ORGANIZATIONAL INFORMATION SYSTEMS

R. L. Ashenhurst

Rapporteurs: Dr. P.E. Lauer
Mr. C.R. Snow
Mr. R. Evans

First Lecture

In his first lecture, Professor Ashenhurst proceeded to outline the history, aims and publications of the ACM Curriculum Committee on Computer Education for Management, chaired by Daniel Teichroew. He suggested that the Committee had been formed partly to fill a gap some thought to have been left by a previous major curriculum effort, that of the ACM Curriculum Committee on Computer Science [Ref. 1]. A widely heard criticism was that eighty percent of the computing world was involved in the area of computer applications to which the computer science curriculum did not seem to address itself. This idea is sometimes referred to as commercial applications or business data processing.

The Committee was informally organized by D. Teichroew and was subsequently awarded a grant by the National Science Foundation. The major thrust of its work was the development of a graduate curriculum in information systems [Ref. 2].

The speaker first listed some preliminary efforts of the Committee which resulted in two reports [Refs. 3,4].

The first of these [Ref. 3] reported on a series of meetings the Committee held to ascertain the present state of matters in the business schools in the United States of America. This report showed that computer-related curricula were not as developed as everybody would have liked to have thought.

The second report [Ref. 4] is the Committee's so-called position paper, documenting the need for education for information systems professionals.

Since the major report [Ref. 2] is aimed at a narrower objective than some people may wish, that is to say at a graduate program, the speaker pointed out that the Committee is also well aware of the needs of continuing education and undergraduate education. Professor Couger

(also a speaker at this conference) is to comment on further Committee efforts in the area of undergraduate programs.

In the preparation of the report the Committee went through a considerable cycle of evaluation and modification. In particular there was a round of academic reviews, that is, reviews by people mostly, but not exclusively, in universities. A preliminary draft was circulated and referees were asked to comment in writing. Furthermore, a two-day workshop of people from industry was held. At this meeting the participants became familiar with the contents of the report and were asked to comment. As a result of these two types of reviewing activities, the Committee felt that the report had received an appropriate amount of scrutiny.

But the speaker made clear that the Committee does not intend to be overly didactic and maintain that this is the way it has to be done. The report embodies a set of recommendations intended to be firm enough to act as a framework for further discussion. It is hoped that it will be agreed that to have such a framework it is almost mandatory to have specific courses designed, to have extensive bibliographies suggested, and so forth. But there was no suggestion that this curriculum be enshrined "as in stone" but rather it was intended to function as a first, moderately definitive round in a process of continuing modification.

In that report the Committee tried to deal with several curricular aspects as efficiently as possible.

Basically, a set of courses was defined which, in effect, themselves constitute an independent "stand-alone" program, i.e. a masters level degree program in information systems development. This is designed as a professional program, two years in extent, starting almost, but not quite, from ground zero.

Next, the report proceeds to explain how one could then abbreviate that program, assuming enough undergraduate preparation, to make it into a one year independent program.

The speaker pointed out that by a stand-alone program he meant one that is offered as an end in itself. The question of just who offers it, be it a university, a business school, a computer science department or any group of well-meaning academics, is a matter of university politics, that is, it is a question of who does what in a particular university.

Beyond this, the Committee extracted from the material involved in the above mentioned programs a program applicable in a business school. The speaker suggested that an American business school could incorporate certain material in this curriculum as a specialty option in an MBA (Master of Business Administration) program. He also pointed out that an MBA program in the United States has very specific general requirements for breadth of general knowledge, hence the whole information system curriculum obviously could not be fitted into it. Thus the problem was to suggest an MBA program with some of this specific orientation.

The Committee also prepared a similar extraction for a computer science program, taking as its model the only thing that approaches a common kind of model, namely 68 [Ref. 1]. Again it was pointed out that the Committee is aware that master's programs in computer science are many and varied, so no attempt was made to be exhaustive. The Committee merely suggested that if one takes the philosophy of Curriculum 68 as specifying a computer science masters degree then the present report presents one way of fitting this specialty into it.

Lastly, a few suggestions were added to the report about industrial engineering and operations research departments, which have this same sort of general problem.

To aid the listener in understanding the general framework underlying the curriculum, the speaker proceeded to make the following basic conceptual distinctions.

The curriculum is in information systems development. Hence, the curriculum as a whole is intended for those who develop information systems, that is, for the professionals in systems development groups, whatever those groups may be called. The term information systems is the basic term denoting what is the topic of discussion. For instance, the topic is not business systems. Sometimes the term organizational information systems will be used indicating that information systems are practically always embedded in organizations of one sort or another. The organizations may be government, non-profit institutions, or whatever other ways there may be of classifying them. In short, an organizational information system is not just a collection of computer programs but a system which runs on a computer system, a complex of hardware and basic operating software, and runs in an organization in which it performs or serves some specific organizational purpose.

A fundamental problem faced by many organizations is how to, even with the best will and all the finances in the world, assemble a group which will effectively develop information systems for that organization. A secondary problem is how the organization incorporates line and staff people who are not in the information systems development group but who will work with (or be interfaced with) the information system. A final problem concerns how the concomitant information processing centres should be staffed.

Referring to the outcome of an earlier informal discussion of participants in the present conference, the speaker went on to say that it seems that there is not enough known about the nature of information systems or how they really should be managed to make the problem of their management simply a question of enough money and qualified staff. Hence, in some sense, the Committee's curriculum had to be based on certain ideas about certain approaches to the notion of what information systems are and what their management and development should be. The result should be taken as a considered opinion of the Committee, which spent many hours debating one or the other facets of the problem and which had the wide input mentioned above from both the academic and industrial circles to get feedback on its initial ideas.

Within the framework of the discussion of these problems Professor Ashenhurst indicated why he assigned the topics to his lectures, the topics of the first to the third lecture respectively being

1) What we can say about information systems.

2) What kind of positions, in the opinion of the committee, are being defined in industry and what kind of education it thinks is appropriate to fill these positions.

3) The details of the curriculum itself.

In the remainder of the lecture Professor Ashenhurst outlined some of the aspects of information systems which he considers basic for purposes of teaching. He suggested that if one is a teaching organization one has to proceed more systematically than, say, the line manager who can have his view of information systems communicated by a certain number of anecdotes about a certain amount of irascible dealings with people, and so forth. In teaching one has to formulate one's view of information systems somewhat definitively, i.e. one must have a point of view.

The speaker went on to express agreement with Professor Page's statement, during his introductory speech, that computer science departments have eschewed this area because it was assumed that there is not enough of a formulated way of looking at information systems or the way information systems work in organizations to really make it any part of curricula as they then existed.

In the report an attempt was made to coin various words and to keep using various concepts which are thought to be important. The report does not begin with a set of rigorous definitions, which was felt to be an activity which often turns out to be fairly sterile, but tries to communicate the meanings of those words by the way they are used, by putting them in italics the first time they occur and being fairly careful with usage of language from there on. The speaker commented that one nice thing about this way of beginning a report is that one can write fairly carefully and it does not necessarily appear that it is overly particular in style. The disadvantage is, however, that one reads the report without realizing that some fairly firm conceptual foundations were being assumed. Hence, one can read it superficially and not get as much of the impact of what was being said, as one would like. In writing the report there were some fairly firm definitions in mind and there were some fairly firm distinctions made having to do with the nature of information systems.

The first of the conceptual notions about information systems is that they are regarded as something that runs on, but not exclusively, a computer system. In fact there is a computer system part and a manual system part, which themselves are just the modus operandi for the information system.

Secondly, an information system runs in an organization. We can in effect think of an information system as something which is surrounded by the organization which it is serving, and consists of a hardware-software complex of some sort and people who are performing various tasks in connection with it.

Thirdly, the term information system development denotes a process which involves

an analysis segment
a design segment
an implementation segment, and
an operation segment.

The curriculum is aimed at professional practitioners who will take care of the analysis and design phases.

Fourthly, it is widely said that the user has been sadly neglected in system design and this was taken very much into account in the curriculum. But the speaker felt it desirable to differentiate between a so called high-level and low-level user. Some people differentiate between the beneficiary and the user and though this may be an awkward terminology, the speaker suggested that this emphasizes the distinction between those who use the information system in the sense that they know nothing about computers but who have to interact with the information system on a daily basis, and those in the organization who are supposed to benefit by the system, as for instance the management would be expected to benefit from a management information system. The need for making the distinction was illustrated by the fact that an information system may be "a beauty" to use but give useless information and thereby not be advantageous to the people who are supposed to benefit. Alternatively, a system can have some wonderful information in it for management and yet that information may be unobtainable because the people who are supposed to deal with the system on a day-to-day basis cannot get at it because, for instance, the operating procedures are too clumsy.

To summarize, when one refers to the user loosely, one should really think of both these facets. By both facets are meant the people outside the system who are supposed to interact with it in one way or another.

Fifthly, there are also the people inside the system. In particular, the speaker distinguished between the people who must operate the system and those who maintain it, and he suggested that these are two somewhat different kinds of activities. To have the system designed so that both operation and maintenance are effective and efficient is clearly a very desirable feature and it is sometimes neglected.

Finally, the speaker distinguished between maintaining the system, which is the sort of direct interaction which makes changes to the system, and modifying the system, which connotes the more high-level system group changes. That is, it should be a requirement of system design that the systems group should be able to modify the system in a coherent and orderly way at the request of the people for whose benefit the system is working. The high-level systems group should be able to specify

these modifications and then the maintenance group should be able to incorporate these modifications without interfering unduly in system operation.

One way of looking at systems, the way represented in the report and the present lecture, is by considering the aspects the system presents to the various people who must interact with it in various ways. Presumably, the system should be designed so that it works smoothly and effectively for all these people. The speaker pointed out that it is difficult to just sit down and say how one designs a system, how one formulates the concept of system so that it will do all these things for all these people. But he also expressed the view that one must have all those facets in mind when one talks about systems development because if one does not, one clearly will obtain something that leaves something to be desired.

Professor Ashenhurst's way of representing systems by focussing on the various groups that one wants to say are within the organization as well as on the aspects the system should present to them is illustrated in the diagram.

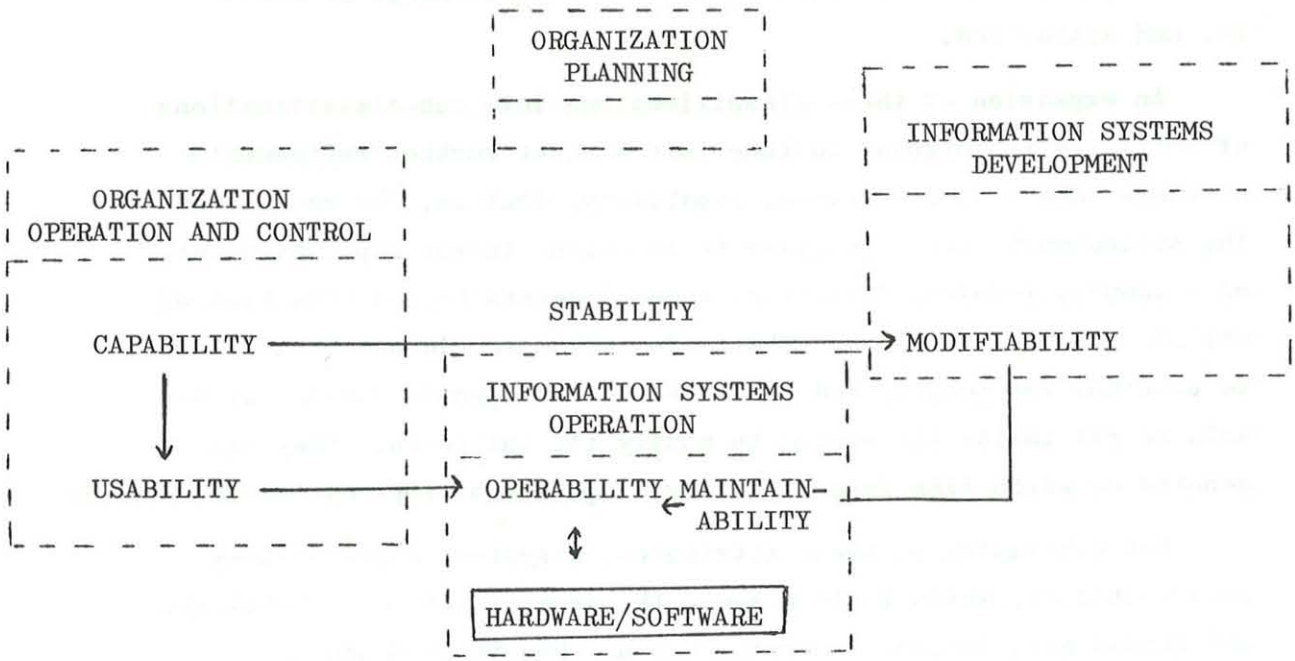


Figure 1

Information systems attributes and organizational functions

The diagram indicates that there are several different groups to be taken into account, the operating group at the bottom, the rest of the organization on the left side, the development group at the right and up above the planning group, who presumably is going to benefit from the information system, but does not have any direct interaction.

The arrows are labelled by different words which all end in the suffix "ability". Hence, it is convenient to call the attributes denoted by these words "system abilities".

Capability is the utility of the system to the high-level user, the person whom the organizational functions of the system are supposed to support, whereas usability is utility for those who directly interact with the system.

Modifiability and maintainability are the two aspects of changing the system once it is developed.

Operability is the facility of the system to be operated on a continuing basis. Stability is another desirable attribute of the system. The system should be stable, that is, predictable and constant in its behaviour, but it should also be modifiable, that is, changeable in its behaviour, if indeed such change is asked for and authorized.

An expansion of these classifications into sub-classifications of abilities is intended to take into account further refinements of these ideas. For instance, capability, that is, the requirement the system must fulfil in order to be useful to the top management of a company involves further side requirements beyond this kind of utility. Some of these are that information should not be available to unauthorized people, and that unauthorized people should not be able to get inside the system to modify it, and so on. They are denoted by words like inscrutability, impenetrability and invulnerability.

But contrasted to these attributes, a system should display predictability, which is in a sense the opposite of inscrutability, and should also display accessibility too when it is found that something has gone awry that human interpretation can remedy. Alterability is an attribute of a system which all too often is not taken into account. The system is designed to work in the stereotype case rather than the anomalous one.

There are also requirements for the internal management, such as being able to verify what is going on, being able to validate the operation and for external agents to be able to audit the system. Again, verifiability and auditability are not taken into account at the development stage nearly as often as one would think or hope.

Similarly, one can subclassify on the usability, operability and maintainability fronts. In particular, the operations group should be able to control the system both in its normal operation and to recover in the abnormal case. This way one can increase the list and one may like or dislike this way of using terms for all these things, but coining words for these notions gives one in effect a checklist of questions one can ask. In fact, equipped with all these words, one can ask some very good questions of a group that is developing a system.

Professor Ashenhurst went on to suggest that the system abilities approach to the characteristics of systems yields not only a tool with which the project management can ask embarrassing questions but one which can serve research in the university context. He felt that these words could be used as the basis; these words could be translated into technical requirements, such as the requirement of being secure to unauthorized access but open to authorized access, that is, the whole matter of systems security and procedures. In effect, the technical requirements here stem from certain operational requirements. One can set down these words, these characteristics, and then say what technical requirements they imply and in particular what techniques of development, what techniques of structuring stem therefrom. In this way progress can be made in a general way regarding the factors that govern systems development.

It may have been noticed that no mention has been made of the particular organizational context in which systems work. It is, of course, necessary that you know that an airline system does one thing, an oil refinery system another and so forth. But here an attempt is made to get general characteristics and set them in a framework which is specific enough so that most questions can be asked and further research can be done.

Professor Ashenhurst next expressed the fact that he thought that the further development of information systems could also proceed on lines suggested by Herbert Simon in his little book The Sciences of the Artificial [Ref. 5]. This thought-provoking book applies to systems in a very general sense, and to information systems in particular. Simon talks about a system as being something with an inner environment and an outer environment. Since he is talking about systems in general, he uses exemplary contexts like a watch, which has the works inside it and has the environment outside of it, namely, those who are looking up the time. Professor Ashenhurst thought this a useful way to look at systems.

For the particular case of "information systems", one conceives of the computer system on which the information system runs as an inner environment, and the organization system outside it as the outer environment.

The speaker also stated that he would like to see this type of analysis carried further rather along the lines of the same kind of conceptual structure that is common in the analysis of operating systems today and mentioned Professor Dijkstra as one of the chief orchestrators of these concepts. In particular, he suggested that the outer level is the beneficiary or the group for whom the system works, and this level has as the interface with the inner level the information system. That is to say, the information system must look a certain way to the people outside and that way is presumably somewhat independent of what the computer system looks like inside, or what the programs and files look like. The information system is supposed to look like an aggregate of information inputs and outputs to the outermost level.

Another level the speaker referred to was that of the user, the direct interactor, who must know something about the programs and files, but does not want to know about the actual computer structures in which these are embedded.

A further level is that of the people who operate the system. They must not only know something about the programs and files, in fact it is not so important to know what programs are doing as to know what unit is doing which program, so to speak. They need to know what the hardware and the basic operating software looks like.

This way of approaching systems is illustrated by a diagram which was accredited by the speaker to D. Teichroew (see Figure 2).

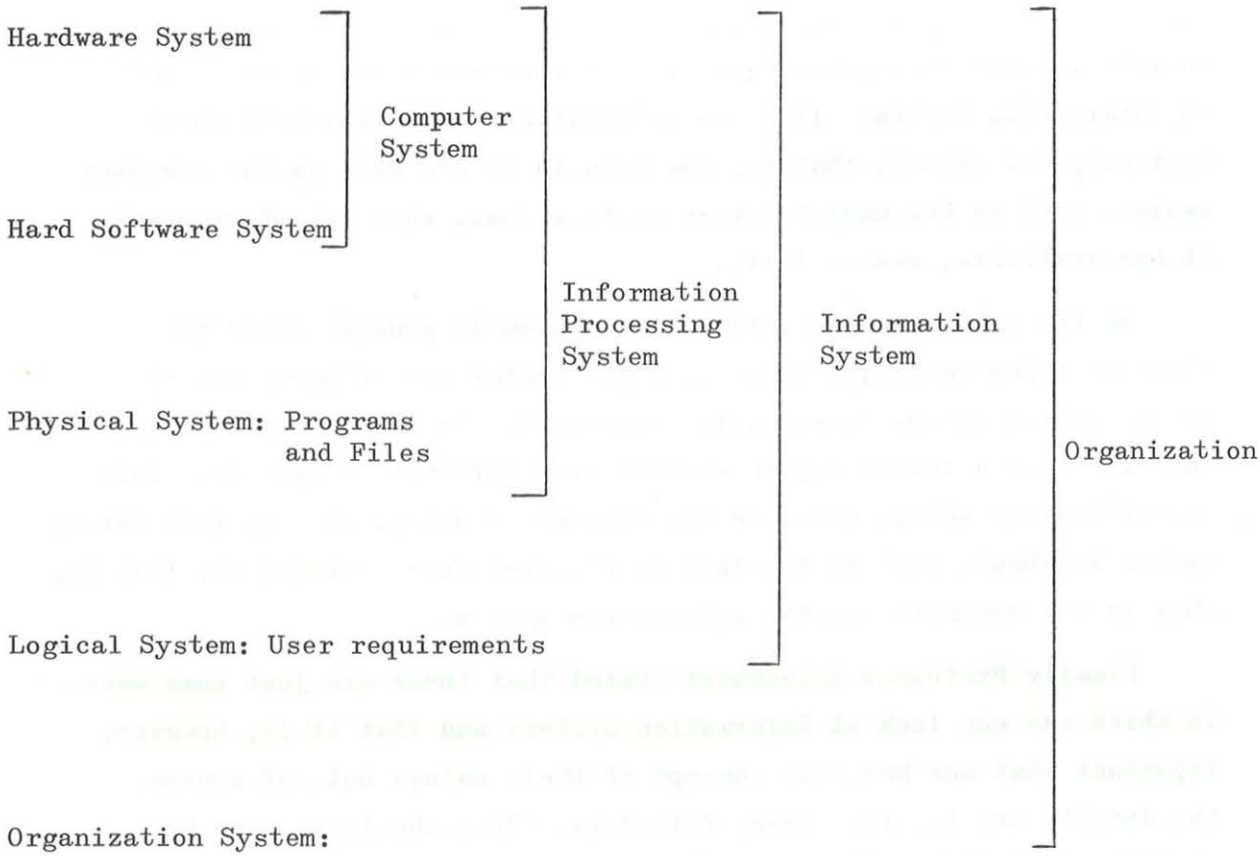


Figure 2

Levels of Systems to be designed, analyzed, constructed and measured.

A third approach which the speaker felt capable of bearing fruit again comes from operating systems. In operating systems studies one talks about processes which are programs in execution, something specified by the program while it is going on, and this concept should be readily amenable to generalization. However, he pointed out the danger that computer scientists might let information systems collapse into operating systems by reducing information systems to our knowledge about operating systems. The distinction that needs to be made is that the operating system of a computer system is in fact an information system. It is an information system which is about that computer system, that is, the data in it are data on the computer system, such as how many hardware units it has, what set of programs it has available, and so forth.

On the other hand an information system in general could be about an organization, a library, a particular set of equipment of an industrial system or something like that. The author added that this might be a fourth way to characterize information systems. This classification scheme would be the outcome of asking what an information system is about, what is the data in it, what does it refer to, that is, what is the semantics of the information system.

Finally Professor Ashenhurst stated that these are just some ways in which one can look at information systems and that it is, however, important that one has some concept of their nature but, of course, the details are far from being filled in. This should be done by computer scientists and others who are interested in promoting this field or perhaps they will fill in other conceptual frameworks that may seem pertinent to them. The result should then be made the basis both of research and the structuring of courses and presentations. In summary of the substance of his first lecture he stated that there must be a greater conceptual foundation in this problem area. To supply this is by no means an impossible task and it is one worthy of intellectual effort. One of the effects that it is hoped that the curriculum report will have is to spur people on in directions as indicated in that report.

Discussion.

Professor Dijkstra asked what the arrows meant in the diagram? (See Figure 1).

Professor Ashenhurst said that he thought of the arrows as meaning the direction of the flow of requirements. The information system is there to give a capability and that is where all the arrows stem from. What capability is required is what determines the modifications and hence an arrow goes from "capability" to "modifiability". Furthermore, what the capability is determines how the user proceeds and how the people who directly interact with the system shall use the system and shall be able to interact with it. The arrows indicate a sort of conceptual requirement flow but indeed, that was not spelled out in text.

Professor Verrijn-Stuart felt somewhat uncomfortable that Professor Ashenhurst had not got some correspondence to the notion of flexibility in his scheme. By flexibility he meant the capability a system has when different things are built in right from the start, thinking of the future for which one has already reorganized the whole setup.

Professor Ashenhurst replied that one had a very particular problem, which was all too familiar in the science of programming, namely did one build in all of the possible adaptations into the original version, or did one design a system so that it could be modified in its behaviour suitable to changing circumstances, flexibility versus adaptibility. One had a trade-off and he thought of this as roughly on the stability-modifiability axis.

He said that "either we designed the system to be highly modifiable to keep up with changing circumstances or we made the system very general, in which case it was supposed to be able to modify itself as the circumstances change. I think in the design of big information systems we have a similar problem as in the design of programs, only a more complex and horrendous one. In the latter case, one has to talk about changing operating procedures and changes to what the people interacting with the system should see and so forth. Yet, it is probably true that the range of things we have to take into account, the range of circumstances we must anticipate, is so great that it is very difficult to design the system in sufficient generality. So you must depend on a continuing process of incorporating modifications".

Professor Verrijn-Stuart stated that the key word was "trade-off".

Second Lecture

In his second lecture, Professor Ashenhurst outlined the specific course-structuring in the curriculum. The educational requirements were mainly that there should be different course streams to fit people for different kinds of tasks. The basic course is designed to be a professional graduate program to train people to work in information systems development. It can be modified to fit into an MBA degree, or to fit into a computer science course. Referring again to Figure 1, the four dotted boxes may perhaps indicate four departments within an organization, or in any case four separable kinds of activity. A student emerging from the basic program would be suited for entry into the information systems development section, but having one eye on the organizational operation and control and one eye on information systems operation.

From the MBA with information systems option the student might be expected to move into the organization operation group interfacing with the information systems operation group, with particular concern for a specific application area (accounting, production, etc.). Being more familiar with management studies, he might also expect ultimate advancement to the organization planning group.

The graduate in computer science with this option would probably tend towards the information system operation group, since he will most likely be less concerned with the overall structure of the system.

The courses themselves fall into four categories, labelled A,B,C, and D.

A - Analysis of Organizational Systems
B - Background for Systems Development
C - Computer and Information Technology
D - Development of Information Systems

Figure 3 gives the titles of the courses in each of these categories.

Analysis of Organizational Systems

- A1. Introduction to Systems Concepts
- A2. Management Functions
- A3. Information Systems for Management
- A4. Social Implications of Information Systems

Basic Tools for Systems Development

- B1. Operations Analysis and Modelling
- B2. Human and Organizational Behaviour

Computer and Information Technology

- C1. Information Structures
- C2. Computer Systems
- C3. File and Communication Systems
- C4. Software Engineering

Development of Information Systems

- D1. Information System Analysis
- D2. Information System Design
- D3. Systems Development Projects

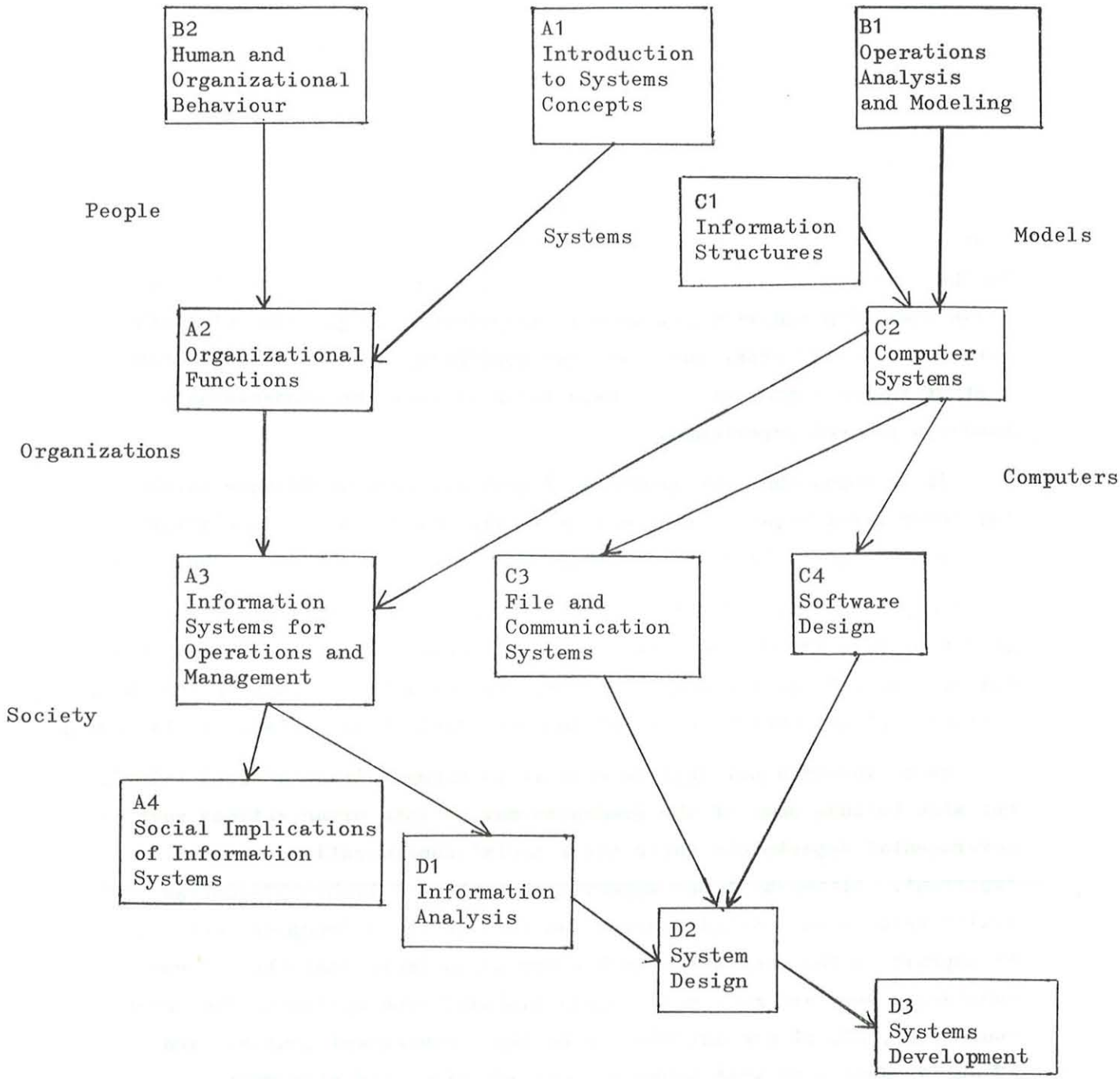
Figure 3

The Thirteen Courses

The A group of courses are preparation for the organization and information analysis aspect of the program, whereas the C group look at the basic computer technology as this is applicable to the program.

The B group of courses sits between these two and form the background for each. These courses include operations research on the one hand, and introductory behavioural science and organizational theory on the other. While mentioning background, it is appropriate to comment on the prerequisites that the program assumes. These are completely outside the program and consist of elementary mathematics (up to and including linear algebra), and also courses in elementary statistics, computer programming, economics and psychology. These subjects taken at first course level only would suffice. The courses in the B group focus the prerequisite background fields which are not readily available at undergraduate level. These courses are also directly applicable to both the A and C courses in the sense that, for instance, operations research can be used to analyse operations both of a corporation and of a computer system. Similarly the behavioural aspects of both the computer systems code and the organizational side must be taken into account. Thus the B courses naturally come between the A and C courses, and they also feed naturally into both information analysis and system design.

Finally the D courses form the heart of the program--these purport to deal with actual information system development. Figure 1 gives the courses in prerequisite-diagram form.



Information Systems Development

Figure 4

Course Relationships

After the background courses, courses A2, A3 and D1 take the student through the information analysis aspects of the program, while C2, C3, C4 and D2 really give a comprehensive set of courses for learning the aspects of system design.

Finally, it is desirable to have a projects course and this is the course D3. It is not so well specified since it is felt that the content of this course is made dependent on the facilities available to the individual institutions. For example, a university with very close ties with industry may find it is possible to give the students actual industrial experience, whereas others may need to be given some sort of in-house project. The main thing is that the students have hands-on project experience.

It is hoped that the products of such a course would have skills for entry level requirements, and there are six key words describing the areas of knowledge that the program aims to teach about. These are:

People, Systems, Models, Organizations, Computers, and Society. It can be seen from Figure 4 how the various courses relate to these areas. For instance, B2 is related to people, perhaps more than any of the other courses (but not exclusively), B1 to models and A1 to systems of all sorts.

A4 is included not just because it is thought to be a "good thing", but also because some of the graduates may go into organizations such as governmental departments where these social considerations are most important. Although it may appear to be out on a limb, this course really belongs to a valid progression through the A group of courses. A1 appears on the diagram in such a way as to imply that the systems considered here are mainly the organizational-type systems. The three courses B2, A1, B1 are intended to be basic conceptual courses, and A2 and A3 deal with what organizations are like, and also what information systems appropriate to those organizations should be like. The course A2 talks about the functions that organizations carry out and that largely defines what types of information systems are required to support this. It defines what those functions are; the way in which the system will support those functions is also defined. The course A3 looks at the next higher level, the tactical level and the strategic level or organizational systems. These concepts will feed into the even broader context--the context of society and what sort of information systems are appropriate, as considered in A4.

Looking now at the C group, C1 is a basic conceptual course also, similar in content to the course I1 as defined in Curriculum 68 [Ref. 1], entitled "Data Structures", being taught widely from Vol. 1 of Knuth [Ref. 6]. As a course it is not too esoteric from a computer science point of view, but in practical terms it describes the uses to which stacks, trees, etc. are put. C2 is the central course in computer systems, and is concerned with the hardware/software/architectural point of view, but again with the practical applications in mind. Rather than discussing the design of such systems, the course deals with hardware modules and the software that drives them, and this is a very introductory course in computer systems.

C3 and C4 are again of a technical nature, but these are somewhat more oriented toward this curriculum. C3 is about file and communication systems both in the hardware/software context and in the user-oriented package context. System design is tending more and more to make use of file handling packages which are currently being developed. Communication systems are concerned with the hardware/software level, but not with the very technical engineering details. C4 is called software design, where software is here intended to mean programs or groups of programs which are designed to be run co-operatively and by people other than the designers. They must be an integral part of the system, and as pointed out by Professor Parnas (also a speaker at this conference), must adhere to certain design constraints and conventions. Finally the D courses are fed logically from the A courses and the C courses.

The courses were modified--new parts inserted--by discussion with other people such as reviewers and the groups from industry with which the Committee had discussion. For example the industrial side was concerned about the problem of equipment conversion considerations. The Committee then redesigned certain parts of the courses to accommodate this.

This program was designed as a two-year course and with a normal load of four courses/semester, this left room for three other electives, meaning that the course could assume even more of an information analysis flavour, even more of a system design flavour or achieve some other general objective. It would also not be unreasonable to allow some courses to be taken at an undergraduate level, so that this would be specific preparation for a one-year graduate program or rather a

five-year Bachelor/Master's program, aiming on entry for that particular Master's specialty. Supporting this, the first year is in some sense more background, and the second year is the more specialized type of course. For instance, computer science graduates would be expected to have C1 and C2 and possibly some knowledge of C3 and C4, but the latter is not common and cannot be assumed. Similarly, an undergraduate business major would know a certain amount about A1 and A3, although he would almost certainly not have the same kind of systems orientation that would be provided here. This would really apply only to a business or computer science undergraduate who knew that this was the graduate course he wanted to take so that he could take the appropriate courses for it. If all these conditions are fulfilled, then it is possible to imagine the program being taken as a one-year post-graduate course.

To tailor this to the MBA business school degree or computer science degree, some whole courses can be taken, and others can be combined, for example (1) a course which is a combination of two A-courses (A1, A3) and (2) courses from a combination of two C courses each can be made up in an appropriate way to form a course for inclusion in a business school degree or a computer science degree. Such a graduate of (say) a computer science course would not have the training in information systems as he would have had had he taken the program as set out here, but on the other hand, he would know more about them than some of the computer science graduates who are emerging now from universities.

Finally, the Committee has put a great deal of thought into how to bring all these concepts to bear on any given course. For example, in course C3 on file and communication systems, a teacher in computer science can take this material and absorb it into a computer science course. This is to a certain extent true, but one of the aspects of file systems is the behavioural aspects, and this should be emphasized fairly strongly, and C3 has been designed with this in mind, although it is hoped that computer scientists will still feel that they can use such a course. This principle has been applied to each course, in order to exploit the interaction between disciplines.

Discussion

Professor Verrijn-Stuart then opened the discussion by asking what the prerequisites for the course were, with particular regard to the mathematical background. By way of answer, Professor Ashenhurst referred the questioner to section 3.1 of the report, which he amplified by saying that the elementary statistics, economics, psychology and computer programming would probably be contained in one course, and the mathematics would depend entirely on what each student had done in his undergraduate course. Probably, though, he would have covered enough mathematics in his first two undergraduate years. Two courses, one in finite mathematics and one in linear algebra would be sufficient. The psychology course is a very elementary course just to lay down the framework for this behavioural aspect of the program. Professor Verrijn-Stuart followed this up by asking what other sort of education the bachelor would be expected to have, to which Professor Ashenhurst replied that he thought that it could probably be applied to almost any other subject interest, since after all information systems can have all sorts of applications. He agreed that almost any graduate would be suitable for this program, provided he had the prerequisites.

Dr. Williams then made the comment that he thought the prerequisites were such that it was possible for them to be picked up early in the program even by people who did not have the formal qualifications such as graduates from other disciplines and people who have already spent some years in business but then return to a university to follow one of these courses. Professor Ashenhurst agreed with this view, but qualified it by saying that there were two types of prerequisites--those which were desirable in the sense that the student should be able to make up in these subjects that which he is missing. Following up his previous point, Dr. Williams then suggested that many students, when studying computer systems, would become bogged down in one particular operating system, and that rather than operating systems as such, should the students not be given a course in computer architecture. He asked whether the Committee had considered this. This was answered by Professor Ashenhurst by saying that he hoped that the courses C2 and C3 would in fact deal with operating systems in general and not become oriented towards one system only. Professor Dijkstra said that this course was all right for vocational training in management and as such it is greatly influenced by soft sciences. He asked whether the content

of the course is sufficient to improve the ways in which computers are used, and Professor Ashenhurst replied that he thought it was.

Third Lecture

Professor Ashenhurst opened the third session by saying that he intended to devote most of the time remaining to him to discussion. The previous day's provocative exchanges would provide a good starting point. However, before the discussion he wished to make three small points.

Firstly, it is necessary to correct a wrong impression which was apparently given previously. The aim of the curriculum is not to produce people to design system software or computer systems, but to produce people who can use these tools. It is possible for the system designer to specialize to a certain extent, but what is essential is that he should be familiar with all the currently available techniques and software packages and be able to use them.

Secondly, it is worth giving a further paradigm to demonstrate another facet of the structure of the thirteen courses comprising the curriculum. This particular approach (Figure 5) hopefully provides some new insight.

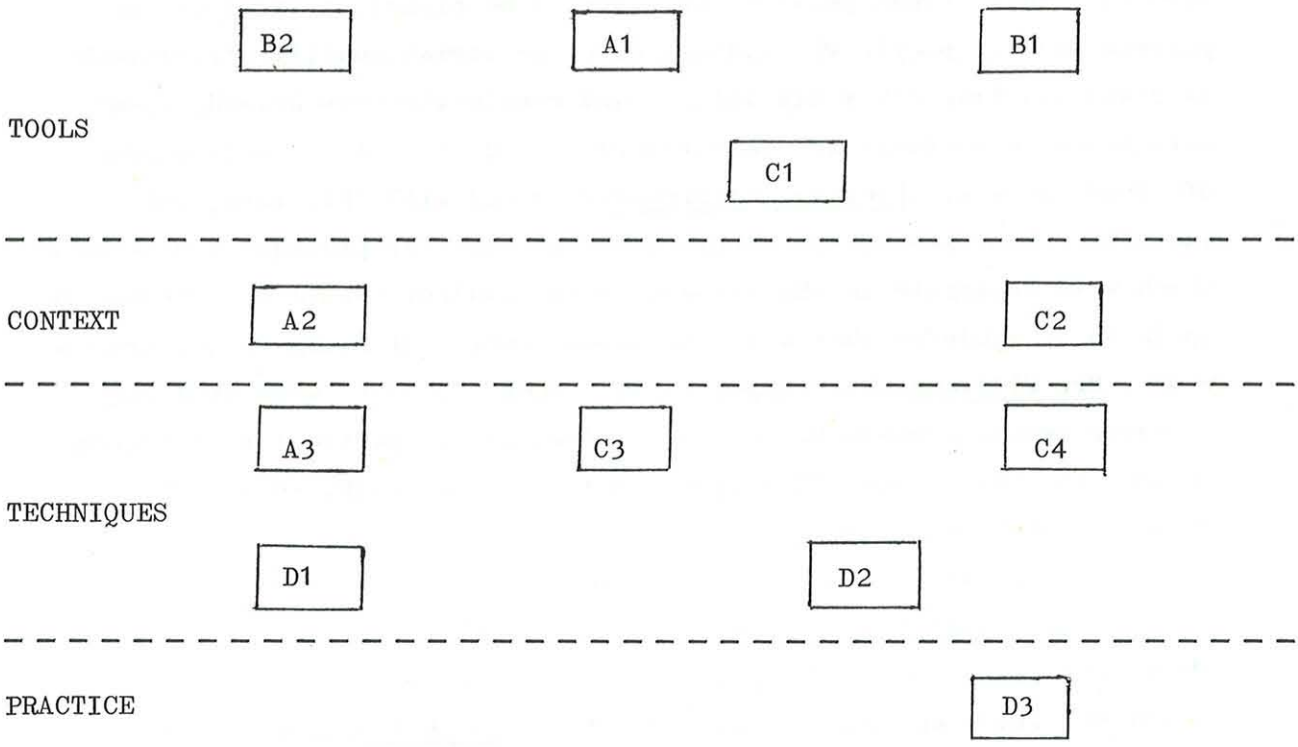


Figure 5
Role of different courses in the curriculum

As can be seen from Figure 5 the courses are separated into four groups:

(a) Tools, in terms of which the student will think.

(b) Organizational functions and computer systems together form the context, or environment, in which the student must work. In the case of computer systems especially it is important that the emphasis should be on context. The purpose of the C2 course is not to teach the student computer architecture or software engineering, but to provide a broad understanding which will enable him to work with and use computer systems.

(c) Techniques. This is the heart of the curriculum in that here we are trying to teach the student "how to do something" as distinct from teaching him the tools and context which he will have to understand first.

(d) Practice. Projects provide the student with an opportunity to put into practice all that he has learned in (a), (b), and (c).

Thirdly and finally, a word about the expression "systems analyst". This is a rather misleading expression, for it implies the wrong things about how to work with information systems. It implies first that one should analyze systems rather than analyze information, and secondly it implies an approach which defines a system and then sets about analyzing it. There are properly two distinct functions: Information Analysis and System Design. The curriculum is trying to encourage an approach which analyzes the information needs of an organization, and then designs a system to meet those needs. Of course, to a large extent these two processes run concurrently, but the essential condition is that system design does not preempt information analysis.

Discussion

At this point Professor Ashenhurst proposed that the rest of the session be devoted to discussion.

Professor Verrijn-Stuart opened the discussion by saying that he felt that there was still confusion as to the precise objectives of the curriculum, and he asked Professor Ashenhurst to reiterate them briefly.

Professor Ashenhurst replied that they were (a) to provide a two-year graduate program that would produce people who would be able to function in entry-level positions in groups dedicated to the development of information

systems, and (b) to suggest courses which might be incorporated in existing MBA and computer science MS courses in order to provide students in these courses with sufficient knowledge to enable them to function well in an information systems environment.

Professor Verrijn-Stuart considered that it was important that the curriculum should be directed towards producing the professional approach to the subject, rather than an approach directed towards research.

Professor Page pointed out that one criticism which might be levelled against the curriculum was that it merely fitted the graduate for an apprenticeship, without giving him a deep understanding of the problems underlying organizations and information systems. He implied that the academically oriented British and European universities might find it difficult to undertake providing a curriculum such as the one proposed by Professor Ashenhurst.

In reply to this Professor Ashenhurst said that the motivation for providing the curriculum was that there was a need for people with this kind of training. He felt that information systems were not being designed successfully because the conceptual foundations necessary were neither known nor disseminated through the educational process. He added that not only was education necessary, but also research into the nature of information systems, with the aim of finding a more formal basis for the subject as a whole.

Professor Page then asked whether Professor Ashenhurst thought that his curriculum would attract people in the top five percent as, he claimed, computer science courses do. If not, then there would be justification for the courses to become means of merely imparting knowledge. The best people are attracted by having to solve difficult problems.

Professor Ashenhurst claimed that there are different motivations, and that it is therefore quite conceivable that the top people would wish to work with information systems rather than in more academically oriented disciplines. Moreover, while admitting that he hoped that good people would wish to work in this field, he said that it was not disastrous if the top five percent were not attracted. He hoped however that people of the highest calibre would be attracted to do research in the field of information systems, if not to work directly on their design and implementation.

Mr. Land felt that the problems associated with information systems were in fact so difficult that they did need the best people to solve them. What was needed at this stage was an identification of these problems.

Professor Ashenhurst agreed, and suggested that when the problems had been better defined, then the best people would be attracted to work on them.

Dr. Florentin criticized the curriculum on the grounds that it was too rigid and standardized. He felt that a curriculum such as the one proposed should merely provide a framework which would allow each university to implement it as appropriate to its own situation. In particular Dr. Florentin criticized the artificiality of constraining each of the thirteen courses to the same length in time. He also felt that the curriculum did not concentrate on providing a sufficient conceptual grasp of the subject.

In reply Professor Ashenhurst claimed that it was necessary to provide detailed subject matter for each course in order to make it clear what he was talking about. He reiterated that the curriculum could be used as a framework without necessary adherence to the detailed course subject matter, and maintained that in fact it was rich in conceptual material.

Professor Dijkstra was concerned that the life-time of the material of the courses was too short. Too much stress was placed on current techniques in management science which were unlikely to be valid in ten years' time. He added that it was questionable whether vocational training, as this curriculum seemed to be, should be done at universities at all.

Professor Ashenhurst said that in his view the curriculum provided the student with a conceptual framework into which he should be able to assimilate new ideas and techniques.

Professor Dijkstra questioned how much scientific value there was in the kind of curriculum proposed by Professor Ashenhurst. He suggested that a subject having scientific value was one which, at least, embraced a coherent body of knowledge and understanding of its own. It was dubious whether the study of information systems could claim such a distinction.

Professor Ashenhurst admitted that the study of information systems relied to a considerable extent on the body of knowledge and techniques of computer science, but he maintained that such techniques were by no means sufficient for a full understanding of information systems, and that it was necessary for new techniques to be developed.

Dr. Florentin, in response to the attacks, direct and implied, on the validity of the study of information systems as a scientific subject worthy of intellectual pursuit, then expressed his view that there were topics in management science that could be taught at great depth, and that therefore there could be a great intellectual challenge which would secure the interest of the best students.

Mr. Bromberger then made two points. First, he was convinced that it would be better to teach the organizational concepts of the curriculum on top of a knowledge of computer systems, rather than in parallel with it as proposed by Professor Ashenhurst. Second, in his experience, those students who were particularly interested in computer science were not at all attracted to working with information systems.

Replying to the second point, Professor Page suggested that it would be wrong for computer science to become a purely research-oriented discipline. Computer scientists must be willing to look at how computers are used. As an example of the dangers of a too narrow outlook he cited the case of the mathematics/statistics departments in American universities which had become embedded in measure theory with the result that they never looked at data.

Professor Ashenhurst, agreeing, said that in general, computer science courses do not look at problems in an entirely practical way, i.e. from the point of view of organizations, and he suggested that his curriculum therefore filled a particular need.

Dr. Hanani then asked a question concerning the prerequisites for a student starting the proposed two-year curriculum. He pointed out that Professor Ashenhurst's proposals were geared to the American university system and would be difficult to implement in European universities, because it was possible only in American universities to obtain a liberal undergraduate education which covered a number of diverse disciplines, for example, mathematics and psychology.

Professor Ashenhurst agreed.

Professor Gilles then made four points:

(a) In some Scottish universities it was possible to take maths and psychology together in an undergraduate course. (b) In his view, some of the top five percent of students were motivated towards commercial work. (c) European universities of the Middle Ages were vocationally-oriented. (d) He was of the opinion that the curriculum as a whole looked too easy. Those courses which constituted the main body of

knowledge of information systems were descriptive, and therefore too easy. In particular, the courses on the computer systems side appeared to be the easier ones from the discipline of computer science, and not the more rigorous ones.

Professor Page agreeing with Professor Gilles, said that he thought that the courses of the curriculum amounted more to "listening and collecting" than to "thinking and doing".

Professor Ashenhurst felt that in spite of what the previous two speakers had said, it would be possible to design the individual courses of the curriculum to contain substantive material which was both challenging and testable.

Professor Melkanoff asked Professor Gilles what courses in computer science did he feel to be the harder ones. Compiler design? Logic circuits? What would Professor Gilles like to see in the curriculum?

Professor Gilles said that he would have hoped to see courses on automata theory and sequential processes included in the curriculum. As it stood, he felt that the hardest courses were A1, B1 and C1.

Professor Ashenhurst said that he believed that a traditionally hard course in computer science, like that of compiler design, was only hard in the days before it crystallized into a formalized discipline. However, he thought that there were no parallels to this in the proposed curriculum. On the other hand, he was of the view that the study of information systems would be systematized, but perhaps not so much as computer science subjects have been, because information systems are embedded in a "fuzzy" world, whereas compilers, for example, are embedded in a deterministic piece of hardware, the computer. He also thought that the proposed curriculum provided a suitable framework for the evolution of the subject.

A number of people expressed the view that such evolution was necessary, and hoped it would take place.

Professor Dijkstra reiterated his concern that the curriculum was geared too much to the teaching of current techniques and methods. He cited the problem of program verification, stating that the techniques presently used and taught were known to be inadequate. His fear was that the curriculum would produce people who would continue using these inadequate techniques long after they became obsolete. These people would be tied to the status quo and to the need to prolong it.

Professor Ashenhurst said that he hoped that the proposed curriculum would lead people to question rather than to accept situations.

References

1. ACM Curriculum Committee on Computer Science. "Curriculum 68: Recommendations for Academic Programs in Computer Science," Communications of the ACM, vol. 11, no. 3 (March, 1968), pp. 151-197.

2. Ashenhurst, R.L. (Ed.) "Curriculum Recommendations for Graduate Professional Programs in Information Systems," --A Report of the ACM Curriculum Committee on Computer Education for Management, Communications of the ACM, vol. 15, no. 5 (May, 1972), pp. 364-398.

3. McKenney, J.L., and Tonge, F.M. "The State of Computer Oriented Curricula in Business Schools 1970". Communications of the ACM, vol. 14, no. 7 (July, 1970), pp. 433-448.

4. Teichroew, D. (Ed.) "Education Related to the Use of Computers in Organizations (Position Paper--ACM Curriculum Committee on Computer Education for Management), Communications of the ACM, vol. 14, no. 9 (September, 1971), pp. 573-588.

5. Simon, Herbert. The Sciences of the Artificial. The MIT Press, Cambridge, Massachusetts (1969).

6. Knuth, D.E. The Art of Computer Programming, Vol. 1: Fundamental Algorithms. Addison-Wesley, Reading, Massachusetts (1968).