1983. The Plan laid down the principles on which courses should be designed; it stressed that the existing courses were too restrictive, and that the new courses should take advantage of the flexibility possible in the modular approach, which it established. To do this, students should be actively involved in the design of courses to suit their needs; they should be able to choose modules through a process of negotiation.

10.1.2 Following from these principles, the factors affecting course design have been described and developed. In particular, the success of the Action Plan in attracting adult returners to further education, and of making modules available in secondary schools, has resulted in the National Certificate being offered to a less homogenous student body than was previously served by further education alone. Students attending further education colleges vary in their previous educational background, and in their reasons for participation; fixed courses of pre-determined modules do not satisfy the needs and interests of a large number of potential and actual students, with the result that the dual objectives, of further increasing participation in further education, and of preparing the future workers of this country for the changing demands of the work-place, will not be fully realised.

10.1.3 The first full courses within the new structure became available in session 1985/86. Courses were phased in over a period of time, and gradually replaced the older, fixed courses. A number of studies have shown that, for various reasons, while new courses and qualifications have come about, taking advantage of the new approach, very few students have been involved in the design of their courses. This has been confirmed by the interviews conducted as part of this work. The reasons for this failure to meet the demands of the Action Plan, in respect of the provision of choice, centered on the lack of training of staff in the skills of negotiation, the failure to convince the relevant staff of the benefits of choice, the frequent changes in the SCOTVEC catalogue, and the number and scale of the other tasks which had to performed; all of which contributed to a climate in which the full implementation of the Action Plan was unlikely. These failures can themselves be linked to the speed with which the Plan had to be implemented, and the lack of resources, even in the short term, to address the problems. The adverse attitude of staff can be traced, at least to some extent, to the administrative difficulties associated with the provision of choice, and the lack of time in which to carry on the process of negotiation. Detailed consideration of the terms, 'choice' and 'negotiation', show that neither can be applied indiscriminately; in particular, choice, linked to negotiation, does not imply free choice.

10.1.4 When the process of selecting modules for the individual student was developed, it was found that the process would be extremely time consuming, to the extent of being impossible, if it were attempted manually; and that was what college staff had in mind when they considered the process. The assistance of a computer system designed for the purpose could significantly reduce the time taken, while still achieving the balance between allowing the student choice and following the principles of course design.

10.1.5 The outline specification for such a system has been developed and certain parts have been prototyped. Those parts of the system which have not been prototyped have been described in sufficient detail that they could be constructed.

The design considerations in the prototype have been explained and the program has been evaluated, as far as was possible given its limited objectives. It has been shown that a system based on the proposals is feasible; that it would help in the processes of choice and negotiation; and that such a system would contain the core components for a larger system which could reduce the administrative burden.

10.2 Recommendations

Recommendations have been made in the text concerning the development of the proposed system. These may be summarised as follows.

10.2.1 If the argument of the Action Plan is accepted, that involvement in course design will help to motivate students and develop an adult sense of responsibility for their own actions, then more choice must be made available to more students; many are already served by the present arrangements, where choice tends to be restricted by outside agencies, but the natural domination of further education by vocational considerations should not be allowed to prevent other students, those not seeking specified qualifications, from benefitting from participation in further education.

10.2.2 The components of the system described in Chapter 6 should be established; it is hoped that the methods explained in Chapter 7, the observations of student responses to the prototyped parts given in Chapter 8, and the recommendations and suggestions given in Chapter 9 will serve as a guide to the design of the system.

10.2.3 The requirements specification was given in 6.2. Following the discussion in Chapters 7 and 8 the specification can now be extended and given in more detail. It is assumed here that proposed programmes will be evaluated to indicate the best from those generated, and that split modules will be allowed. The reworded specification is given below.

1) The student will specify his interests by identifying relevant job codes taken from the SUPERCLASS system (or a file based on it).
2) The output from the system must consist of :-
   a) lists of relevant modules - possible programmes to be followed by the student.
   b) a Course Plan showing the order in which modules must be taken to

comply with the requirement that before a module is studied any prerequisite modules (entry requirements) have already been completed.

3) Each programme must be capable of completion within the time available to the student.
4) The programmes should not include
   a) modules which the student has studied previously
   b) modules which are equivalent to other qualifications the student may have (e.g. SCE subjects)
5) The user interface must be designed for operation by students who may be assumed to be novice intermittent users.
6) The system should include documentation describing the SUPERCLASS system and the way in which the students should use it to identify job codes.
7) The system should be operable in three parts :-
   a) Construct the Pre-entry List
   b) the Selector and Sequencer programs
   c) the Evaluator
8) Part (a), used to construct the Pre-entry List, should be designed with the knowledge that students confuse the names of some SCE subjects offered at different levels.
9) The Selector sub-program must be designed to accommodate at least twenty job codes.
10) The Selector sub-program must be able to accommodate sixty modules.
11) The interface between the Selector and Sequencer sub-programs must be designed to allow the student to alternate between them quickly.
12) The interfaces between the three parts must allow the student to leave the system after completing each part and re-start later without having to re-enter information supplied earlier.
13) The splitting of modules should be under the control of the student.
14) All parts of a module, whether split or not, must be studied in a continuous block (i.e. the parts must be in adjacent columns or rows in the Course Grid).

15) A module cannot be split more than once.
16) Compulsory modules must appear in all programmes for the student.
17) The student should be able to mark some modules as highly desirable, and these modules should appear in programmes in preference to modules not so marked.
18) The programmes must be capable of being modified, at a later date, without the need to re-supply information given earlier.
19) The evaluation of each programme should include :-
    a) the degree of breadth provided by the programme;
    b) the degree of articulation between modules within the programme;
    c) the degree of articulation with modules in possible later programmes.
20) The time required to learn how to use the system should be minimised.
21) The time taken to produce evaluated programmes should be minimised. The times given below should be used as a guide.
    a) 10 minutes to complete the Pre-entry List
    b) 40 minutes to generate one programme.

10.2.4 The file manager for the MODULES file is a key component in ensuring that valid data on each module is recorded; in addition, this program should be designed to provide the links between the module file and the others, so that changes in the former are reflected in the latter. From the comments in Chapter 9 it can be seen that the MODULES file manager can be used to construct the SUPERCLASS and JOBS-MODS files as the module records are entered. (It is recommended in 9.6.8 that these two files be combined.)

10.2.5 The system described here is intended to facilitate choice and negotiation, but the files required for its operation would form the basis of a bigger system, which could usefully include record-keeping, certification and careers guidance.

10.2.6 As stated earlier, in Chapter 3.5, the system used by SCOTVEC to number the modules has varied over the years, and at present two different and incompatible systems are used. To avoid unnecessary complications in programming, all module numbers should be changed to the newer, seven digit system, as quickly as possible. It should be noted, however, that the prototype programs operated on both numbering systems without difficulty.

It is also recommended that the definition of the new 'seven' digit module reference number be changed to include the SUPERCLASS identifier. (At present the SUPERCLASS identifier is frequently omitted.) As there are only one hundred and thirty-three classes at the second level, as explained in 9.6.8, they would form a useful primary index to the MODULES file.

10.2.7 Changes that should be made to the existing files.

In order that the integrated system being described here could operate, certain changes will have to be made to the existing data-bases; where these do not already exist they will have to be created.
These changes will now be described.

10.2.7.1 SCE-CODES

Each year the SEB publishes details of all the examinations conducted by them (SEB 1990). The simplest way to construct and maintain the SCE-CODES file is to copy the list of names and grades of examinations into a simple data-base, and then assign each subject at each grade a unique and unchanging reference number. As subjects or grades change, the new versions can be added to the file. Subjects or grades which have been discontinued should be left in the file for some years, so that the examination passes of students returning to college after a break from formal education can still be identified.

10.2.7.2 COLLEGE

Each college should establish and be responsible for maintaining its own file. The file should contain information related to college policies, such as the names and numbers of any modules deemed to be compulsory for all students, and the amount of study time that is recommended or considered essential. This file might also contain any alterations to the values attached by the Evaluator program to the various criteria on which courses can be judged.

This file could also indicate which modules are actually being taught, or can be taught, in the college. This would be done most easily by listing the reference numbers of the relevant modules. However, a more reliable method would be to indicate in the module records (in the MODULES file) that the college can offer it. This avoids any possible errors in re-typing and keeping up to date the reference numbers.

10.2.7.3 SUPERCLASS

It has been recommended that the system should use its own copy of the SUPERCLASS file, and that this should be amalgamated with the JOBS-MODS file. This reduces still further the possibility of errors, by ensuring that all codes are linked directly with a job name.

10.2.7.4 JOBS-MODS

This file, relating SUPERCLASS job codes to relevant modules, can be constructed automatically when records are added to the MODULES file. The information should be supplied, initially, by the writers of the Module Descriptors. Modules are frequently written and revised in response to the perceived needs of students and/or industry. The module writers are in the best position to list the vocational areas to which the module is relevant. To avoid the danger of the module writers being too 'blinkered', too concerned with the original reason for writing the module, it should be possible for others to recommend additions to this list of vocational areas. This can be done by passing the Module Descriptor to the other Sector Boards, who would consider the relevance of the module to the area for which they are responsible. (The Sector Boards are the committees responsible for overseeing the use and development of specified parts of the module catalogue.) Lecturers who teach the module might also see its relevance to jobs or interests not already identified, and might inform SCOTVEC of this.

These additions would be made when the module is next revised.

10.2.7.5 QUALIFICATIONS

If this file is not already in a data-base, this should be done by SCOTVEC by listing the name of the qualification and the reference numbers of the modules considered necessary. Each qualification should be given a reference number by SCOTVEC. This number will be used internally by the system and, if quoted in literature issued by the professional bodies, would allow students to find the qualification among the many listed. The first part of the reference number should be a SUPERCLASS job code, which will allow computer assisted searches to be carried out. The qualification name and the relevant module reference numbers should be entered into the file by SCOTVEC as the final stage in the process of agreeing an equivalence.

10.2.7.6 SCE-EQUIV

This file should be maintained by SCOTVEC - in at least some cases SCOTVEC has specifically designed sets of modules equating to SCE subjects, and they would merely be 'listing' work they have done themselves.

10.2.7.7 ADVANCED

The recommendations for this file are the same as for the QUALIFICATIONS file.

10.2.7.8 MODULES

While SCOTVEC are reasonably advanced with this data-base, it will be necessary for additional information to be included, and all relevant data must be coded to allow computer analysis and to facilitate searching for relevant modules.

10.3 Using the system on a micro-computer

10.3.1 In trying to determine whether the system could operate on a desk-top computer it was first necessary to estimate the size of the system. The calculations in Appendix 11 are subject to a number of uncertainties and the result must be regarded as approximate. The system is estimated to occupy 8 Mbytes. Even the relatively old Macintosh Classic has a built-in 40 Mbyte hard disc and could accommodate the system with ease.

The data required to update the files annually occupy 600 kbytes, which can fit on a single disc. (The high-density discs used by the Classic can contain approximately 1.4 Mbytes of data - even the double-density discs can hold 780 kbytes). If a number of computers in a room are networked only one need contain the whole system - the hard discs on the other machines would be free for the other uses to which these machines are put.

It is therefore possible to envisage the system being distributed, initially, as a set of 6 discs, to be copied onto a hard disc. Thereafter, updating the data files would be achieved by distributing a single disc of alterations, to be copied into the system using a relatively small program to replace the originals with the updated version. The distribution cost would be offset by the cost of notifying colleges of these changes on paper (as is done at present), and is, in any case, negligible when compared with the cost of distributing the revised Module Descriptors.

10.3.2 To allow colleges to carry out preliminary interviews in, say, March or April, the catalogue of modules available would have to be published, and the data-base 'frozen', before that time. The data-base would contain information on all modules available for the coming session. The data undergoing development would be listed, but would be flagged as 'not available until....'. This would allow colleges to plan to include the module when it becomes available. At the appropriate time the modules recently developed would have the flag changed to 'ready for use'. This process (or something very similar) is already familiar to SCOTVEC. For example, in Chapter 3, describing the alterations to the catalogue, attention was drawn to a number of modules recorded in the catalogue as not being available at the time of publication of the catalogue, but which would be available by September of the same year.

10.3.3 A more flexible approach would be to allow modules to be added to or deleted from the catalogue at any time - as soon as the required work had been completed. Modular programmes could be constructed at the end of each block, taking into account the success (or otherwise) of the student and any alterations to the catalogue. In this way new modules would be used at the earliest opportunity, and students would always be making use of the most up-to-date catalogue (subject to the college 'flagging' the module to indicate that it was ready to offer the module). If a year's programme is constructed at one time (and this has the advantage that the student can see the whole pattern for the year), there is the danger that a module selected in May (at a preliminary interview), to be studied during the following December, may be deleted in August. This could have repercussions for the rest of the programme. This difficulty can be overcome if modules to be deleted are flagged with a 'use by' date, i.e. although identified as superfluous to the catalogue a module can still be used up to a certain date. Users constructing programmes at more frequent intervals would be warned that the module is about to become redundant, and would be directed to its replacement.

10.4 Postscript

10.4.1 The Success of the Action Plan.

The summary given above is necessarily brief. Unfortunately, brevity tends to lead to generalisation; in highlighting, particularly in 10.1.3, the failure to meet the needs of the Action Plan, the successes have been ignored. It is undoubtedly true that large numbers of students currently engaged in National Certificate courses are adequately served by the existing provision. For those interested in gaining a recognised award, the National Certificate has increased the number and scope of such awards. Awards are now possible at intermediate levels, allowing students who would previously have been unable to benefit from further education to participate successfully. The National Certificate can be used as a stepping-stone to more advanced courses, allowing older students, who may have been isolated from education for some time, to acquire, or re-acquire, the study

skills needed to allow them to progress. The principles of modular provision have been adopted in other areas not directly linked to further education; it is currently being considered as an approach to the problems of older pupils in secondary education.
Further education is not as inadequate as might be concluded from the summary given above.

10.4.2 A study of this kind necessarily produces a 'snapshot', or series of snapshots, of the situation over the period of the project.
The National Certificate is not a static system. Constant change, identified in Chapter 3 as a contributory factor to the difficulties surrounding the increased provision of choice and negotiation, has not been confined to the number of modules in the catalogue, or the numbering system used to identify them. All areas of the National Certificate have been, and still are, subject to change and revision.

10.4.3 It has been assumed throughout this work that the intentions of the Action Plan, in respect of choice and negotiation, were to be implemented. It is, of course, possible that since the introduction of the National Certificate this has changed; that choice and negotiation are considered lost causes, and that the emphasis has changed. No evidence was found to suggest this; but it is possible that, just as the National Certificate, and the structure by which it is delivered, has changed, so too has the educational philosophy behind it. This would be unfortunate; the Action Plan gave convincing reasons for encouraging choice through negotiation; there is no evidence to suggest that these reasons are no longer valid.

Appendix 1

Table 1

Gross enrolment ratios for third level education

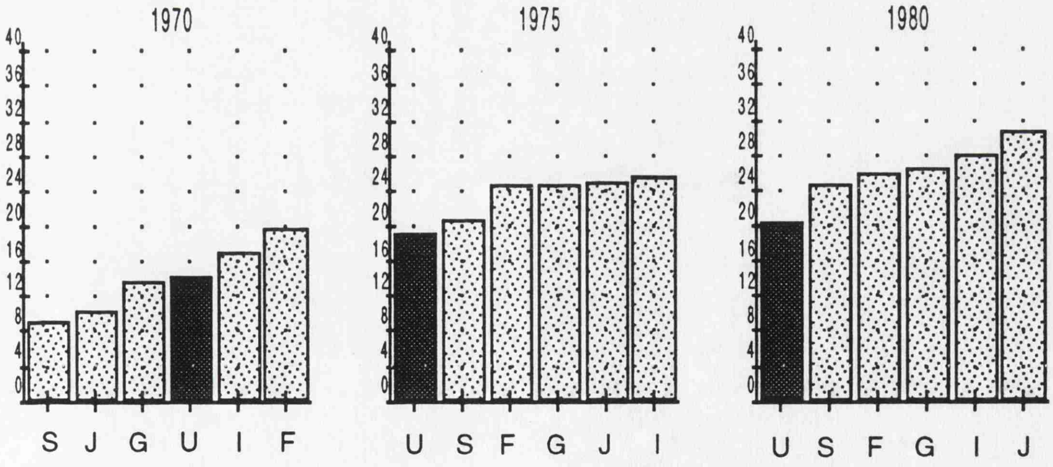
	Japan	France	Germany (Fed.Rep.)	Italy	Spain	U.K.
1970(1)	10.0	19.5	13.4	16.7	8.9	14.1
1975(1)	24.6	24.5	24.5	25.1	20.4	18.9
1980(2)	30.5	25.5	26.2	27.7	24.2	20.1

(1) Figures taken from UNESCO 1981
(2) Figures taken from UNESCO 1990
The 'Gross enrolment ratio' is the ratio of the number of student enrolments to the number in the relevant age group. UNESCO defines the age group for inclusion in the 'third level' of education as being from 20 to 24 years old. The third level of education comprises all post-secondary education.

These figures are shown diagrammatically in Figure 1, below.

Figure 1

Gross enrolment ratios for third level education



S=Spain, J=Japan, G=Germany, U=United Kingdom, I=Italy, F=France

In 1970 the United Kingdom was third, behind Italy and France. By 1975 the United Kingdom had fallen to sixth place. All six countries had increased the proportion of young adults continuing in tertiary education, but the United Kingdom was not responding as quickly as the others. By 1980 the other six countries had moved even further ahead. The 1980 figures would not been available at the time of publication of the Consultative Paper, but figures showing the continuing trend following 1975 would have been.

As a basis for understanding the Government's concern, the figures are less than satisfactory as they include all post-secondary education, rather than highlighting the situation in further education alone. However, if it can be assumed that these figures were representative of those applying to further education then it is clear that the United Kingdom had fallen behind its industrial competitors in its provision of post-secondary education and was not responding to the challenge to close the gap.

Appendix 2.1

Specimen Module Descriptor
(SCOTVEC Catalogue 31 May, 1985)

National Certificate Module Descriptor

Ref.No./Date	02105	Published Edition Spring 1985
Title	OFFICE ORGANISATION AND INFORMATION PROCESSING	
Type and Purpose	A Specialist Module which is concerned with the organisation of office work and with current developments in information technology. It will be of particular interest to the student who is hoping to obtain a supervisory position in business and administration.	
Preferred Entry Level	(a) 02100 Introduction to the Office and General Office Services, and (b) knowledge of business documents, e.g. 02101 Business Documents and Methods of payment, or 02250 Financial Recordkeeping 1.	
Learning Outcomes and	The student should: 1. know the role of the office as an information processing centre; 2. know and use the principles of forms design control; 3. know and use the principles of work control; 4. know the current applications of electronic technology to the office; 5. know the importance of a satisfactory working environment.	
Content/Context 5:	Corresponding to the above Learning Outcomes 1 to 1. Functions of the office; the office as an information processing centre; centralisation of office services. 2. Principles of forms design and control; common faults in forms design. 3. Introduction to the principles of work control; control over quantity of work performed and methods of avoiding bottlenecks; control over quality and identification of causes of errors; budgeting quantities of work performed within specific times and comparing actual with budget. 4. Introduction to the application of electronic technology to document preparation, message distribution, personal information management, and information access; the concept of the electronic office; factors affecting the rate	

	of introduction of office automation. 5. Physical conditions affecting health and efficiency of office staff; lighting, heating, ventilation, noise, decor, space; minimal standards required by the Health and Safety at Work Act 1974 (incorporating the Offices, Shops, and Railway Premises Act 1963); potential health hazards in the electronic office.
Learning and Teaching Approaches	Information technology is the fastest developing area of industrial and business activity in the Western World, and the new technology is affecting office work in particular. It is important therefore, for students to understand the functions of an office in order to appreciate the applications of modern technology to those functions. Whilst traditional procedures cannot be overlooked in preparing students for today's office, the teacher should take every opportunity to stress the changes taking place, for example, word processing for typing, communicating word processors for internal mail, electronic filing for filing cabinets, and company databases for directories. Practical work, group discussion and individual research should be encouraged. The use of films and other audio visual aids can also be valuable in this context provided they are as up-to-date as possible. A visit to an exhibition of the latest office equipment followed by constructive discussion would also be meaningful. A visiting speaker either from an organisation in the forefront of office technology, or from a specialist (e.g. from British Telecom on message distribution) should considered. Students should also be encouraged to read appropriate articles from magazines and textbooks. Previous knowledge gained in the use of business documents may be used as a basis for introducing forms control and work control. A detailed knowledge of legislation affecting the working environment is not expected, but reference might be made to possible disputes over poor working conditions and the role of the trade unions in such cases. Students should maintain a folio of completed work for assessment.
Assessment Procedures	This module will be internally assessed. Learning Outcome 1 will be assessed by 5 short written answers. Learning Outcome 2 will be assessed by examination of the student's assignment folio. Students should submit the best 2 assignments for assessment. One assignment should be concerned with forms design and one with forms control.

Learning Outcome 3 will be assessed by examination of the student's assignment folio.
Students should submit their best assignment for assessment.
Learning Outcome 4 will be assessed by 10 short written answers covering the content.
Learning Outcome 5 will be assessed by 5 short written answers.
The assessment should be carried out continuously and opportunities should be provided to rectify unsatisfactory performances.
The basis of an acceptable level of performance in the module will be satisfactory achievement of each of the learning outcomes.
Performance criteria for this module are clearly stated in its guidelines.

Appendix 2.2
National Certificate Module Summary

NO. & TITLE:
6170051 Administrative Procedures

ENTRY

Successful completion of modules 6170011 Clerical Procedures, 6170021 Communicating with Callers and Colleagues and 6170040 Information Technology - Office Applications, is recommended.

QUALIFICATION

National Certificate Module As Title

MODULE DETAILS:

This module develops administrative and information processing skills so that you can take part in planning and administration. On completion you will be able to find and handle information, make arrangements for travel, arrange meetings, deal with electronic information, maintain records and filing systems, print reports and create and maintain relationships with staff and clients.

ASSESSMENT:

To gain the module you will have to carry out a number of tasks to show that you have skills in dealing with a variety of administrative procedures. Your tutor will observe and record your progress during the module.

Appendix 3.1

The following case Studies are summarised from
"Choice in a modular system" John Hart and Sandy Pirie
SCRE Edinburgh June 1985

Case 1 Full-time Secretarial Studies

Prior to the start of the course, students were given a list of modules. Distinctions were made between those considered most essential, important and lesser importance. Advice was given about appropriate modules based on self-defined vocational destination and qualifications held or expected in August.--(1)
They chose modules for the 1st cycle (13 weeks). The results of SCE examinations were sent to a senior lecturer who re-assigned the student if the results were not as expected.
Cycle 2 - The senior lecturer devised programmes as a continuation of 1st cycle. Students discussed their work with 2 lecturers. They could opt out of part by arrangement with the senior lecturer.
Cycle 3 - Choice sheets were given out 2 months in advance. Students chose from a range of vocational modules.
Reasons for allowing choice - they understood the modular system, it was their last chance to choose, and they were more confident and responsible--(2)

"The object of the exercise is to devise the most suitable programme for the individual student."
"Negotiation is a difficult task to accomplish, in that it demands a balance between the exercise of professional judgment on what is appropriate for the student, and attempting to secure some degree of student autonomy; and, also, some students may be less than clear regarding their vocational intentions." --(3)
The vocational aspect was considered the essential criterion by both staff

and students.

The attitude of FE lecturers tends to be that the purpose of teaching is to impart vocational skills, and most full-time students are thought to take courses to obtain marketable skills. This contrasts with the Action Plan, which maintained that the vocational impulse had become less viable and less important than previously---.(4)

The function of Guidance was to find out the student's vocational intentions or to assist in clarifying objectives and to help students see what options were available.

Guidance at different stages is required.

Staff require time to ensure adequate discussion with students.

Notes

1) This being a course in 'Secretarial Studies', it is not surprising that the 'vocational destination' was the main criterion.
2) The modules in the course were all vocational. Reasons were given for allowing choice, as if this was unusual. No reasons were given for not allowing choice earlier. To say that students were given choice because it was their last chance is specious - why not give them choice earlier?
3) Negotiation is difficult. Students will be unclear as to their vocational intentions if they are not given careers guidance at an early stage.
4) The Action Plan was at variance with the attitudes of staff and students.

Case 2 Full-time Business studies.

Prior to the start of the course, the programme structure was explained. In each cycle there were five core modules and two vocational electives (chosen from 6), plus one personal and social development module.

Students were allowed to change modules within two weeks of the start of the first cycle.

To do a second cycle module, the student must have attained the relevant first cycle module.

A new module was introduced, which could be taken by some students.

"there was negotiation, although the amount of flexibility was minimal."----(1)
Similarly, in the third cycle there was little flexibility.
Time-tabling was crucial.
The electives were taught to all sixty to seventy students at the same time (2 afternoons per week).
For maximum flexibility they used one and a half hour units, not three hours.
The stated administrative difficulties were;

monitoring student attendance;
keeping track of their progress at any time;
identifying students with difficulties.---(2)

Lecturers were concerned about the criteria used by students in choosing modules, e.g. they wanted to stay with a particular group of friends, they were influenced by knowledge of the lecturer teaching the module, or they chose the easiest modules.--(3)

a) Articulation of modules leads to restriction of free choice.
For example, if 'Computing 1' is taken in the first cycle, 'Computing 2' should be in the second cycle - what if the student fails? (most second cycle modules have first cycle entry requirements).-------------------------(4)

b) Timetabling and administrative work was time-consuming.
Student timetables were only checked after they had chosen - they could have made bad choices, or their programme might suffer from lack of coherence. Where choice was limited, this is not a problem.
If students had more choice, programme guidance would have to be essential and regular. It was felt that this would make too many demands on staff time.---(5)

c) On a fixed course students have a clear sense of vocational direction, by discussion with staff.
In this college it was felt that student autonomy meant that they had a choice of course, but not elements within it.----------------------------------(6)
On a more flexible course the development of vocational intentions can be assisted more overtly by allowing the students to come to a 'curricular cross-roads' and have to decide where they are going and why.--(7)

Notes

1) Is negotiation meaningful if there is little flexibility?
2) The common difficulties associated with record-keeping.
3) Are lecturers justified in these low expectations about student motivation in choosing modules?
4) Second cycle modules should not all have entry requirements. To achieve flexibility it must be possible to re-teach some of the earlier modules.
5) Choice was restricted because of the time required for proper guidance.
6) On the old system students had the same choice - the Action Plan clearly meant something different by 'choice'.
7) If students were forced to choose from the start, they might develop their 'vocational intentions earlier.

Case 3 A full-time Pre-Nursing course

Staff identified the students' interests, vocational intentions, qualifications, and expected results.

Four basic programmes were devised, each containing vocational and non-vocational choice, and options to do SCE subjects.

Those not fitting into the basic programmes constructed their own from the modules in the four programmes. College wide non-vocational electives were available.

Note

This appears to be a good example of choice, although little is said about negotiation.

However, this might be the same course as that discussed in Appendix 3.4, Case 3, in which case the situation is less favourable.

Case 4

College policy was that each full-time course should contain twenty-four modules (8 in each of 3 cycles).

Of the eight, seven were to be vocational, with one from college-wide modules.

It was not compulsory to choose the eighth module, nor were they forced to choose outside the vocational area.---(1)

The difficulties were seen as;

getting student timetables (to identify free time);

achieving viable class sizes for the college-wide modules.------------------(2)

Of all eligible students, half chose a college-wide module, but there was a significantly higher drop-out rate for these (student commitment was not great).--(3)

It was decided to stop college-wide modules after the first cycle.

Notes

1) Only one 'choice' could be taken from outwith the department, and it was not compulsory to take it.
2) Difficulties with keeping records and timetabling.
3) Students probably assume that, if a module is optional, it cannot be very important.

Case 5

Full-time students took twenty-seven modules in the year - 6 from a college -wide selection (2 per cycle).

All periods lasted three hours.

Four periods were spent on the college-wide modules.

Each student was timetabled to be free in two of these periods.

In the first cycle some chose from within the vocational area. The policy was changed later; in the second and third cycles students must choose from outwith vocational area. --(1)

This policy was itself later amended; students could choose from their own

department if the vocational bias of the chosen module was distinct from that of the rest of their programme.--(2)
In the first and second cycles students were given handouts on electives. Also, in the second cycle a publicity stand gave information on electives. Students then gave their first, second and third choices.
In the third cycle each department had a named specialist and all full-time students had to discuss choices.

Notes

1) An attempt to force attention away from the vocational area.
2) Reversion to vocational priority, but at least a widening of perspectives.

Appendix 3.2

The following case studies are summaries of information contained in
"16+ Action Plan in Scotland"
Staff Development Project, Volume 2, Section 4 (CAST 1987)

Case Study 1 (p.36)

Dundee College of Further Education, Dept. of Catering and Hotel Management.

The session is split into 3 thirteen week blocks.

Block 1 consists of core modules. Blocks 2 and 3 contain 6 core and 2 electives from college wide provision.

There are no electives in block 1 because of administrative difficulties-----(1) and because it is thought that students are not sufficiently familiar with modular provision to make reasoned choices.-----------------------------------(2)

All programmes have Communication, Personal and Social Development, Numeracy, and Computer Literacy built into them.-------------------------------(3)

"Staff at first level guidance are aware of the constraints of time and expertise when dealing with difficult and time consuming guidance and support."--(4)

Additional information, on the college as a whole, is given on pp.40-54.

All full time courses consist of a minimum of 22-24 modules.

Modules are taught in three hour blocks, with eight modules being studied in each block.

For blocks two and three, once the six core modules are timetabled the two electives can be slotted into the remaining four half days.

Elective modules are generated from staff with spare capacity in their timetable, and modules able to take infill places.-------------------------------(5)

Students are told (p.46) that a module may be chosen for the following reasons :-

a) It is closely linked to subject area being studied.
b) It will give you a new skill

c) It is a subject or topic of interest to you.--(6)

Notes

1) Administrative difficulties given as reason for restricting choice.
2) Students need early induction period to explain the modular system.
3) A common core to ensure a basic breadth to the student's education.
4) Problems with the time required, and a recognition that skill is required for proper negotiation.
5) Infilling gives a wider choice, but is restricted to the extra number that the class can take.
6) These reasons are in line with the intentions of the Action Plan.

Case Study 2 (p.55)

Telford College, Catering Department

The session is split into 3 x 13 week blocks.

This is a large department, taking over 1000 new students each year. This means that it can exploit the advantages of a large-scale 'operation'. E.g. Communication modules are available in four different sessions per week. There is thus little danger of a clash with other core modules.-----------------(1)

Programmes consist of four elements :-

a) A recommended core,
b) General electives, drawn from other departments
c) Specific electives, drawn from their own department
d) Open learning opportunities.---(2)

This model of programme design is best for full-time students.

".... scope for choice is considerably more limited for day-release students." (p.55)

This case highlights the 'chicken and egg' situation.

The department does not have accurate information on demand until the students have chosen their modules - but the students cannot choose modules until the department offers them. The department therefore estimates demand and constructs a provisional timetable.

Provisional timetables are compiled at the end of March, to allow pre-entry

interviews to take place in the summer term.
(Re-working of the timetable takes place when student demand is known)
These interviews are with the Head of Department. The student then either proceeds to a second interview with a Senior Lecturer, or is offered guidance for other possibilities.--(3)
In the second interview the student selects the basic type of programme he wishes to follow. "It is open to the student to opt out of core modules, for example because he/she may already have gained certain modules or their equivalents." (p.56)
Some core modules are not vocationally-specific.(***)
The student then chooses general electives. Some are of general educational interest, some relate to personal interests. It is not compulsory to choose any from this group. (*But see ***, above.*)----------------------------(4)
The programme is completed by choosing specific electives from within the department, and from any available open learning opportunities.
Administration :-
Initially these arrangements were carried out without computer assistance. This was a "huge task". (p57)
A computer program was needed

a) to prevent double-bookings
b) to prevent overbooking staff
c) to give flexibility to allow changes to rooms, staff and modules as late as possible.
d) to devise individual student timetables when presented with the list of modules to be taken.

The updating of student records and completion of returns to SCOTVEC was a major problem.
E.g. One tutor has 18 students.
Each student takes 22 modules. The overall result is that information on any student is spread over 22 registers out of 500. The only method of gaining information on a student's progress was by physically searching the registers.
"It was difficult enough to achieve this for certification purposes; it was

plainly impossible to provide tutors with up-to-date records for guidance purposes."--(5)
A computer system now operates, whereby lecturers maintain registers, attendances and achievement by feeding into a central data-base. The tutor can obtain a print-out for any student at any time.

Notes

1) The advantages of size.
2) This appears to offer scope for choice and flexibility.
3) Early estimation of likely demand is required.
4) Students will be taking modules that are not vocationally specific. If they do not choose some from the electives, they will be taking some in the core anyway.
5) Administrative difficulties prevent adequate guidance. In some colleges this was the reason for reducing choice.
6) This is the 'obvious' solution to record-keeping problems.
   In general, this seems to be the best example showing how the ideals of the Action Plan can be achieved.

Case Study 3 (p74)
Integration in Electrical Engineering
The reasons given for students taking the programme were to :-
a) obtain a group of modules equivalent to previous qualification eg SCOTEC, SCOTBEC, CGLI
b) satisfy Industrial Training Board eg EITB, CITB, RTITB
c) gain skills to pursue personal interest
d) obtain recognised off-the-job competencies for MSC YTS

"Essentially, integration is the merging of the content of two or more modules, often re-ordering the learning experience into a more cohesive pattern." (p75)
The first example given is that of integrating two modules into one larger module, the learning outcomes of each being 'dovetailed' together so that they follow a logical teaching order. The two modules are 64302 : Electronic

Components and Combinational Logic, and 64303 : Wiring and Assembly Techniques.

The integrated module resulting is taught in the following order;

(LO = Learning Outcome)

64302	LO1	640303 LO4 (Safety)
64302	LO2	is stressed throughout.
64303	LO1	
64303	LO2	
64303	LO3 and 64302 LO3 Assessment using project	
64302	LO4	
64302	LO5	
64302	LO6	

"From our experience of integrating these two modules and using a project as the vehicle for learning, students without any previous electronic skills or knowledge can complete their study of these combined modules in approximately 50 hours. It can be seen that this represents some saving in learning time as well as increasing student motivation since the project generates greater interest than a series of unconnected exercises." (p76)-(1)

The second example integrates the twenty-four modules of the first year of a programme for students following an Electrical Installation Programme.

To give an indication of what is involved, the following serves as an example :-

"During each six week period of attendance, a central skill module is made the vehicle for coping with the support modules. By completing various tasks in the central skill module the student has to use skills developed in the support modules. An example of this is in 64112 which requires a student to erect cable tray systems. To secure them to walls and ceilings the student is applying skills required in 64121 Layout and Fittings, while he/she also uses ladders and scaffolding systems which form part of 64108 Work Safety." (p78)

In connection with a third example, the study states, "In order to avoid unnecessary duplication of content between the modules these 9 modules have been integrated into two learning schemes, one dealing with analogue aspects, the other with the digital aspects, of electronics."(p78)

Notes

1) This kind of integration reduces flexibility - the two modules cannot be taken separately.
2) Because of the links between the modules it is unlikely that any could be taken independently.
3) Again, the student only has a choice between the analogue and digital programmes - there is no choice within the programme.

Integration on this scale is effectively a reversion to the old style of course, where a fixed programme was offered. The only choice the student has is whether to take the whole course or not.

Appendix 3.3

The notes contained in Cases 1 and 2 resulted from informal discussions with lecturers in the Department of Business Technology and Administration in a large technical college. The conversations were guided by the content of the studies given in Appendices 3.1 and 3.2, with a view to identifying any important differences between the situation as reported in those studies and that pertaining to more recent times.

Case 1 Discussion with the Head of Department.
November 1987

1) The department operates 3 x 13 week blocks.
   In the first block we offer modules selected by the department. These are selected according to what the 'normal' Secretarial course , or Computing course, would contain.
2) Students still tend to think in terms of traditional courses.
3) 12 - 15 modules make up the first block. Students are put in groups according to previous qualifications (from school).
4) There is not a lot of negotiation in the first 13 weeks, because they don't know what they are negotiating, and we don't know until they arrive, what (qualifications) they are going to have.
   The programme is prepared within the first two or three days.
5) Even if they all wanted to do e.g. word processing, we couldn't because we don't have the resources. They might not have the entry qualifications in any case.
6) Students don't know very much about the process and cannot play a large part in the negotiation process.
7) Nowadays some students coming from schools have some knowledge of modules.
   This is creating an additional problem in that they may have already taken some of the 'core' modules. While the Region don't want small

groups, we have to appear (sic.) to give personal programmes.

8) Although students now have more idea of modules, they still don't play an active part in negotiation. They tend to think of it in sequential terms - Typing 1 goes on to Typing 2 etc. They have to be forced to go into anything outside the core group. They are very vocationally orientated.

9) In the second 13 week block we separate them according to aptitude, ability and desire. We offer a wider range and the original classes do not remain as such beyond a certain point.

10) There is a continuous system of guidance and counselling.
There are 1007 students in the department.
Every group has a councillor, who gets continuous feedback.
At the timetabling stage we know what is required.
They tend to negotiate in groups rather than individually. They want to stay 'with their pals.' Other factors are taken into account by student e.g. what would give them a Wednesday off.

11) There is no set time when the lecturer interviews with respect to the next part of the course. It is an ongoing process. We couldn't possibly sit down and re-interview 500 pupils and re-timetable.
It's a theoretical ideal. We have not been able to achieve it in practice.

12) We completely change the modules every 13 weeks.

13) In the 3rd block there is a reduction in the numbers of students. They tend to home in more practical situations. e.g. in Computing they've done the 'applications', now they want to do specific data-bases, spreadsheets, accounting packages etc. They want more depth now.

14) We've probably taught every Business Administration module , a lot of the inter-disciplinary modules, and most of the Computing ones. There aren't that many unusual ones. One or two areas might give more problems e.g. some aspects of Public Administration, or "Setting up a small business".
There are very few modules in our area that we could not teach. We are teaching at advanced levels as well, so we have a wide spectrum of staff. (32 full time staff.)
More problems are caused by restriction on equipment, e.g. computers.

It's an organisational problem.

15) There are about 130 YTS students. Small numbers are 'in-filled' into existing courses. Sometimes a larger number come from one Managing Agent. They can have a specific course designed for them.

16) Senior lecturers are more responsible (than the Head of Department) for designing courses for students.
Senior lecturers spend a lot of their time fitting students to modules - its an everyday occurrence.
It has become an instinct. It's more of an art than a science.

17) Some modules are done by open learning, and in evenings, which gives more flexibility.

18) 10% of the problems are Educational, the other 90% are organisational and administrational.

19) Every student is interviewed in between April and June.

20) Some apply for HNC courses on the basis of expected results, then find they fail some of their school subjects. They then want a modular course. This is a problem - do we put them through a complete modular programme? Do we advise them to go back to school to get the other Highers, etc.?

Notes

(1) and (2) If the courses are presented as fixed programmes, the students will think in those terms.

(4) No advance planning, interviewing or induction?

(5) This would have been the case under the old style of course. The second sentence seems irrelevant to the first. Are these excuses for not facing the difficulties of choice?

(6), (7) and (8) The modular system can be explained to those not familiar with it.

(7) Problems with resources restrict choice.

(10) Guidance may be continuous, but it does not start early enough.
Do students really allow these peripheral factors to significantly affect their choice of module?

(11) Considerable time is required for negotiation.

(13) This is a necessary consequence of articulation.

(14) The large department has advantages, but these must be exploited by an efficient administrative system.

(15) The course seems to be designed for a group of students, rather than for individuals.

(16) 'students are fitted to modules' - should this be 'modules are fitted to students'?

Choice is time-consuming.

Is the 'art rather than a science' used as an excuse not to face the problems of organising it effectively?

(18) More work needs to be done to solve these administrative problems.

Case 2 Discussion with Senior Lecturer in charge of YTS students.
January 1988

1) YTS students join courses on three different occasions; August, December and April.
   They start within the first three weeks of the start of a cycle (there are three cycles per year).
2) There are about 120 YTS students in the department.
   The minimum number in a class is ten, the maximum is twenty (twelve is supposed to be the smallest viable class).
3) Sometimes the lecturer goes to the work place - this allows smaller numbers because the managing agent is paying the fee.
   Some agents are quite keen on this because YTS students see the relevance between work and the module.
   This is happening more and more.
4) Small numbers - how do you choose courses?
   The Managing agent is now called the Approved Training Organisation. We go to the firm and see the students and the Training Officer. We choose suitable modules.

Students have a period of induction before their arrival at college.

Some firms send the students to college for induction - it lasts about 1 hour.

They all do the same type of work, mostly clerical.

5) Larger numbers - how do you choose courses?

Where there is a large number from one firm we give them a full day during the summer holidays to explain the set up, tell them the modules, and they discuss them. It's a well worn procedure.

6) What modules are offered in the 1st block?

In the first block we stick to General modules.

In the 2nd and 3rd we offer general and special modules.

All Business Studies modules are offered.

If there is one that we don't run, and they request it, we will run it if there are enough students.

We can run virtually any of the General modules.

We offer all the modules that are offered to full-time students, because we know these classes are already running.

We also run day-release classes.

YTS students are frequently mixed with full-time students.

7) What modules are offered thereafter?

About 8 weeks into the 1st block, we give a list of what is being offered to the Managing Agent - they speak to the students, then the lecturer goes and explains any modules that are not obvious.

Others can be offered if numbers permit.

We sometimes hope to run a module - if we don't get the numbers we drop it.

8) How long do they attend college?

2 year YTS = 65 days day-release 2,2,1

(2 days per week in 1st, 2 in 2nd term, 1 in 3rd term)

1 year = 35 = 1 day per week for 3 terms

(in the 2nd year of the 2year YTS they also do 1)

9) Are they interviewed?

We have Programme Review Meetings - at least every 3 months - This

involves the MSC, college, trainer's union, students' representatives.
We form the plan of next modules.

10) What part do they play in the negotiation?
We give them the list of possible modules, and go through their qualifications.
If they have O-grade Accounts they don't need to do Financial Record Keeping, although their employer may want them to because there is a computing element which is not in the O-grade.
I try to advise them on the basis of what they have got and what they should be aiming for.
They get a 'plain language' guide to modules - this contains entry requirements, content, and exit behaviour.
The titles of modules are frequently misleading.

11) There isn't so much negotiation in the first term. A lot have done modules at school, so they have some idea what's involved.
In the 2nd block they are good at telling you which modules are useful.

12) Do they have higher aspirations?
Some hope to be doing an HNC after YTS.
If they do 12 relevant modules they can get on the HNC course. They are told which modules are relevant. They take this into account.
They frequently hope that if they don't get a job they will go on to do a higher qualification, full-time.
We don't steer them away from modules that are a dead end because they will do 16 modules and only need 12 for the HNC.

13) Why are they attending the college?
The majority would have attended even if they were not on YTS.
They are looking for a recognised certificate.

14) What is the completion rate like compared with other students?
Very good. If they fail, its usually because of poor attendance.
That's assuming they are well fitted to modules. You have to take into account whether they shoula be doing the modules in the first place.

15) Are they re-interviewed?
They are re-interviewed every six to seven weeks.

They are not always interviewed together.

If I know that there is a group together I ask along 3-4 from one employer. I tell them what the available modules are. They come back next week.

16) Are there any cases where you felt you had arranged a particularly suitable course?

Not in particular. We try to ensure that they are all relevant.

See Appendix 3.4, Case 4.

17) Have you had any difficulty in arranging courses?

See Appendix 3.4, Case 5.

18) How much time do you spend arranging courses, interviewing etc.?

All the time. I am in charge of 4 full-time courses and two day-release. I spend most of my time on YTS.

I would guess my time is split about half and half between administration and discussion.

It's tedious getting results quickly.

Students hand in work late. Employers want to know if the student has passed. I have to go to the lecturer. He says they will pass, if they hand in a piece of work. It takes a lot of time, checking.

I spend a lot of time working out fees.

19) Do you get feedback from students?

Quite a lot. They come and see me if they want to.

They are told at the start to decide within two or three weeks of the start of a cycle whether they want to continue with the modules. I try to slot them in somewhere else if they change their minds.

Success is still good even when they have missed three weeks of the module. They work harder. YTS students are really good in this department. This department gets better YTS students than some. We get the ones with a few O-grades. In other departments the YTS students are probably less motivated.

20) Have any students done modules at school? or before?

If they have 3 Highers we offer HNC on day release. With 3 Highers they would cope too easily with modules.

Many have done one or two modules at school, e.g. Introduction to Computers. We move them on from that, e.g. Dbase or Computer Applications. Some slip through; they don't tell us they've already done a module.
It's difficult to find out if they achieved all the learning outcomes, because they don't have the certificate. There is a breakdown somewhere. Certification takes a long time.

21) Do students have a main criterion in selecting modules (or expressing interest in modules)?
Not before the induction. After that they know what they want to end up with.
They're thinking of the job that they would like.
Across the professional studies modules assessment procedures are very similar, so that's not a selection criterion. I don't know about the computing modules.
There's not very much teaching, it should all be activity based. They won't know about the teaching method until they do the module. I try to explain about the modules, but I can only give details about those I know in depth.

22) Do you have returning adults?
A lot, usually for a half or full day.
"Financial Record Keeping 1" is popular with unemployed students. Most have worked in an office, but don't have qualifications. They hear about modules and come to the college. They are aiming for a recognised qualification to help with their job. They usually opt for one more, usually computing.

23) Does the Department use modules from outwith the department.?
Yes. Financial record keeping is done in other departments.
If we can do the module we teach it. We don't send them out of the department.
It shouldn't make too much difference because the standards are the same. We have internal moderators.
There are modules which they have to do which we don't teach e.g.

Communications is a core module - all students do it - in the General Education department.

What's in the Core depends on what you are thinking of. The SNC Business Studies has six in the core (6 for Secretarial, and for Computing). We only mention that if a YTS student wants to go on to more advanced study he will require them. That's the only time it's relevant. It's not just a case of modules replacing old courses. They have to keep these six in mind, but they are doing sixteen. If they want to go into a particular field they must remember the six.

24) What training have you been given in Negotiation?

I took up responsibility in April. I have no training. I was involved in YTS before when I was a Senior Lecturer 1 before (I am now a Senior Lecturer 2).

25) Have you had any training in course construction?

No. I feel I've done a good job, but training would have helped.

If I was given training now it would be irrelevant. If anyone took over from me they should spend time learning. I took over from a lecturer who is still here and she helps when required.

26) Do you use new modules (new to the catalogue)?

Yes. As soon as a new module arrives we look to see where we can fit it in. They arrive on the desk on a regular basis.

I pass it on to a specialist in the department, to see what it involves. It's not a question of whether we will offer it. It's only a question of time to work out the assessment and teaching. There's timetabled time to work it out.

27) Are there any complete sets of module descriptors e.g. databases?

a) in the college?

There is a complete set in the secretary's office.

I wanted to have a look at a module - I just went along and looked at it.

b) elsewhere?

No.

We were thinking of running a course in Public Administration. I looked up the catalogue and found the titles, then checked the module descriptors.

28) Have there been any significant changes since the department started offering modules?

Some assessments were pitched at too high a level, and some material was boring. We now improve rather than write our own.

We frequently think that a variation on a module would be better. Two in the department are doing that now. They'll submit it to SCOTVEC, and see if it is accepted.

We might not be aware of another module. We wouldn't try to make just a small variation .

29) Size helps choice. We have over fifty staff. A smaller college or department might have more restricted choice.

Notes

(2) and (3) When more money is available, smaller classes can be run. This would allow more flexibility.

(5) Modules are chosen by the college.

(6) This kind of flexibility is possible because of the modular system.

(9) The interviews do not seem to involve the individual students - the impression is given that modules are chosen for the block of students, not individually.

(10) Employers influence the modules chosen.

Information about modules has to be in plain English - the module descriptors are written for lecturers.

(11) Early induction would allow students to make informed choices in the first cycle.

(12) Articulation with more advanced studies, i.e. beyond National Certificate.

Provided the later courses do not have too many entry requirements, flexibility can be encouraged in the National Certificate.

(13) The Action Plan intended to reduce the vocational element .

(15) There seems to be little negotiation.

(18) Negotiation is time-consuming.

Working out fees could be done on a computer.

(20) Administrative problems relating to the interchange of information between schools, SCOTVEC and the college.

(21) Vocation is the main reason for taking modules.
Assessment procedures and teaching methods are mainly irrelevant to choice.

(22) The modular system encourages people to return to college, because they can take as few modules as they need.

(23) Some modules are included to broaden the students education.

(24) and (25) There was little training in negotiation and programme design.

(27) Module descriptors are only available on paper. Finding a suitable module without knowing its title would be extremely difficult.

(28) The modular system encourages revision of modules to improve them. Without central control over the writing or revision of modules effort is likely to be duplicated.

(39) Size is an advantage.

Case 3

These notes resulted from observation of the interviews in which students selected the modules for their programmes, i.e. the negotiation process in operation.

The course for which the students were choosing modules was a Pre-nursing course, designed to give girls experience of the nursing environment and related studies before they embarked on a full Nursing course.

April 1986

There is a published programme describing the course.

Eight half-days have to be selected.

Each half-day, certain classes are timetabled. The student chooses from those offered.

12 classes are offered, of which 1 is compulsory (re-named "Core")
An additional note states that, "These represent the study programmes most likely to appeal to pre-nursing students. Other options are available."

Comments :-
Of the 12 classes, 5 are modules. The others are SCE O- or H-grade courses.
Of the 5 modules, 2 blocks are compulsory (core) - the "Nursing Modules", and "Additional Nursing Modules".

After observing a number of interviews between the Head of the General Studies Department and the students, it became clear that :-

The first question was "Has this student already got the required entry qualifications for the Nursing course?"
If 'no', the student was immediately timetabled for the required classes - which were all SCE courses.
If 'yes', the next question was "What would you like to do, from these on offer?"
In some cases this meant re-taking classes e.g. Tuesday morning contained only Higher grade Anatomy, Physiology and Health, and O-grade Biology, which the student might have already taken at school.
Thus there was vary little meaningful choice, and where a choice existed the decision was mainly based on personal preference, - "what would you like to do". There were few suggestions that some would complement others, or be of use later. For example, one student chose "Car Maintenance" because she helped her father service the car, another chose "Food and Nutrition" because she liked cooking.
In the course of the afternoon, "Car Maintenance" was the only module chosen that did not appear on the 'suggested' selection.

The real purpose of this course was
(a) to gain the minimum requirements for entry to the Nursing course, and

(b) to give students experience of working with people, particularly the elderly or handicapped (these were the Nursing Modules).

Most of the student's timetable could have been filled by reference to her application form. The interview only served to fill up the remaining time, and the criterion was primarily interest and enjoyment.

Appendix 3.4

The five case studies reported here arose from the questions;
"Have you designed any programmes of which you were particularly
a) pleased
b) disappointed?"

Case 1

The student wanted to join the Police force as a cadet. The Police suggested that he was a little young, and should seek to improve his qualifications before applying.

The course arranged consisted of 2 SCE Highers, and modules in Sociology, Typing, Computing.

Comments by Tutor:
1) This programme avoided overloading him with SCE courses, and gave experience relevant to his future work.
2) This type of course, mixing SCE with modules, would not have been possible under the previous arrangements.

Notes:
Relevance to future career was the main deciding factor in the choice of modules.
The number of modules was chosen to achieve a reasonable workload.

Case 2

The student wanted to work in her brother's hotel.
The course chosen contained modules in Cooking, Food Service, Typing, Computing.

Comments by Tutor

1) This course gave the student experience of some of the aspects of Hotel work likely to be relevant.
   She might return later to take courses in Personal Presentation (i.e. as a receptionist) and some of the more business oriented modules.
2) There was never an 'equivalent' course under the old system.

Notes:

Relevance to future work was the main criterion.
The possibility of a later return to the college was considered, the modules chosen being those considered more immediately relevant.

Case 3

A Managing Agent for the YTS scheme sought a course for a number of students. Discussion showed that most of the students were employed, in various posts, within the Leisure Industry, e.g. Golf clubs, Sports Centers, Social clubs.
The course selected contained Catering (especially convenience foods), Food Service, Reception, First Aid, Life-Saving.

Comments by Tutor: This could not have been done before modules.

Notes:

The course was designed to widen the experience of the students to include other aspects of the work they were already doing.

Case 4

One boy now has a full-time job with Polaroid. He was working with Polaroid in the computing section, and wanted purely computing modules. This was very difficult but I got him in. It isn't what they are supposed to do. But he did get a job at the end of the day. He did six computing modules.

Notes:
Pressure from employer restricted choice of modules.

Case 5

We had a girl in at one time, who wasn't very able. She could only do General modules, couldn't cope with the Specialised modules. For example, "Local Economy" is a very general module - she coped with that alright, but not "Introduction To Economics".

I feel we should have been able to offer her more General modules. She was working in an office and they wanted her to do only Office modules. We could have offered more variety, but they didn't want that. I felt restricted .

This is the only one that comes to mind out of about 120 - it isn't bad.

Notes:
Pressure from employer restricted choice of modules.

Appendix 4

The SUPERCLASS System

"SUPERCLASS. Subject Structure for Learning opportunities.

Training Access Points Local Database Version"

Department of Employment 1989

SUPERCLASS is a subject structure for organising and searching databases and other information on training and education opportunities.
(DoE 1989, p1)

(Lower case is throughout. The word 'and' is always shown as '&'. Commas are not used.)

There are seventeen main classes. The letters I, O, Q are not used as class identifiers.

Main Classes

A	business & management
B	law politics & economics
C	arts crafts & hobbies
D	culture society & education
E	language communication & self help
F	music & performing arts
G	sports games & recreation
H	food catering leisure tourism
J	environment security health & safety
K	agriculture horticulture & animal care
L	sciences & mathematics
M	health & personal care
N	architecture & construction
P	computers electrical & electronic engineering
R	engineering production & industrial design
S	minerals materials & fabrics
T	transport services & vehicle engineering

Each class is further subdivided.

An example is shown below.

A business & management
- A.1 business & management (general)
  - A.11 international business studies
  - A.12 european business studies
  - A.13 business organisations (general)
  - A.14 management studies (general)
    business studies = A.14
  - A.15 administrative management
    - A.151 chartered secretaries
      company secretaries = A.151
    - A.152 charity administration
  - A.16 management services
- A.2 small businesses
  - A.21 cooperatives
  - A.22 self employment

: etc.

Up to 6 stages of increasing generality link each specific class with one of 17 main classes. (Ibid. p1)
If there are no courses on the very specific subject the client wants, they may like to see information on more general courses as a compromise. (Ibid. p3)

The system is revised from time to time. If a local user finds the need to add to the system, he can define a new number initially at local level. This new number can then be incorporated within the national structure, so that from then on the number will be used by all users to mean the same thing.
For example, under Modern European Languages (E.542), groups of languages are listed alphabetically under various numbers as shown below.
modern european languages E.542
a-d E.5422
f E.5423
french E.54231
g E.5424
german E.54241
greek E.54242
h-k E.5425

If a local user wanted a number to refer specifically to Italian, he would insert into the table "italian E.54251". He would add this to his own local data-base where it would be used by other local parties. If it were deemed necessary to have this number at national level, the number could be incorporated into the national data-base. Once it appears there, all users would refer to Italian by this number.

Appendix 5
The CAST Computerised Guidance System

This appendix describes the system viewed in February 1990 at the CAST offices, Jordanhill College.

The package contains a number of related programs.

a) Search for a module.

The database can be entered by any of three routes :-

1) by giving the module reference number
2) by stating a cognate group number
3) by giving keywords.

1) Module number

   Information on a specific module can be obtained by giving the reference number of the module.

   For each module, the data comprises the information in the Module Descriptor, to the end of the Learning Outcomes, i.e. Reference number, Title, Purpose, Preferred entry level, Learning Outcomes.

   In general, this information is shown on a single screen.

   In addition, information is available on 'local classes', i.e. the college using the system would enter the information on the classes normally run, and a contact name.

2) Cognate groups

   These are the 10 categories into which the SCOTVEC catalogue was divided.

   Having selected the category, the user is presented with a set the of sections within the category. From that level the user can see the number and title of each module in the section, and can then refer to the information on the module (see above).

3) Keywords

Each module has associated with it a number of keywords taken from the title and Module Descriptor. By specifying a keyword the user is given a list of the modules associated with that keyword.

b) SUPERCLASS

Having chosen a specific occupation by reference to the SUPERCLASS file, the user is shown a list of modules relevant to the occupation. Module information can then be obtained - see (a)1, above.

Alternatively the user can use an 'Indexed Search'. The user starts to type the name of an occupation. As he 'sends' each word, by pressing the RETURN key, he is given a list of the occupation names that satisfy the criteria so far. It is in fact an alphabetical list of occupations.

A third way of using the system is by specifying a recognised award. The program then supplies the list of modules required for that award.

c) Construct a client profile

This is an expert system which guides the user through a series of questions. It produces a copy of the user's answers. The user then takes this to a Careers Guidance Officer, who can advise on the choice of modules.

Appendix 6

SCE Equivalences - Examples
From "Memorandum on Entry Requirements to Courses of Teacher Training in Scotland 1989." SED HMSO Edinburgh

	Ordinary grade at C or above	Higher grade at C or above
English	61003 Communication 3	61004 Communication 4 and 81009 Literature 1
Mathematics	61057 Mathematics 3	61059 Analysis Algebra 2 61062 Calculus 1(A) and 61086 Calculus 1(B)
Modern Language		61421/2/3/4/5 Modern Language 3
Physics	69030 Physics 1 69031 Instrumentation and Measurement Techniques	69033 Physics 2:Mechanics: 69034 Physics 2:Waves 1: 69036 Physics 2: Electricity and Electro-magnetism 1: 69032 Physics 2:Heat or 69042 Models: and 1/2 module (still to be developed)
Chemistry	69050 Chemistry 1	69051 Chemistry 2 :Inorganic: 69052 Chemistry 2:Organic: 69053 Chemistry 2 :Physical: 69062 Chemistry:Laboratory Techniques
Biology	69070 Biology 69071 Human Physiology	69072 Cell Structure and Biochemistry 69073 Metabolic Pathways and physiological processes 69076 Homeostasis and Control in Animals 69077 Plant Physiology 69078 Genetic Investigations
Computing	61092 Computer Hardware 61093 Computer Software 61095 Introduction to Computer Applications 61105 Social Implications of Microelectronics and Information Technology 1 and one of	

	61111-5 Computer Programming Elementary	
Economics	62200 Local Economy	
	62201 Introduction to Economics	
	62203 Contemporary Economic Issues	
	62204 Organisation of Industry	
Accounting	62252 Financial Record Keeping 3	62254 Accounting 2
		62302 Costing 3
	62253 Accounting 1	
Secretarial	62109 Typewriting 3	62112 Typewriting 6
	62113 Audio Typewriting 1	62114 Audio Typewriting 2
	62100 Introduction to The Office	62100 Introduction to the Office
	62101 Business Documents and Methods of Payment	62101 Business Documents and Methods of Payment
		62105 Office Organisation
		62106 Secretarial Duties

Appendix 7

****
CONTENTS OF FILE.JOBS
File was last updated on Thu.10 Jun 1993.21:09:16
There are 110 non-zero entries.
****

JOBCODE	N	JOBNAME	MODULES
1	A3122	1	Report Writing
2	A318	1	Meetings and Committees
3	A81	1	Office Supervision
4	A821	3	Office Filing
5	A822	10	Business Correspondence
6	A831	1	Secretarial/Management
7	A833	6	Medical Secretarial
8	A834	5	Legal Secretarial
9	A835	5	Personal Assistant Skills
10	A841	12	Shorthand in English
11	A851	3	Audio Typing
12	A852	17	Copy Typing
13	A854	14	Keyboarding
14	A861	2	Telephonists
15	A862	3	Recptionists (business)
16	A87	2	Word Processing Skills
17	A883	1	Office Computer Appreciation
18	A886	2	Electronic Mail
19	A887	1	Office Copying
20	C21	2	Design (general)
21	C75	1	Ceramics (craft)
22	D21	2	Literature (general)
23	E11	5	Basic English
24	E12	3	Writing & Reading (literacy)
25	E122	3	Writing (literacy)
26	E25	3	Preparation for Training & Education
27	E62	2	Writing (authorship)
28	E623	2	Creative Writing
29	E65	4	Public Speaking
30	E66	1	Communication skills
31	F55	2	Music Appreciation
32	F92	3	Singing
33	F922	1	Singing (solo) and Voice Studies
34	F93	3	Musical Instruments (Performance)
35	F931	1	Musical Instruments: Keyboard
36	F932	2	Musical Instruments: Stringed
37	F933	2	Musical Instruments: Wind
38	F934	1	Musical Instruments: Percussion

39	G1	2	Sports Studies & Combined Sports
40	G32	1	Swimming
41	G321	1	Swimming (Learners)
42	G322	2	Swimming (Advanced)
43	G323	1	Water Polo
44	G324	2	Life saving (Swimming)
45	G361	2	Canoeing
46	G362	1	Sailing
47	G3625	1	Windsurfing
48	G364	2	Rowing
49	G4	2	Athletics, Gymnastics & Combat Sports
50	G44	2	Gymnastics
51	G441	2	Trampolining
52	G45	2	Weightlifting
53	G46	2	Combat and Martial Arts
54	G462	2	Wrestling
55	G47	2	Fencing
56	G61	2	Ice Skating
57	G63	1	Skiing
58	G65	2	Curling
59	G711	2	Association Football
60	G712	2	Rugby Football
61	G714	2	Gaelic Football
62	G72	2	Hockey and Related Games
63	G721	2	Lacrosse
64	G731	2	Basketball
65	G732	2	Handball
66	G733	2	Netball
67	G734	2	Volleyball
68	G741	2	Cricket
69	G743	2	Baseball
70	G752	2	Bowls
71	G76	2	Golf
72	G781	2	Tennis
73	G782	2	Badminton
74	G783	2	Squash
75	G81	1	Walking
76	G82	1	Climbing
77	G83	1	Orienteering
78	G881	2	Archery
79	G911	2	Table Tennis

80	L11	8	Science (General)
81	L12	8	Science Introductory Studies
82	L13	5	Science Laboratory Practice
83	L15	5	Research Methods
84	L21	2	Physics (General)
85	L22	2	Mechanics
86	L2232	1	Hydraulics
87	L23	2	Optics
88	L242	1	Cryophysics
89	L25	2	Electricity & Magnetism (Physics)
90	L251	1	Electronics (Physics)
91	L261	2	Nuclear Physics
92	L262	2	Atomic Physics
93	L263	1	Molecular Physics
94	L264	3	Radiation Physics
95	L31	1	Theoretical Chemistry
96	L35	1	Applied Chemistry
97	L36	1	Chemistry (General)
98	L411	2	Mathematics (General):Unspecified
99	L77	1	Navigation
100	L831	1	Biology (General)
101	M7	2	Psychology
102	N6	5	Wood & Woodworking
103	N61	3	Woodworking
104	P213	1	Analogue Electronics
105	P24	1	Optoelectronics
106	P236	1	Computer Hardware
107	P5	1	Information Technology / Computer Applications
108	P51	4	Computer Appreciation
109	R43	2	Product design
110	R69	1	Tools:Use & maintenace

CONTENTS OF FILE.MODULE ON Mon.15 Jun 1992.18:04:32

Number of modules in data-base = 184

A8	6170011	Clerical Procedures	2
A8	6170021	Communicating with Callers and Colleagues	1
A8	6170031	Reception	1
A8	6170040	Information Handling: Office Applications	1
A8	6170051	Administrative Procedures	2
A8	6170061	Handling Business Documents	1
A8	6170090	Introduction to Office Management	1
A8	6171011	Working in an Office	1
A8	6171021	Working in a Reception Area	1
A8	6171031	Processing Business Transactions	1
A8	6180011	Producing Text	1
A8	6180021	Text Processing 1	0.5
A8	6180031	Text Processing 2	1
A8	6180041	Text Processing 3	1
A8	6180051	Text Processing 4	0.5
A8	6180061	Text Processing 5	1
A8	6180071	Text Processing 5 - Medical	1
A8	6180081	Text Processing 5 - Legal	1
A8	6180090	Introduction to Word Processing	1
A8	6180100	WordProcessing Skills	1
A8	6180111	Audio Typewriting	1
A8	6180121	Audio Typewriting - Medical	1
A8	6180131	Audio Typewriting - Legal	1
A8	6180201	Shorthand Transcription 1 (50 wpm)	3
A8	6180211	Shorthand Transcription 2 (60 wpm)	1
A8	6180221	Shorthand Transcription 3 (70 wpm)	1
A8	6180231	Shorthand Transcription 4 (80 wpm)	1
A8	6180241	Shorthand Transcription 4 (80 wpm) - Medical	1
A8	6180251	Shorthand Transcription 4 (80 wpm) - Legal	1
A8	6180261	Shorthand Transcription 5 (90 wpm)	1
A8	6180271	Shorthand Transcription 5 (90 wpm) - Medical	1
A8	6180281	Shorthand Transcription 5 (90 wpm) - Legal	1
A8	6180291	Shorthand Transcription 6 (100 wpm)	1
A8	6180301	Shorthand Transcription 6 (100 wpm) - Medical	1
A8	6180311	Shorthand Transcription 6 (100 wpm) - Legal	1
A8	6181011	Using a Keyboard	0.5
A8	6181022	Producing Text from Audio	1
C2	61705	Two-Dimensional Design - Experimental Printmaking	1
C7	8357560	Art/Design Project 1 - Ceramics and Pottery	2
D2	7111021	Literature 1	1
D2	7111011	Introduction to Literature	1

E6	7110011	Communication 1 : Spoken Communication	0.5
E6	7110021	Communication 1 : Written Communication	0.5
E6	7110031	Communication 2	1
E6	7110041	Communication 3	1
E6	7110310	Practical Talking Skills	1
E6	7110320	Practical Writing Skills	1
E6	7110401	Creative Writing 1	1
E6	7110521	Public Speaking	0.5
E6	91245	Oral Presentation Skills	1
E6	7110330	Communication in Meetings	1
E6	91246	Writing Papers and Reports	1
E6	7110411	Creative Writing 2	1
E6	7110051	Communication 4	2
E6	7110051	Communication 4	2
A8	6180011	Working in a Reception Area	1
F5	81661	Listening to Music 2	1
F9	71619	Music Making - Solo 1 (Brass)	1
F9	71620	Music Making - Solo 1 (Strings 1)	1
G3	80100	Introduction to Aquatics	1
G3	80121	Canoeing 1	0.5
G4	80143	Athletics 1	0.5
G8	80141	Archery 1	0.5
G7	80142	Association Football 1	0.5
G7	80144	Badminton 1	0.5
G7	80145	Baseball/Softball	0.5
G7	80146	Basketball	0.5
G7	80147	Bowling	0.5
G7	80148	Cricket	0.5
G6	80149	Curling 1	0.5
G4	80150	Fencing 1	0.5
G1	80151	Fitness Training 1	0.5
G4	80153	Gymnastics	0.5
G4	80152	Golf	0.5
G7	80155	Hockey 1	0.5
G7	80154	Handball 1	0.5
G4	80157	Martial Arts	0.5
G7	80156	Lacross 1	0.5
G3	80159	Rowing 1	0.5
G7	80158	Netball 1	0.5
G7	80160	Rugby Football 1	0.5
G6	80162	Skating 1	0.5
G7	80161	Shinty 1	0.5
G7	80163	Squash 1	0.5
G7	80165	Tennis 1	0.5
G4	80166	Trampolining 1	0.5
G9	80164	Table Tennis 1	0.5
G4	80168	Weight Training 1	0.5
G7	80167	Volleyball 1	0.5
G4	80169	Wrestling 1	0.5
G3	80101	Diving 1	0.5
G3	80102	Life Saving 1	0.5
G3	80103	Personal Survival 1	0.5
G3	80104	Swimming 1	0.5
G3	80105	Synchronised Swimming 1	0.5
G3	80106	Water Polo 1	0.5
G8	80241	Archery 2	0.5

G7 80242	Association Football 2	0.5
G4 80243	Athletics 2	0.5
G7 80244	Badminton 2	0.5
G7 80245	Baseball/Softball 2	0.5
G7 80246	Basketball 2	0.5
G7 80247	Bowling 2	0.5
G7 80248	Cricket 2	0.5
G6 80249	Curling 2	0.5
G4 80250	Fencing 2	0.5
G1 80251	Fitness Training 2	0.5
G7 80252	Golf 2	0.5
G4 80253	Gymnastics 2	0.5
G7 80254	Handball 2	0.5
G7 80255	Hockey 2	0.5
G4 80257	Martial Arts 2	0.5
G7 80256	Lacrosse 2	0.5
G7 80258	Netball 2	0.5
G3 80259	Rowing 2	0.5
G7 80260	Rugby Football 2	0.5
G7 80261	Shinty 2	0.5
G6 80262	Skating 2	0.5
G7 80263	Squash 2	0.5
G9 80264	Table Tennis 2	0.5
G4 80266	Trampolining 2	0.5
G4 80265	Trampolining 2	0.5
G7 80267	Volleyball 2	0.5
G4 80268	Weight Training 2	0.5
G4 80269	Wrestling 2	0.5
G8 80122	Climbing 1	0.5
G8 80123	Hillwalking 1	0.5
G8 80124	Orienteering 1	0.5
G3 80125	Sailing 1	0.5
G6 80126	Skiing 1	0.5
G3 80127	Windsurfing 1	0.5
G3 80221	Canoeing 2	0.5

M7	7177011	Introduction to Psychology	1
R6	74701	Introductory Workshop Skills	1
R4	61795	Product Design 1	1
R4	1716	Drawing Skills: Technical Drafting 1	1
R4	1862	Three-Dimensional Design	1
R4	64804	Design Studies	1
N6	95654	Working with Wood - Household Accessories	1
N6	95653	Working with Wood - Garden Furniture and Fitments	1
N6	95655	Working with Wood - Hobbies	1
N6	95656	Working with Wood - Wooden Toys	1
N6	85300	Working with Wood 1	0.5
F5	81660	Listening to Music 1	1
N6	85639	Working with Wood 2	1
F9	71621	Music Making - Solo 1 (Keyboard)	1
F9	71622	Music Making - Solo 1 (Strings 2)	1
F9	71623	Music Making - Solo 1 (Percussion)	1
F9	71624	Music Making - Solo 1 (Woodwind)	1
F9	71625	Music Making - Solo 1 (Voice)	1
F9	71610	Music Making - Group 1	1
F9	71611	Music Making - Group 2	1
F9	71612	Music Making - Group 3	1
L7	90016	Land Navigation using Topographic Maps	0.5
M1	6190020	Medical Terminology	2
M7	7177031	Occupational Psychology - The Indivdual and Work	1
P5	81093	Introduction to Computer Software	1
P5	71092	Introduction to Computer Hardware	1
P5	81091	Introduction to Computers	1
P5	91090	Using a Microcomputer	1
L4	7180321	Core Mathematics 3	1
L4	7180331	Core Mathematics 4	1
L1	3161001	Introducing Science	1
L1	3161011	Measuring and Recording Science	0.5
L1	3161101	Science in Context 1	1
L1	3161121	Introducing Science Investigation Skills	0.5
L1	3161131	Science Practical Skills	0.5
L1	3161111	Science in Context 2	1
L1	3161251	Science Investigation Skills	0.5
L1	3161261	Experimental Procedures - Biology	1
L1	3161271	Experimental Procedures - Chemistry	1
L1	3161281	Experimental Procedures - Physics	1
L1	3161291	Experimental Procedures - Science	1
L2	3171001	Introducing Heat	0.5
L2	3171011	Introducing Electricity	1
L2	3171021	Introducing Radioactivity	0.5
L2	3171031	Introducing Mechanics	1
L2	3171041	Introducing Waves and Optics	0.5
L2	3171101	Radioactivity	0.5
L2	3171111	Mechanics	0.5
L2	3171121	Electricity	1
L2	3171131	Optics	0.5
L2	3171141	Analogue and Opto-Electronics	0.5
L2	3171151	Properties of Matter	0.5

Appendix 9
Information for Students
The Module Selector Program.

(Because of the binding requirements, these pages have been reduced.)

The Module Selector Program finds all the SCOTVEC modules connected with a particular job. To use it, you have to tell the computer which jobs interest you. The Selector program then finds the modules that will teach you the skills required for the job.

The program also has to do other things; for example, it would be annoying if the program listed a lot of modules that would be useful to you, and then you found that they cannot be fitted into a timetable. The program checks this and tells you if there is going to be a problem.

Stage 1

The first stage in using the program is to tell it your present qualifications. You may already have sat some SCE Standard grades and completed some SCOTVEC modules. The computer needs to know about these - you don't want do the same module twice, or take modules that are equivalent to a Standard grade subject that you have already passed.
The information you give is recorded on disc in a file. Later, when you have completed some more modules, you add these to the file so that the computer can find even more modules that will extend your skills.

To simplify the job of telling the computer about your present qualifications you should fill in the **Qualifications Form** which you will be given. Please make sure you list SCE subjects (Standard grades or Highers) separately from SCOTVEC modules. The computer handles these differently.

You should fill in the Qualifications Form now.

Once you have filled it in you can use the first part of the Module Selector Program.

While you are waiting for the computer you should read the next set of instructions.

Qualifications Form

Name :- ______________________ Number :- __________

SCE subject name	Exam	
Grade		
1 ______________________	___	___
2 ______________________	___	___
3 ______________________	___	___
4 ______________________	___	___
5 ______________________	___	___
6 ______________________	___	___
7 ______________________	___	___
8 ______________________	___	___
9 ______________________	___	___
10 ______________________	___	___
11 ______________________	___	___
12 ______________________	___	___

Give the name and reference number of any SCOTVEC modules you have completed.

SCOTVEC module name	Reference
1 ______________________	__________
2 ______________________	__________
3 ______________________	__________
4 ______________________	__________
5 ______________________	__________
6 ______________________	__________
7 ______________________	__________
8 ______________________	__________
9 ______________________	__________
10 ______________________	__________
11 ______________________	__________
12 ______________________	__________

Stage 2

The second stage in using the program is to tell the computer which jobs you are interested in. This is done by typing in a SUPERCLASS job code - read on.

The SUPERCLASS System

The SUPERCLASS system gives a name and a code number to every job or job area.
Jobs are grouped into 14 Main Categories, and each category is then sub-divided into other categories.

For example, the first Main Category is

A Business & Management

Each Main Category is divided into 9 sub-categories :- for example

A Business & Management
- A1 Business & Management (General)
- A2 Small Businesses
- A3 Management Skills, Systems and Techniques
- etc.

Each of these is divided again, for example :-

A Business & Management
- A8 Office and Secretarial Skills
  - A81 Office Supervision
  - A82 Clerical Office Skills
  - A83 Secretarial Skills
  - etc

These categories are divided even further, for example

A821 Office Filing

As the code number gets longer, the job it describes becomes more and more precise.

You should give the most precise code you can.

The complete SUPERCLASS system list thousands of job names and codes. Only some of them have been fed into the computer at the moment. These are all listed on the pages that follow.
The heading on each page is "Code N Jobname".
"Code" is the code you have to write on the SUPERCLASS FORM which you

will be given.
"N" is the number of modules linked to that code. If the column is blank, there are no modules linked to that code (so there's not much point choosing that code).
"Jobname" is the name of the job that the code represents. This is the column you should be reading to find a particular job.

Before using the computer you have to fill in the **SUPERCLASS Form**, which you will be given. Finish reading these instructions before you fill in the form.

The SUPERCLASS Form

The SUPERCLASS Form is divided into columns headed Level 1, Level 2, etc.
A Level 1 code has one letter in it, for example, B.
A level 2 code has a letter and a number, e.g. B1
A level 3 code a letter and two numbers, e.g. B12
and so on.

Choose a level 1 code that interests you, and write this on the form.
Now follow through the SUPERCLASS list and write in a level 2 code **from the same category.**
Continue working through the list, filling in codes at deeper and deeper levels, as far as you can.

You can then start on a different Level 1 code, if you want to.

This is necessary for the following reason :-
The program is designed to find modules that are connected with the job codes. If you only typed a Level 1 code, this is a very general job name - there could be hundreds of modules connected with that job. Deciding which ones you want to take would be difficult. If you can narrow down the job name the program can pick only those modules that you really need.

You should fill in the SUPERCLASS Form now.

Once you have filled in the form you can start using the Selector program.

While you are waiting for the computer you should read the rest of this information, as this will help you understand what the computer is doing.

What the computer does

When you type in a job code, the first thing the computer does is check that the code is one of those listed.
If it is listed, the computer will then start looking for modules to suit that job.

No Modules available.
The SUPERCLASS system was designed separately from the SCOTVEC system of modules. There are some jobs listed in SUPERCLASS that don't have any modules connected with them yet, so you might get the message that there are no modules available.

If there are modules available, the computer will list them.

Entry Requirements
Some modules can be taken by anybody, but some can only be taken if you have the right qualifications (These are called Entry Requirements). If the computer finds a module to suit a job, it has to check that you have the qualifications to take that module. If you have, there is no problem. If you haven't, the computer will find the modules you need to take to get the qualifications. Your module course will be made up of all of these modules. You will see the computer listing the entry requirements. If a module number is given without a name, it means that the module is already in the list.

When the computer has checked everything, it lists all the modules in your course, so far.

You will now see the message

0 Check that modules fit
1 Add more modules
2 Delete some modules
Type the number of your choice.

0. Check that modules fit.
What this does will be described later. To decide whether it is worth doing this yet, follow these instructions.
Look at the top of the screen. You will see the Number of modules listed, and the Course length. There are 48 spaces in a course (that is the Target).

If the Course length is getting close to 48, you should use option 0, Check that modules fit. If the Course length is bigger than 48 you should definitely use this option.

1. Adding modules

If you want to add some more modules (and the target is a Course length of 48), choose this option.
The computer will ask for another job code. You can use any code, but if you want some leisure modules don't wait too long before you say so.
You will occasionally find that adding another job code does not add any new modules - the modules are already in the list.

2. Deleting modules

When you see the list of modules, there may be some you don't want to do. You can tell the computer to remove these, after choosing this option.

BUT

Suppose you want to take a module, 'X', and this has an entry qualification, module 'Y'. You don't fancy the sound of module 'Y', so you tell the computer to remove it. But you cannot take module 'X' without module 'Y' - you haven't got the qualifications. The computer will warn you about this and ask if you really want to remove module Y. If you say 'Yes', the computer will remove **both** of the modules, because it won't let you take module 'X' without module 'Y'.
This is unfortunate, but you will just have to accept that to be able to take some modules, you have to take others that you don't really want.

0. Check that modules fit

When you choose this option, you will see various messages. These are simply to tell you that the computer is still working, because it takes some time to do all the arithmetic.
You will see the following messages :-

Checking modules

The computer is sorting the modules into different types.

Branches

Some modules are entry requirements to a number of others. The computer has to work out the links between these modules.

Sequences

Where one module is an entry requirement to another, this is a sequence. Some modules have so many entry requirements that you could not possible complete them all in one year. It is a waste of time keeping these modules in the list, because they won't fit into a timetable. The computer checks for sequences that are too long.

If it finds any it deletes them, and you might see the message "Deleting modules". There is nothing you can do about this. If you really want to do these modules you will have to come back for a second year. By that time you will have passed the entry requirements.

Splitting modules

This is caused by the same problem as 'Sequence too long'.

To try to help, the computer looks for ways of 'squeezing' in the entry requirements. It does this by 'splitting' modules. This means that instead of taking a module for 3 hours a week for 13 weeks, you can take it 6 hours a week for 6 weeks - this gives you 7 weeks spare to fit in another module. Because of this, it is possible to complete some of the more difficult modules in one year instead of having to take two years.

Start and finish times

The computer has to work out where the modules can be fitted into the course plan.

It sometimes has to work this out more than once to take account of sequences.

It doesn't usually take very long.

The Course Plan.

Eventually you will see a grid showing sets of dashes, and the list of modules down the right hand side of the screen. As you watch, the modules disappear from the list and appear in the grid. The computer has worked out where to place the modules and slots them in.

If you are lucky the computer will be able to fit all of them in. If not, it will tell you which ones cannot be fitted.

You will then see the message

0 Print data then Continue or Finish
1 Continue altering the course

If there are lots of blanks in the Course Plan, choose option 1. This will allow you to add more modules (by giving more job codes) or delete modules (so that others will fit).
If you choose to Print data, the computer will print out the list of modules and the Course Plan.

It will eventually ask, "Do you want to print the other data", and tell you that this would not normally be given to a student. This extra data is used in testing the program - you will find it doesn't mean anything to you, so you should answer "N".
You then have the chance to continue altering the course, or finishing.

The SUPERCLASS Form						
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						/

****

SUPERCLASS categories and number of modules
File was last updated on Thu,10 Jun 1993.21:09:16
There are 110 non-zero entries.

****

A	Business & Management
B	Law, Politics & Economics
C	Arts, Crafts & Hobbies
D	Culture, Society & Education
E	Language, Communication & Self Help
F	Music & Performing Arts
G	Sports, Games & Recreation
H	Food, Catering, Leisure, Tourism
J	Environment, Security, Health & Safety
K	Agriculture, Horticulture & Animal Care
L	Science and Mathematics
M	Health & Personal Care
N	Architecture & Construction
P	Computers, Electrical & Electronic Engineering
R	Engineering Production & Industrial Design
S	Minerals, Materials & Fabrics
T	Transport Services & Vehicle Engineering

Code	N	Jobname
A		Business & Management
- A1		Business & Management (general)
- A2		Small Businesses
- A3		Management Skills, Systems & Techniques
A31		Management Skills
A312		Business Communication Skills
A3122	1	Report Writing
A318	1	Meetings and Committees
- A4		Human Resources Management
- A5		Financial Management & Accounting
- A6		Marketing, Sales & Distribution
- A7		Information & Library Management"
- A8		Office and Secretarial Skills
A81	1	Office Supervision
A82		Clerical Office Skills
A821	3	Office Filing
A822	10	Business Correspondence
A83		Secretarial Skills
A831	1	Secretarial/Management
A833	6	Medical Secretarial
A834	5	Legal Secretarial
A835	5	Personal Assistant Skills
A84		Shorthand
A841	12	Shorthand in English
A85		Typing/Keyboarding
A851	3	Audio Typing
A852	17	Copy Typing
A853		Shorthand Typing
A854	14	Keyboarding
A86		Reception & Telephone Switchboard Work
A861	2	Telephonists
A862	3	Recptionists (business)
A87	2	Word Processing Skills
A88		Office Technology Skills
A883	1	Office Computer Appreciation
A886	2	Electronic Mail
A887	1	Office Copying
- A9		Public Administration

	Code	N	Jobname
	B		Law, Politics & Economics
-	B1		Politics
-	B2		Economics
-	B3		Law
	C		Arts, Crafts & Hobbies
-	C1		Fine & Graphic Arts
-	C2		Design (non-industrial)
	C21	2	Design (general)
-	C3		Crafts (general), Collecting & Antiques
-	C4		Decorative Arts & Crafts
-	C5		Woodwork & Basketry Crafts
-	C6		Metal Crafts
-	C7		Glass, Ceramics & Stone Crafts
	C75	1	Ceramics (craft)
-	C8		Fabric Crafts
-	C9		Home Maintenance Crafts
	D		Culture, Society & Education
-	D1		Social & Cultural Studies
-	D2		Literature
	D21	2	Literature (general)
-	D3		History
-	D4		Archaeology
-	D5		Religious Studies
-	D6		Philosophy
-	D7		Social & Welfare Work
-	D8		Education & Training
	E		Language, Communication & Self Help
-	E1		Basic Skills
	E11	5	Basic English
	E12	3	Writing & Reading (literacy)
	E121		Reading (literacy)
	E122	3	Writing (literacy)
-	E2		Career Change
	E25	3	Preparation for Training & Education
-	E3		Self Help & Personal Development
-	E4		Communication for and with Disabled People
-	E5		Languages & Language Studies
-	E6		Communication & Mass Media
	E62	2	Writing (authorship)
	E623	2	Creative Writing
	E65	4	Public Speaking
	E66	1	Communication skills
-	E7		Audio & Visual Media
-	E8		Print & Publishing

Code	N	Jobname
F		Music & Performing Arts
– F1		Dance
– F2		Theatre & Dramatic Arts
– F3		Variety, Circus & Modelling
– F4		Theatre Production
– F5		Music, History & Theory
F55	2	Music Appreciation
– F6		Music of Specific Periods
– F7		Music of Specific Kinds
– F8		Music of Specific Cultures
– F9		Music Performance
F92	3	Singing
F922	1	Singing (solo) and Voice Studies
F93	3	Musical Instruments (Performance)
F931	1	Musical Instruments: Keyboard
F932	2	Musical Instruments: Stringed
F933	2	Musical Instruments: Wind
F934	1	Musical Instruments: Percussion

Code	N	Jobname
G		Sports, Games & Recreation
— G1	2	Sports Studies & Combined Sports
— G2		Air Sports
— G3		Water Sports
G32	1	Swimming
G321	1	Swimming (Learners)
G322	2	Swimming (Advanced)
G323	1	Water Polo
G324	2	Life saving (Swimming)
G36		Water Craft Sports
G361	2	Canoeing
G362	1	Sailing
G3625	1	Windsurfing
G364	2	Rowing
— G4	2	Athletics, Gymnastics & Combat Sports
G44	2	Gymnastics
G441	2	Trampolining
G45	2	Weightlifting
G46	2	Combat and Martial Arts
G462	2	Wrestling
G47	2	Fencing
— G5		Wheeled Sports
— G6		Winter Sports
G61	2	Ice Skating
G63	1	Skiing
G65	2	Curling
— G7		Ball & Related Games
G71		Football
G711	2	Association Football
G712	2	Rugby Football
G714	2	Gaelic Football
G72	2	Hockey and Related Games
G721	2	Lacrosse
G73		Hand Ball Games
G731	2	Basketball
G732	2	Handball
G733	2	Netball
G734	2	Volleyball
G74		Batting Games
G741	2	Cricket
G743	2	Baseball
G75		Bowling Games
G752	2	Bowls
G76	2	Golf
G78		Racquet Sports
G781	2	Tennis
G782	2	Badminton
G783	2	Squash
— G8		Country & Animal Sports
G81	1	Walking
G82	1	Climbing
G83	1	Orienteering
G88		Target Sports
G881	2	Archery
— G9		Indoor Games
G91		Table Games
· G911	2	Table Tennis

Code	N	Jobname
H		Food, Catering, Leisure, Tourism
- H1		Hotel & Commercial Catering
- H2		Cooking & Food & Drink Preparaion
- H3		Home Economics
- H4		Food Science
- H5		Food Manufacture & Distribution
- H6		Tourism, Leisure & Arts Management
J		Environment, Security, Health & Safety
- J1		Occupational Helath & Safety
- J2		Public Health & Safety
- J3		Cleansing
- J4		Security
- J5		Fire Prevention & Fire Fighting
- J6		Pollution Control
- J7		Environmental Conservation"
K		Agriculture, Horticulture & Animal Care
- K1		Agricultural & Horticultural Studies (general)
- K2		Agricultural Sciences
- K3		Crop Production
- K4		Gardening, Forestry & Amenity Horticulture
- K5		Forestry
= K6		Animal Husbandry
- K7		Agricultural Engineering & Farm Machinery
- K8		Agricultural & Horticultural Management & Maintenance
- K9		Veterinary Service & Pet Care

	Code	N	Jobname
	L		Science and Mathematics
-	L1		Science & Technology (general)
	L11	8	Science (General)
	L12	8	Science Introductory Studies
	L13	5	Science Laboratory Practice
	L15	5	Research Methods
-	L2		Physics
	L21	2	Physics (General)
	L22	2	Mechanics
	L223		Fluid Mechanics
	L2232	1	Hydraulics
	L23	2	Optics
	L24		Thermal Physics
	L242	1	Cryophysics
	L25	2	Electricity & Magnetism (Physics)
	L251	1	Electronics (Physics)
	L26		Physics of Matter
	L261	2	Nuclear Physics
	L262	2	Atomic Physics
	L263	1	Molecular Physics
	L264	3	Radiation Physics
-	L3		Chemistry
	L31	1	Theoretical Chemistry
	L35	1	Applied Chemistry
	L36	1	Chemistry (General)
-	L4		Mathematics
	L41		Mathematics (General)
	L411	2	Mathematics (General):Unspecified
-	L5		Astronomy
-	L6		Earth Sciences
-	L7		Surveying & Cartography
	L77	1	Navigation
-	L8		Life Sciences
	L83		Biology
	L831	1	Biology (General)
	M		Health & Personal Care
-	M1		Health (general) & Health Administration
	M11		Health (general)
	M12		Health Administration
-	M2		Medical Sciences
-	M3		Medical Technology & Pharmacology
-	M4		Dental, Opthalmic & Audiology Services
-	M5		Paramedical Therapies
-	M6		Nursing
-	M7	2	Psychology
-	M8		Personal Care Services
-	M9		Personal Health Care & Fitness

	Code	N	Jobname
	N		Architecture & Construction
-	N1		Architecture
-	N2		Planning
-	N3		Building Surveying, Estimating & Valuation
-	N4		Construction & Property Management
-	N5		Building & Construction Operations
-	N6	5	Wood & Woodworking
	N61	3	Woodworking
-	N7		Construction Site Practice
-	N8		Civil Engineering
-	N9		Strutural Engineering
	P		Computers, Electrical & Electronic Engineering
-	P1		Electrical Engineering
-	P2		Electronic Engineering
	P21		Electronic Engineering (General/basic)
	P213	1	Analogue Electronics
	P24	1	Optoelectronics
-	P3		Control Engineering
	P236	1	Computer Hardware
-	P3		Control Engineering
-	P4		Computer Systems & Software Engineering
-	P5	1	Information Technology / Computer Applications
	P51	4	Computer Appreciation
-	P6		Electrical & Electronic Servicing

Code	N	Jobname
R		Engineering Production & Industrial Design
- R1		Engineering / Technology / Manufacture (general)
- R2		Engineering Systems & Services
- R3		Production Managment / Quality & Reliability
- R4		Industrial Design
R43	2	Product design
- R5		Engineering & Plant Safety
- R6		Production Process Work
R69	1	Tools:Use & maintenace
- R7		Testing, Measurement & Precision Engineering
- R8		Mechanical Engineering
S		Minerals, Materials & Fabrics
- S1		Mining, Oil & Minerals Technology
- S2		Materials Engineering & Technology
- S3		Chemical Engineering & Technology
- S4		Textiles, Fashion & Furnishings
- S5		Leather Footwear & Fur
- S6		Furniture Manufacture
- S7		Paper & Board
T		Transport Services & Vehicle Engineering
- T1		Transport (general)
= T2		Aviation
- T3		Marine & Waterway Transport
- T4		Rail Transport
- T5		Driving & Road Safety
- T6		Freight Handling
- T7		Road Transport Operation
- T8		Vehicle Maintenance & Repair
- T9		Vehicle Manufacture & Sales

Appendix 10

Operating Instructions for the prototype programs.

Equipment required

BBC Master 128 micro-computer, with VDU (monochrome or colour).

Single or twin drive, double sided, 80 track disc drive.

Epson compatible printer (if printer options are to be used).

Starting up

Ensure that the equipment is connected and switched on.

The VDU display should show the message;

```
Acorn MOS
Acorn 1770 DFS
BASIC
>_        (flashing cursor)
```

Proceed as follows;

1) Insert the disc in disc drive 0 (labelled A on some disc drives).
2) If the drive is switchable, ensure that it is switched to 80 track.
3) Type *ADFS <RET>

   (The instruction <RET> means press the key marked RETURN.)

The drive should run for a few seconds. The symbol > should then reappear, followed by the flashing cursor.

On the rear left of the keyboard there are three red lights. The light for CAPS LOCK should be on. If it is not, press the CAPS LOCK key (in front of the light).

Running the prototype programs

Hold down the SHIFT key and tap the BREAK key.

The disc drive will run for a few seconds. A message will appear briefly, and then the Main Menu will appear.

Either of the two programs can be run by selecting from the options displayed.

Warning!

If a printer is not connected do not use the versions that print responses. The computer will attempt to print at various times. If it cannot, because there is no printer connected, the program will 'freeze'. (Press ESCAPE and re-start by holding down the SHIFT key and tapping the BREAK key.)

If the Selector/Sequencer program is being run, and the Course Grid is inspected on screen, do not attempt to print the data unless a printer is connected and switched on (for the reason given above).

Construct the Pre-entry List

Follow the on-screen instructions to the point where the student is asked to give his name. Any characters can be typed at this point.

When asked to give a student number, type a single digit. The students who tested the system were given longer numbers. Using a single digit will ensure that a new file is started, i.e. to simulate a student who has not used the system before.

To demonstrate the 'SCE' part of the program, respond 'Y' when the student is asked if he has any SCE subjects.

Follow the on-screen instructions to record SCE subject passes.

To demonstrate the 'modules' part of the program, respond 'Y' when the student is asked if he has passed any modules.

As no error-checking is performed, any numbers/letters can be given as module reference numbers.

When quitting from this part of the program, the message "The information would now be recorded on disc" will appear. To prevent alteration to the data on the disc it has been 'protected' and cannot have additional information recorded on it. In normal use the data would be filed for future use.

To simulate the condition where a student wishes to add to the information recorded (e.g. after he has completed some modules) re-start the program (using SHIFT-BREAK). This time give the student number '9999'. This accesses a file recorded by the author before the disc was protected. The previously recorded information is displayed and the student is invited to add to it.

The SELECTOR / SEQUENCER Program

On selecting this program the following message will be displayed;

```
OPENING FILES.  PLEASE WAIT
FILES STILL TO BE OPENED = 5
```

The number of files counts down until the screen clears and the message shown below appears.

```
Using the SUPERCLASS system, give the code for the job you are interested in.

                                                   Number of modules = 0
                                                   Codes already entered
                                                                 -------

JOB CODE _   <Flashing cursor
```

The job code used should be one of those from the JOBS-MODS file (see Appendix 7). Only rudimentary error-trapping has been provided, and results are not predictable if other codes are used.

The instructions for the use of the program are given on-screen at each stage. Where a response can only be a single digit, no <RET> is required. Hence, <RET> is only required after a job code has been typed, or a module has been identified for deletion, by specifying its position in the list - as there can be more than 9 modules in the list, this number may contain more than one digit.

The easiest way to see the typical operation of the program is to type the following list of codes (by choosing option 'A', Add more modules) :-

A822 (Business Correspondence)

A854 (Keyboarding)

A862 (Receptionists (Business))

This results in a list of modules which might be of interest to a student hoping to work in an office.

The following processes will be observed :-

A list of modules associated with the each code is displayed.

Entry requirements are then located - some of these are already in the course and are not repeated.

Some modules e.g. 'Text Processing 5' will be displayed initially, but will then be deleted. This module cannot be fitted into a one year course.

Some modules will be split to allow long sequences to be fitted into the course.

At this stage choose option 'C' instead of adding more modules. The Course Grid will appear and gradually fill up as modules are placed. The split modules have the suffix A or B.

Now choose option 1, Continue altering course.

As the student may now feel he has enough work related modules he might try adding some recreational job codes.

Use the following codes, but **Check the Course Plan** after adding each.

F92 (Singing) (all modules fit)

G72 (Hockey etc.)

The resulting modules displace "Producing Text from Audio", which can no longer fit in the course. This is because the Hockey modules form a sequence and are fitted in preference to the independent module "Producing Text ....". In an improved version the student would be able to mark the "Producing Text ..." module as essential, to force the program to fit it, probably at the expense of at least one of the hockey modules. In the prototype version this can be achieved by choosing to "Continue altering course" and then Deleting Hockey 2 (module 24 in the list). All the modules can now be fitted.

As only non-adjacent spaces are now left, only half-length modules can be fitted. The following list of codes, all related to sports or recreational activities can be used to fill the Course Grid.

G81
G82
G83
G63
G361 - then delete Canoeing 2, which displaces "Producing Text ....".

The result is a full programme, of which 36 half-modules are related to working in a office and the remaining 12 are sports and recreations.

Appendix 11
Memory requirements of the system

The system required by the college consists of the data files and the Selector/Sequencer/Evaluator programs. The college does not need access to any of the file editor programs except for that of the COLLEGE file. In this section the approximate sizes of the files and programs will be calculated, as this will be of value in deciding the hardware requirements. For simplicity a number of assumptions will be made in these calculations, as follows:-

a) all data is stored as alphanumeric characters.

b) one character requires one byte of space.

c) one kilobyte equals one thousand bytes. (The abbreviation 'k' will be used to mean kilobyte.)

In the calculations that follow, where approximations are made they will always err on the side of over- rather than under-estimating the space required.

SCE-CODES

At present there are about sixty subject names in the file, of which the longest name contains forty characters. Each name is associated with three code numbers which can be up to three characters long. Thus the longest record possible contains forty-nine characters (fifty will be used in the calculation). If the file is constructed so that all records have the same length, and the number of subject names rose to one hundred, the file would contain 100 x 50 characters = 5 000 characters = approximately 5 k.

COLLEGE

The college file will contain the reference numbers of the three or four modules considered essential for all students and a number of weighting factors to be applied in the Evaluator. If, as is recommended, the file does not record the reference numbers for the modules available at the college, the file length will be negligible.

SUPERCLASS / JOBS-MODS

If these two file are amalgamated as recommended, each record will consist of a job code, the job name and the list of modules associated with the job. It is difficult to estimate the size of this file. The simplest calculation would assume that each module is related to a different job code - the file would then contain three thousand records. However, a number of modules are related to the same job code, which reduces the size of the file, while many modules are related to a number of job codes, which increases the size. In the prototype, there are 110 job codes associated with 184 modules. Assuming that the full-size files are related in the same proportions there would be approximately 2 000 job codes (3 000 x 110 / 184 = 1793).

The longest job code contains 6 characters.
The longest job name in the prototype file is "Information Technology / Computer Applications", which is 46 characters long (assume 50).
The modules should be related to the most specific job code that can be identified. This reduces the number of modules associated with any job code. Assume that 5 modules will be related to each job code. Note that this assumes 5 modules related to every code, at every level used.
Each module reference will be 9 characters long, assuming that the module reference numbers contain seven digits (as given by SCOTVEC) and two characters for the SUPERCLASS category (as is recommended). Each record will contain 6 + 50 + (5 x 9) = 101 characters. Round this down to 100. (This can be justified in view of the large over-estimate caused by rounding 1 793 to 2 000 in estimating the number of job codes.)
The file size will be 100 x 2000 =200 k.
Note that this assumes that either the information is going to be displayed as a series of menus, or that, knowing the job codes will allow a map to be displayed (see Figure 9.3.8.2.1).

QUALIFICATIONS

The number of modules listed for each equivalence depends on the number of compulsory modules and the number of elective modules offered in each list. In addition, some equivalences do not list all the entry requirements; these need not appear in the QUALIFICATIONS file, as the Selector program will identify them from the MODULES file. It is thus difficult to estimate the size of the file with any certainty, but an attempt was made to do this, as will now be described.

The 'Guide to Modular Programmes' (SCOTVEC 1986) contains 156 pages of equivalences. Using the random number generator in the BBC Master, five random numbers between 1 and 156 were generated. These were treated as page numbers. The programme listed on or partly on each of these pages was then examined. The number of modules listed for each programme was counted. The results are shown in Table 1

Table 1

Number of modules listed for five randomly selected programmes
(Data taken from SCOTVEC 1986)

Page	Compulsory modules	Electives
9	28	5 16
49	5	
57	17	
60	16	
136	3	2 4 9 3 7 5 5 3 4 6 5 5 5 7 6

It should be noted that some programmes do not list the entry requirements to all the modules listed; hence the programmes will frequently be longer than appears from the figures above.

The space required to store the corresponding data can be calculated as follows (3);

a) For every module listed, 1 integer to store the reference number.
b) For every elective list, 2 integers; one to store the number of modules in the list and one to store the number to be selected.
c) For every equivalence, 2 integers; one to store the number of compulsory modules and one to store the number of elective lists.
d) Every integer occupies 5 bytes.

e) For every equivalence, to store the name of the qualification, 2 bytes plus 1 for every character.

Applying these rules to the examples given in Table 1, above, yields the memory requirements shown in Table 2, below. The columns (a) to (e) in that table correspond to the components listed above.

Table 2

Memory required to store data for five randomly selected programmes.

(Data taken from SCOTVEC 1986)

Page	(a)	(b)	(c)	Number of integers	(d)	Number of characters	(e)	Total
9	49	4	2	55	275	46	48	323
49	5	0	2	7	35	43	45	80
57	17	0	2	19	95	35	37	132
60	16	0	2	18	90	30	32	122
136	79	30	2	111	555	68	70	625

See text for meanings of column labels.

The average memory requirement was 256.4 bytes; consider this rounded to 256.

If there were five hundred 'average' equivalences, the memory requirement would be 128 k.

SCE-EQUIV

For each of the 100 (assumed) SCE subject names, each of assumed length 40, there will be a set of modules which, together, are considered to teach the same skills and/or knowledge. The number of modules in the set varies - inspection of Appendix 6 shows that the maximum number for an Ordinary grade was 5 (Computing) and for Higher grade was 6 (Secretarial). Assume 5 modules are quoted for each of the subject names. Each reference number contains 9 characters.

The total file length will be = 100 x 40 x 5 x 9 = 180 k.

ADVANCED

Making the same assumptions as for the QUALIFICATIONS file, this file will be 128 k.

MODULES

Assume 3 000 modules. The approximate length of a record can be calculated as shown below.

Field	length	
Category	2	
Reference number	7	(the category is already quoted)
Title	50	(one of the two longest names in the prototype was "Two-Dimensional Design - Experimental Printmaking"
Length	3	(e.g. '1.5')
Number of entry requirements	1	(e.g. '5')
List of entry requirements	45	(5 entry requirements, each of 9 characters)
Number of job codes	1	(assume '5')
List of job codes	30	(5 codes, each of 6 characters)
Number of exit modules	1	(e.g. '5')
List of exit modules	45	(5, each of 9 characters)
Taught in college	1	(this is the flag to avoid having to re-type reference numbers into the COLLEGE file. It is either 'Y' (yes) or 'N', the default)
Total	186	
Round up to	200	

For 3 000 records, the file length will be 3 000 x 200 = 600 k.

Module Summaries

A typical module summary is given in Appendix 2. Some of the information in that can be obtained from the module record. The information which is not available elsewhere occupies less than a quarter of a page. Assuming a line contains sixty letters, and the new information occupies 11 lines, the Summary occupies 660 bytes. Rounding that to 1 k, and doubling it on the assumption that Summaries will be made more informative, gives 2 k per

Summary.

Assuming 3 000 modules gives a file length of 3 000 x 2 k = 6 Mbytes.

The programs

Estimating a length for the programs is even more of a guess than the calculations given above. However, this work was typed using Microsoft Works 2e, which is an integrated word processor, drawing, database, spreadsheet and communications package. It is considerably more complex than the proposed system - the only part of the system that would resemble such a package would be the SUPERCLASS map (see Figure 9.3.8.2.1). The Microsoft Works package occupies 920 k (including the dictionary and some accessories). Assuming 1 Mbyte for the proposed system is almost certainly a considerable overestimate.

The Total

The calculations given above assume 3 000 modules. The Catalogue, however, continues to grow. The two sets of figures given below are (a) the size of the file for 3 000 modules, and (b) the size of the file per module (where the file size depends on the number of modules). This will allow an estimate to be made of the rate of growth of the system.

	(a) kbytes	(b) bytes
SCE-CODES	5	
COLLEGE	0	
SUPERCLASS / JOBS-MODS	200	100
QUALIFICATIONS	128	
SCE-EQUIV	180	
ADVANCED	128	
MODULES	600	200
Summaries	6 000	2 000
Program	1 000	
Total	8 241	2 300

The estimated size of the system is therefore approximately 8 Mbytes.

The high density discs used in the Macintosh Classic on which this work was typed have a capacity of approximately 1.4 Mbyte. The system will probably fit on 6 discs.

The last information on changes to the Catalogue (see Table 3.4.1) shows an increase over the previous year of 256 modules. The information on these modules would occupy 256 x 2 300 = 588.8 k. A single high density disc could contain information on up to 608 modules.

(1 400 000 / 2 300 = 608) This leaves plenty of room for information on alterations (rather than additions) which, unless the Module Summary is changed, occupy very little space.

References

BELL, D., MORREY, I., PUGH, J. (1992). *Software Engineering. A Programming Approach.* Hemel Hempstead : Prentice Hall International (UK) Ltd.

BLACK, H. (1988). *The National Certificate* . in Professional Issues in Education, Vol 2, Assessment : A Changing Practice. Edinburgh : Scottish Academic Press

BOEHM, B.W. (1979). *Software Engineering: R & D trends and defense needs.* in Research Directions in Software Technology, Wegner, P. (ed.), Cambridge, Mass.: MIT Press

BOOTH, P. (1989). *An Introduction to Human-Computer Interaction.* London : Lawrence Erlbaum Ltd.

BROWN, S. (1989). (ed.) *Assessment : A Changing Practice.* Professional Issues in Education, Vol. 2. Edinburgh : Scottish Academic Press.

CARROLL, J.M. (Ed.) (1991). *Designing Interaction. Psychology at the Human-Computer Interface.* Cambridge : Cambridge University Press

CAST (1987) *16+ Action Plan in Scotland. Staff Development Project* Volume 2, Section 4, Programme Design. Glasgow : Curriculum Advice and Support Team

COWAN, R. (1981) *International Yearbook of Education Vol.XXXIII - 1981.* UNESCO

DoE (1989) *SUPERCLASS Subject structure for Learning Opportunities. Training Access Points. Local Database Version.* Department of Employment

DUNNING, J. (1977) *Assessment for All. Report of the Committee to Review Assessment in the Third and Fourth Years of Secondary* Education. Edinburgh : HMSO

FEU (1981) *Vocational Preparation.* Further Education Curriculum Review and Development Unit.

GATHERER, W.A. (1989) *Developing the Secondary Curriculum* . in Professional Issues in Education, Vol. 5, Curriculum Development in Scotland. Edinburgh : Scottish Academic Press.

HANSON, O. (1982) *Design of Computer Data Files.* London : Pitman Publishing Limited.

HART, J. and PIRIE, S. (1985) *Choice in a Modular System.* Edinburgh : The Scottish Council for Research in Education

HART, J., ROGER, A., MULVIE, A., MUNN, P. (1987) *Managing Modules. Guidance and Choice in Further Education Colleges.* Edinburgh : The Scottish Council for Research in Education

IBM (1992). *Object-Oriented Interface design. IBM Common User Access Guidelines.* Que Corporation.

JOHNANTHAN, R. (1987). *The Case For and Against Modularisation.* in Scottish Educational Review, Volume 19 Number 2, November 1987. Edinburgh : Scottish Academic Press Ltd.

McDONALD, J.E., STONE, J.D., LIEBELT, L.S. (1983). *Searching for Items in Menus : The Effects of Organization and Type of Target.* in Proc. Human Factors Society, 27th Annual Meeting.

MARCOTY, M. (1991). *Software Implemenatation.* Hemel Hempstead : Prentice Hall International (UK) Ltd.

MUNN, J. (1977) *The Structure of the Curriculum in the Third and Fourth Years of the Scottish Secondary School.* Edinburgh : HMSO

MUNN, P. and MACDONALD, C. (1988) *Adult participation in Education and Training.* Edinburgh : The Scottish Council for Research in Education

NAPS, T.L. and NANCE, D.W. (1992). *Introduction to Computer Science : Programming, Problem Solving, and Data Structures.* New York : West Publishing Company

SCOTVEC (1985) *The National Certificate. Catalogue of modules available.* Glasgow : SCOTVEC

SCOTVEC (1986) *The National Certificate. Guide to Modular Programmes.* Glasgow : SCOTVEC

SCOTVEC (1988) *The National Certificate. Catalogue of Module Descriptors.* Session 1988-89. Glasgow : SCOTVEC

SCOTVEC (1992) *The National Certificate. Catalogue of Module Descriptors.* Session 1992-93. Glasgow : SCOTVEC

SEB (1991) *Scottish Certificate of Education. Conditions and Arrangements 1991.* Scottish Examination Board.

SED (1979) *16-18s in Scotland. The First Two Years of Post-Compulsory Education.* Scottish Education Department Edinburgh : H.M.S.O.

SED (1983a) *16-18s in Scotland. An Action Plan.* Scottish Education Department Edinburgh : H.M.S.O.

SED (1983b) *New Courses for 16-18s in Scotland. The Government Action Plan.* Scottish Education Department

SED (1985a) *Directory of Day Courses for 1985-86.* Scottish Education Department.

SED (1985b) *The Negotiation Process.* Scottish Education Department.

SED (1985c) *Selecting a Programme of Modules.* Scottish Education Department.

SED (1988) *The National Certificate 1986-87.* Statistical Bulletin No. 8/F5/1988. Edinburgh : Government Statistical Service

SED (1989) *The National Certificate 1987-88.* Statistical Bulletin No. 7/F5/1989. Edinburgh : Government Statistical Service

SED (1990) *The National Certificate 1988-89.* Statistical Bulletin No. 14/F5/1990. Edinburgh : Government Statistical Service

SED (1991) *Six Years On. Teaching, Learning and Assessment in National Certificate Programmes in Scottish Further Education Colleges. A Report by H.M.Inspectors of Schools.* Edinburgh. H.M.S.O.

SHNEIDERMAN, B. (1987) *Designing the User Interface : Strategies for Effective Human-Computer Interaction.* Addison-Wesley Publishing Company

SOMMERVILLE, I. (1989) *Software Engineering.* Addison-Wesley Publishing Company.

TOWNSEND, B.J. (1985) *File Handling on the BBC Microcomputer.* Macmillan Publishers Ltd.

WARWICK, D. (1987) *The Modular Curriculum.* Basil Blackwell Ltd.

UNESCO (1990) *Statistical Yearbook, 1990.* UNESCO

UNESCO (1991) *Statistical Yearbook, 1991.* UNESCO

GLASGOW UNIVERSITY L