RECORD OF SOCIETY OF ACTUARIES
1977 VOL. 3 NO. 4

ACTUARIAL SOFTWARE

Moderator: GODFREY J. PERROTT. Panelists: JOEL C. MAGYAR, J. RAE JAMIESON, ALBERT J. KLEINBERG, JR.

I. HARDWARE

MR. GODFREY PERROTT: The first area we will address is hardware. So that we have a common frame of reference, I will define the five categories into which I have divided hardware. These are main-frame computers, minicomputers, microcomputers, time-sharing, and remote job entry. In the context of this session, these terms have the following meanings:

1. A main-frame computer is a medium to large size computer, usually with a substantial amount of memory. A large amount of online storage may be connected to it through disk drives, and it is capable of high speed input-output operations using magnetic tapes and line printers. It typically costs from $250,000 up into the millions of dollars, and is under the control of the Data Processing department.

2. A minicomputer is getting harder and harder to define. For the purpose of this session, we are considering the smallest minicomputer to be the IBM 5100, whose cost starts at around $10,000, up to much larger machines costing up to $200,000. A minicomputer is usually designed for interactive use and usually does not have very strong file processing techniques. A minicomputer could be under the control of the actuarial department of an insurance company.

3. A microcomputer is a relatively new entry to the marketplace and is primarily directed at hobbyists. It can be characterized by being cheap (under $5,000) and being much more of a do it yourself computer than either minicomputers or main-frame computers. Frequently, microcomputers are sold in kit form and the service arrangements made for them are much more like a home stereo as opposed to those we normally associate with computers.

4. By time-sharing, we mean using a terminal to interact with a computer. This can be either a computer provided by a commercial time-sharing vendor, or a company's inhouse main-frame computer.

5. By remote job entry, we mean using the terminal to submit work to a computer which is then processed on a delayed basis and not interactively. After the work has been processed, it may either be printed on a printer at the central computer site or it may be possible to access the output through the terminal.

MR. JOEL C. MAGYAR: Today in the U.S. and Canada, over 100 companies offer commercial time-sharing services which can be made available to you by installing a teletypewriter or other data terminal in your office to

communicate via telephone lines with the computer. In addition, the major computer companies all have software available to enable a company to install a time-sharing system on its main-frame computer.

I will be addressing this section of my remarks to interactive time-sharing systems, those which enable multiple users to gain simultaneous access and to interact with the system in a conversational mode.

The typical actuarial problem requires comparatively little input and output, often with no files of data; but it often requires large and complex calculations. For these purposes, the least expensive and slowest type of terminal, the teletypewriter, which operates at 10, 15 or 30 characters per second serves quite well.

Time-sharing offers many attractive benefits to the actuary:

1. Ease of use - Terminals are extremely easy to use, and the software and debugging aids available make programming quite simple and fun to learn. Any of you who have grappled with the complexities of job control, for example the JCL necessary to define the files, the input and output devices, etc., which must straddle both ends of an IBM/370 PL/1 program, will find the simplicity of such functions under a time-sharing system to be a joy. Access to the computer requires only a few brief commands from the user.

2. Man/machine interaction - Interactive time-sharing permits direct, instantaneous communications between the user and the machine. You test and debug your programs while you write them, as the computer checks and aids you on each step.

3. Fast turn-around - With interactive time-sharing you don't wait to see the results of your jobs; you enter your data and can have the results available to you nearly as fast as the computer can calculate them. That frustration of waiting for several hours or overnight to get your output back from your in-house computer center, only to find that an item of input was mispunched and you have got garbage, is eliminated. You can rerun immediately.

   We have found this a great advantage for such jobs as running asset shares for product pricing. We can make a run, see the results, change assumptions as necessary, and rerun in a matter of minutes.

4. Flexibility - You buy only as much computing power as you need. What you pay the vendor will depend on how much you use their services. Generally, the only fixed costs are the rental of a terminal and possibly minimum service charges. You can have access to any number of vendors and take advantage of differences in pricing structures, languages, and ready-made program libraries.

The benefits I have just mentioned apply whether you are using a commercial vendor or an in-house system. Additional benefits are available when you go outside:

1. Choice of programming languages - Many services offer a choice of languages so your users can work with the language that is best for

your problem. I will be discussing some of these languages a bit later. Most likely your company would support only one language in an in-house time-sharing environment.

2. Program libraries - There are many ready-made programs available for specific applications. Using them can bring significant savings of dollars in programming costs and get you going much sooner. For example, if you are interested in corporate modeling, there are quite a few generalized packages available.

It can be extremely expensive to set up an in-house time-sharing system. An operating system must be purchased or rented, and installed. If you don't have teleprocessing capabilities on your main-frame computer, a substantial investment in hardware and software will be necessary. Assuming you already have teleprocessing capabilities, an operating system to support APL will cost from $400 to $1,600 per month. Furthermore, systems personnel must be trained to provide technical support and maintenance for the system.

In the short run, it is preferable to get experience in time-sharing with outside vendors. Costs can vary widely from service to service and, unfortunately, are structured in such a way as to make comparisons between services very difficult. Also, there are high and low cost companies. Those services offering the more sophisticated equipment and/or large technical staffs must amortize their comparatively higher overhead and are more expensive. It may pay to use a smaller service if you don't need the computer power or if you will be doing your own programming.

Typically, there are three basic charges for services:

1. Usage or terminal connect time - Usually an hourly charge for the period of time for which you are "on line" or connected to the computer.

2. Central processor time - Specific charges for each minute or second of time during which the computer is working for you.

3. Storage - Charges usually expressed as so much space per day or month for storing your programs or data.

In addition, the service may or may not impose a minimum monthly charge. (If you plan to make limited use of a number of services it might be worthwhile to concentrate on those which impose no minimum.) Typically, total costs using one 30 character per second terminal practically full time, should run about $3,000 - $3,500 per month.

It pays to benchmark test your typical job on several services. We we are currently using two services and have found that our asset share program costs double on one service versus the other.

Overall, interactive time-sharing can be quite cost effective, especially when your time is taken into account. You can have your results and make your decisions much more quickly than under batch processing.

As an example of the efficiency of time-sharing, I was recently asked to supply some complete tables of (1) present values of $1.00 due at the end of n months and (2) present values of $1.00 per year for x years with the first payment due at the end of 0 through 12 months, at split interest rates. The tables were to cover 40 years, 480 months. I assigned the problem to a Part 3 student who was able to write, debug, test and execute the program which produced the required tables in a form ready for reproduction in only two hours.

MR. ALBERT J. KLEINBERG: I agree that time-sharing and the ability to work in an inter-active environment presents the user with an extremely powerful tool.

However, compared with some of the alternatives available in the market today, it is difficult to justify the use of time-sharing in any but extreme situations.

Time-sharing presents at least two major weaknesses. First, it encourages the user to respond quickly. The computer will compile and/or test a program "instantly." The user must then respond to either the compilation diagnostics or the execution errors. He is discouraged from carefully reviewing the errors since this would involve monopolizing the computer terminal and increase the terminal connect time charges. This results in an increased likelihood of either incomplete corrections or additional errors.

Second, through the use of a cathode ray tube or typewriter, the user does a greater percentage of his own keypunching. For some reason, in many situations, the user does 100% of the keypunching. Clearly, from an economical basis, this excessive amount of keypunching cannot be justified.

In theory, these two weaknesses could be eliminated or, at least, controlled by proper management. In practice, these weaknesses are ever present.

Most of the advantages of time-sharing can be maintained in a properly managed remote batch processing environment. There are many computer services which offer almost instantaneous turn-around through high and medium speed terminals. This presents the opportunity to meld the advantages of time-sharing with the economic savings of a more traditional computer operation. The software available on many of the large facilities gives the user all the tools that are available in time-sharing.

In summary, if time is of the essence and cost is not a factor for a specific project, then time-sharing is the proper tool. In the more typical situation, it is difficult to justify the additional expense (both in terms of hardware and personnel) associated with time-sharing, for the small marginal advantage time-sharing presents over an efficient remote batch operation.

MR. J. RAE JAMIESON: Actuaries are dedicated hobbyists, so I am certain that we have in our Society, individuals who have built or purchased a personal computer of the "micro" variety. However, I will dispose quickly of this area by commenting that such hardware does not readily lend itself to a business environment. My principal reason, which may possibly be faulty, is that programming of such hardware is frequently done either in

binary machine language or a low-level assembler language. Hence, the already inscrutable workings of the actuarial mind are further clouded in software which is incomprehensible to any but the individual who produced it. It will become clear later on in the panel presentation that I belong firmly to the "documentation and control" school of software management, whether exercised by actuarial or data processing management. Hence to me, microcomputers represent a threat rather than a useful tool. Discussion, either for or against this view, is invited at the close of this portion of my remarks.

By contrast, minicomputers offer many of the advantages of time-sharing without certain of the disadvantages. Before extolling the virtues of these tools, let me point out several of their more important drawbacks.

They are slow, not in the physical sense, but in the data processing sense. They transfer data at what seems to be incredibly slow rates in many cases. Whether data is stored on magnetic tape cartridges or paper tape, the physical process of reading in or writing out a program or a small data file often ties up the machine and the person using it to an extent not imaginable to someone familiar only with time-sharing through a large-scale main-frame computer. As a reference point, an APL workspace is loaded into active memory on commercial time-sharing in so few seconds or fractions thereof that the user cannot distinguish the time required from the time taken to flash back the message "mission accomplished."

However, a large workspace for the IBM 5100 Minicomputer, for example, may take several minutes to load if the cartridge must be wound to another location and if there is a considerable amount of APL code to be converted from data storage format. And the same software written out on a magnetic cartridge could require 3 to 5 minutes. Unless the user is able to do some of the thinking work during this period there is a dollar loss, together with a possible frustration index. Machines which use floppy disk, such as the MCM 800, can reduce this idle time to more manageable proportions. Other models, such as Wang and Hewlett-Packard, for example, are not familiar to me, so I invite comment during the discussion.

The other feature which illustrates slowness is execution time for the software itself. As an example, APL code is interpreted, rather than compiled. When the main-frame computer is an IBM 370, or an Amdahl 470, or similarly powerful processor, large arrays can be manipulated in quite complex ways with response times measured in seconds rather than minutes. However, a typical pricing model computation done on an IBM 5100 took 3-1/2 minutes to perform, and took over 10 minutes on an MCM 800; each has the maximum capacity currently marketed.

There are other lesser disadvantages of "minis," most of which do not impact the research-type applications of a typical actuarial department. However, certain quasi-actuarial applications are probably not suited to most minis for one or more of these reasons:

1. Necessity for accessing one or more company data bases.

2. Necessity for developing large data files other than company data base files.

3. Applications involving massive logical operations such as computing, sorting, or editing.

4. Applications requiring massive output printing capacity, with or without complicated data formatting.

A typical such quasi-actuarial job which belongs within the data processing department and the main-frame computer is life policy valuation. However, many small ancillary valuations are ideal minicomputer applications. Within the next few years, however, the foregoing may be totally altered.

Now let's examine some of the advantages of minis. The most obvious and most readily quantifiable is hardware cost. Compared with time-sharing cost, minis are downright cheap. In our organization, we have cut our annual time-sharing bill from almost $50,000 to less than $10,000 by purchasing first one and then another mini. The total purchase price of the two was less than the $40,000 saved each year. Maintenance costs are no higher than rental of terminals had been, and one of the minis doubles as a so-called non-intelligent terminal for those applications where time-sharing is unavoidable.

Another less widely known advantage is that individuals can be trained more readily to use interactive languages. While FORTRAN and APL are natural vehicles for trained actuaries, so that they tend to rapidly assimilate the languages as working tools, there can be a psychological or even a technical barrier to the use of personal computing for personnel less mathematically oriented. This tendency is aggravated when such an individual is aware that every failure represents hard dollar costs incurred by the company with their time-sharing vendor. However, minis permit an individual to practice "for nothing" until sufficient confidence has been gained. In the actuarial department of our company the usage of interactive computing as a tool has grown from a start of two individuals alone to its present state, in which virtually 100% of the staff is involved. At least two companies have reported purchasing a minicomputer solely to train people on the use of APL, although I personally consider this too much specialization.

Finally, the presence of a mini in the actuarial department encourages its use for jobs which would otherwise fall into the familiar mold of clerical computation on the usual multi-column worksheet. Such jobs tend to be overwhelmingly boring for clerical employees since all too frequently they involve apparently complex calculations whose meaning and import is largely obscure. When such a task is converted to a programming challenge on the mini, together with data entry performed by someone with keyboard skills, the result is usually better job satisfaction all round, together with better computational reliability.

For those who are interested, the September 1977 issue of Scientific American is almost completely devoted to discussions of present and future hardware capabilities and what these will mean to us shortly. Also LOMA published a report entitled Minicomputer Utilization in Life Insurance Companies which shows the degree of penetration of this tool as of about one year ago, together with company by company responses regarding typical applications.

MR. TIMOTHY F. HARRIS: I know of two microcomputers that use BASIC and sell for about $600; for $1,200 you can get one that also has a color display screen.

At American States we have two HP9830A's and have just purchased a Wang PCSII. We had an IBM 5100 on a trial basis for several months and found that the HP and Wang were much faster. The HP's came with tape drives but we also purchased a hard disk unit and the Wang PCSII comes with a floppy disk unit; therefore data and program access time is of the order of microseconds. Both machines are able to access our data base through the use of "Data-Comm" programs and hardware purchased from the manufacturers. Both machines have much better program editing capabilities than we found with the IBM 5100.

We did not feel that the availability of APL was enough reason to purchase the 5100. In fact, we prefer BASIC since the clerical staff does some programming and has to be able to run and possibly maintain some of the programs written by the actuarial staff.

MR. THOMAS P. TIERNEY: I agree with you, Al, about time-sharing but for a different reason. On paper it looks very good. The problem is that its use requires a lot of personal discipline. People tend to be sloppy, get lazy, and that runs up the charges. Also, Al, if you're not using time-sharing, how do you get your computing work done?

MR. KLEINBERG: We are on remote batch with a commercial vendor and have a remote job entry station. We can request turn around that is instantaneous although it's not interactive.

MR. TIERNEY: Do you use standardized computation packages or is there a lot of personalized programming done in your shop?

MR. KLEINBERG: Both. The routine is standardized, but a fair amount of the work that we have been doing recently has required some special coding.

MR. JOHN MILLER: Mr. Harris, can the Hewlett-Packard or Wang or both be used effectively as a word processor? Also, what is "BASIC extended"?

MR. HARRIS: Both the Hewlett-Packard and the Wang can be used for word processing. They both accept alphabetic strings. You can store letters, pull them back out, change a line here and there, etc. We intend to use the Wang quite a bit in the life company for letters to agents and to insureds, ledger statements, etc. Hewlett-Packard does not follow the standard BASIC. Their new machine, which is much faster than the one we have, uses somewhat extended BASIC portions of APL and is getting into FORTRAN.

II. SOFTWARE

MR. KLEINBERG: The development of new computer software by the actuarial consulting firms can be summarized with three words: models, projections and, or course, ERISA.

The requirements that ERISA placed on the Enrolled Actuary in his selection of actuarial assumptions have resulted in modifications to the

pension valuation systems. The major changes have been the incorporation of select and ultimate rates of turnover and salary increase, and the use of retirement decrements. The select period, with respect to salary scales, begins with the valuation date. With the current high inflation rates and the requirement that the actuary use his best estimate, it may be desirable to use a select salary scale reflecting a decrease in inflation during the next 5 to 10 years and a lower assumption of salary increases thereafter. With the many difficulties an actuary has in selecting his assumptions, it is unfair to add the restriction of an inadequate valuation system.

Historically many actuaries have accepted a new census for each actuarial valuation. With the increased emphasis on gain and loss analysis, many actuaries are reconciling data between actuarial valuations. This reconciliation also creates a data base which facilitates the production of annual benefit statements. Because of the requirements of accrued and vested benefit disclosure, increasing numbers of plan sponsors have issued annual statements. Further, if the data is reconciled between valuations, the auditor is more likely to accept its accuracy or at least reduce the amount of testing he must do in order to satisfy himself as to the accuracy of the data.

This increased emphasis on improved data has resulted in the development of computer software to facilitate the collection and maintenance of employee data files.

Further, ERISA requires the actuary to maintain his information for a six year period, and without an efficient data base system the actuarial firm would soon be overrun with files.

To meet these new requirements, firms must develop data base systems which, from a single file, will (1) permit the recreation of the data used for any prior actuarial valuation, (2) faciliate the calculations of actuarial gains and losses, (3) provide accurate data for the production of employee benefit statements and (4) provide a basis for experience studies of turnover, retirement and salary increases.

During the last few years, there has been substantial interest in projection valuation methods. These valuation methods differ from traditional actuarial valuations in that the methods not only value the current plan participants but also measure the impact of future new entrants. A major reason for the increased popularity of these methods has been the availability of improved computer systems.

There are two basic types of projection valuations. The first is used to determine annual pension costs. Briefly, it involves a projection of the current census to determine the expected payouts during a specified period and the target value at the end of the period. The length of the period is dependent on many factors and would typically be at least 20 years and possibly as long as 40 years. The target value, which is the funding goal, could be the liability for vested or accrued benefits or the accrued liability under the entry age normal cost method. The present value of the future payouts and target value less the current assets is expressed as a percentage of future salaries or as dollars per active life year. The annual cost is then determined by applying this factor against either the current population or payroll.

The second type of projection valuation is not typically used to determine current costs but is used to determine a range of future costs under alternative assumptions. The plan participants are projected for a 10 to 20 year period. At the end of each year, a traditional actuarial valuation is performed using the normal actuarial assumptions. The projection of the plan participants and fund assets are completed on alternate assumptions which would, only coincidentally, be the same as the regular assumptions. The typical report to the client would include a combination of different assumptions for inflation, return on assets, rates of new entrants and possibly retirement and turnover. In addition, the systems usually permit plan amendments during the period.

The two types of projections are similar. However, the first is a valuation method which determines present values and current costs by projecting the plan participants. The second is not truly a projection valuation but is a projection of the employee data and plan assets which is then used to complete traditional actuarial valuations.

Clearly, these types of valuations would not be possible without sophisticated computer software. Most of the software developed for these systems require not only parameters but also moderate amounts of computer coding for each application. Because of the size of the computer needed to run these projection systems, many firms are using outside vendors rather than their own inhouse computer systems.

The third and most recent area of computing software development is the creation of computer models. One of the primary applications of this type of software has been the use of a model to simulate the performance of a pension fund.

Through the use of the Monte Carlo method these systems develop not only the expected return but also the possible deviation from the expected.

The Monte Carlo method is used by many computer models. It consists of repetitive trials with an assigned success probability. A record is maintained of the number of successes and failures. A success or failure is determined by comparing the probabilities with random numbers. It is possible to determine the mean and standard deviation or produce a visual graph of the result.

The pension fund models typically allow the client to select various economic assumptions and the mix of the fund between stocks, bonds and government securities. The purpose of the studies is to determine an appropriate asset mix, that is, a mix that considers both the client's ability to assume risk and the anticipated plan liabilities. The plan liabilities are usually determined by a projection valuation. It is also possible to create a model to determine the plan liabilities. However, with the speed and cost of the computers available today, it is preferable to use the actual plan participants.

MR. MAGYAR: Actuaries are not meant to be professional programmers. We therefore prefer languages which are easy to and easy to use. There are a number of programming languages which are suitable for our work.

When used in conversational mode through time-sharing, these languages can provide a high degree of interaction between the programmer and the computer throughout the programming and debugging process. Generally, each statement of the program is checked for syntax as it is entered; the computer signals any errors, and corrections can be made immediately. Once the program, or segment is entered and checked, it may be compiled into machine language and then executed in the usual manner of batch processing, or it may be executed immediately in what is referred to as interpretive mode, depending on the system used. Under interpretive operation, each source statement is translated into the appropriate machine language and executed before the next statement is translated.

FORTRAN has been widely used for a number of years for actuarial and scientific programming. It uses expressions and symbols similar to algebra for its computational and logical processes. Many different versions of FORTRAN exist: conversions of FORTRAN programs from one version to another can usually be made fairly easily. FORTRAN can be used to write programs efficiently and execute jobs completely in a time-sharing mode, used to write and debug programs under time-sharing which are later to be converted into production programs for use under a batch mode, or to write programs directly for batch processing.

BASIC (Beginners' All-purpose Symbolic Instruction Code) is probably easier for non-programmers to learn than FORTRAN. However, it is generally less powerful than either FORTRAN or APL. It was developed at Dartmouth College to provide non-programmers with a means of writing programs in an easy to use language resembling standard mathematics. Many systems offer their own enhancements to make BASIC more powerful. Since these are not standardized, using them may lock you into a specific vendor, since conversion to another system may be difficult. BASIC compilers generally emphasize rapid calculation and ease of use rather than efficient utilization of machine capabilities. Batch mode compilers for this language are rare.

APL is becoming more popular with actuaries. It was designed specifically to permit clear, concise expression of mathematical algorithms. It is probably the easiest to learn and most powerful language available for actuarial use. It is designed to work with vectors or arrays as its natural elements. Thus, to perform an addition of two matrices that in other languages may require a number of loops and a dozen or so statements becomes merely A+B in APL. In addition, APL contains many useful functions that cannot be expressed concisely in other languages. For example, only a single symbol is needed to perform the reverse summation so frequently used in calculating commutation functions.

APL can be learned and used in small pieces - what a student doesn't know won't hurt him. This statement cannot be made about most other languages. You can use the language with a minimum of study. If you need the advanced procedures, you will find the effort needed to master them well worthwhile.

However, because of the conciseness of the language, APL programs are sometimes hard to read or comprehend. Efficiency of program execution is also comparatively low since APL is generally available only in the interpretive mode and the program must be translated each time the program is executed.

Students knowledgeable in FORTRAN or BASIC may, unfortunately, need to "unlearn" some concepts of these languages. If the typical FORTRAN "do loop" is carried into an APL program, execution costs can be very high. For example, we found that tightening an asset share program to remove unnecessary looping resulted in a 2/3 cost saving in each run.

As with BASIC, some vendors offer extended versions of APL and the same caveat concerning being locked in holds here.

In addition to general programming languages, there are specialized languages that can be quite useful. For example, ACT (An Actuarial Programming Language), developed by Dr. David R. W. Jamieson while an actuarial student at Sun Life of Canada in 1970, is available from several time-sharing vendors. Written as a subset of APL, the ACT language extends the scope of APL into actuarial mathematics. The foundations of the syntax and some indication of its implementation in APL may be found in Jamieson's article, "ACT: An Actuarial Programming Language," in the January 1972 issue of The Actuary.

Other languages such as COBOL and PL/1 are available, but since they are oriented more to processing data rather than to calculation, they are more difficult to learn and to work with and less suitable for typical actuarial problems. Recently, some languages which enable you to describe worksheet problems in "English" have become available, but these are generally proprietary and have not been standardized.

MR. JAMIESON: Interactive computing, as it is used either in time-sharing or minicomputer environments, is ideal for research and development. Hence it is the most natural tool available to the actuarial department of a life company. Applications are limitless, as can be judged by consulting the LOMA reference quoted previously. However, asset share analysis or pricing models are the most frequent in the life business. Other applications include marginal tax rates, surplus analysis and dividend formulas, health insurance rate-making, etc.

As for language to be used, let me give you our own experience. To get the ball rolling and to encourage the use of personal computing, we permitted any language supported by a time-sharing vendor. At first we had mostly FORTRAN, but APL gained a foothold. After several years, we still permitted any language but advised the use of APL as the tool of preference. Our current situation is that APL is the only language permitted, either in time-sharing or on the minis. Perhaps others here today would be willing to share their views with us.

MR. TIERNEY: Joel, you mentioned FORTRAN as being appropriate for actuarial computation and PL/1 as not being appropriate. I believe that is bad advice although, I will grant you it is an opinion which is widely held. I would add, however, that it is a concensus among computer language theorists that we must get away from the first generation assemblers and the second generation monoliths like FORTRAN and COBOL and get into the third generation modular languages such as PL/1 and, of course, APL. A good example of this would be the action of the Russian government, which about four years ago, decreed that no batch languages will be used in Russia except ALGOL 68 and PL/1.

MR. MAGYAR: My comment was based on my personal experience, not that of my company. We had been using FORTRAN on an IBM 1130 for actuarial work but the EDP department decided that the machine was outmoded, too expensive to keep and would have to go. They wanted us to use their 370's. They would make the 370's available and teach PL/1. The teaching of basic PL/1 was a five week course given to actuarial students a half day each day. I believe there was another week devoted to some enhancements our people had to know to use our system. When they finished that course there were very few people who were able to be effective in PL/1. On the other hand, for APL, we'll either hand over a textbook to the student or send our people to a typical APL class which is four half day sessions, usually spread over a two week period. People who come out of this course can be effective and can use the computer. I have no argument that PL/1 is not a good language for professional programmers, but I think for casual usage it is too complex.

MR. JOHN D. HARSANT: I hope you will agree that in fact our work in this area is the same on both sides of the Atlantic. However, some of my practices are different from those which have been expressed today. We're a relatively large partnership with 25 partners, all Fellows of the Institute of Actuaries or the Faculty of Actuaries. Responsibility for the initiation, design and execution of software for actuarial work in the firm is vested in a committee, the membership of which includes a D.P. manager, two partners from our offices in London and Birmingham, and other partners to spread representation over the firm. Our D.P. department is completely under the control of the partners in the firm; we have our own main-frame which is resident on the premises of our principal office. The purpose of the committee I just referred to is fourfold:

- is to examine proposals to see if they're worth pursuing,
- to assign priorities to different jobs,
- to check on the technical content of the specifications, and
- to insure that when the work is done the other partners in the firm will accept that work.

To implement this work we have at present an establishment of two systems analysts, each of whom have two programmers. These are professional D.P. people who have no actuarial training, but they come to us with extensive D.P. backgrounds and they work in an office with an actuarial staff. We do this to solve the problem of communicating actuarial concepts to data processing people. This is one of the major problems we have had to face, and we've overcome it by taking two measures.

First, we have partners who are competent systems analysts and who design the large systems or act as consultants to the small systems. Then, we have the D.P. personnel directly concerned with the actuarial work sitting in the same room and very close to actuarial personnel who

are engaged full time on actuarial work. That way we establish a situation where any questions can be passed across informally. We find that it works very well. The main-frame is on the same floor as these personnel and if there are small errors in work which is being processed in batch mode the person who's running it can query whoever is running the job.

The equipment we use is an ICL 2904. I'm told it's similar to a Univac 9030. Access to this system is through a data control clerk. The actuarial staff do not have direct access, but are on the same floor. We obtain a strict control over the processing work without making it too remote. For our offices outside Liverpool, we have the work sent in by post or train. This is quite satisfactory for the turn around required for pension fund valuations.

The valuation of our pension plans, which is the major part of our work, is carried out by one set of programs which are parameter driven. Provided the data is submitted in standard form, which it usually is, no programming of any kind is required to value the benefits on any basis. This system is written in COBOL with FORTRAN subroutines. The idea being that the COBOL does the testing of the data and the output procedures, and the FORTRAN does the actuarial work. This we find very satisfactory and at the moment I see no reason why we should change. Interactive systems have been experimented with and used in certain marginal cases in my firm, but I'm very concerned to keep my actuarial staff from programming too much.

I ought to point out that in the United Kingdom labor is still relatively cheap compared with computer facilities. Wages convert at $4 to the pound as compared with current exchange rates of $1.75 to the pound, so labor is less than half the price it is over here. This means that the point at which interactive programming is cost effective is further away in the United Kingdom than it is in this country. I am in favor of using it where it is cost effective. The scale of operation is important when you're considering interactive programming. We have offices in Washington, D.C. and Toronto which are relatively small. For these offices it makes sense to use a time-sharing service with APL where the overhead for the system is low, even though the running costs may be greater. However, in the United Kingdom where we have enough volume to support a main-frame, it doesn't make sense. What I'm doing is trying to keep my staff away from programming except for applications where I think APL is most appropriate. If the application requires to be run more than three or four times it is converted to a parameter driven system on the main-frame to impose the discipline of the office on it.

I am, however, in favor of using interactive programming in my region of England (Liverpool and Manchester). I am instituting a training program to encourage the use of APL amongst the offices there.

III. MANAGEMENT AND CONTROL

MR. PERROTT: When using computers to solve actuarial problems is discussed, both the hardware and software areas usually receive plenty of attention. Unfortunately, the management and control of computers does not receive enough attention. In particular, verifying that a computer program does what it is meant to do, and documenting that verification receive scant attention.

Most actuarial students treat the computer in the same manner as they do a calculator. It's a tool they are competent to use, and which they do not need to explain how they used. From a control point of view, the computer is far more complex than a calculator and can be misused in subtle ways that even the person using it is not aware of. If you are using calculations performed on a computer to make decisions, you should document the process at least as carefully as you would a manual process. For a manual calculation, you require work papers to be dated and initialed by author and then checked by another person to ensure that your decision is based on good information. Thus, for a computer calculation, you need to show how the program was checked to see that it did what was intended (by hand calculations, by an independent second program, or by matching an existing program). In more critical cases, you might also document exactly which version of the program was used for the calculation by listing out the source code immediately after the calculations are performed.

I would like to make my own plea that the computer not be treated as a mythical black box, but as a top caliber actuarial student. Even though you personally have enormous faith in a particular actuarial student, you would still want his hand calculations checked by someone else. In the case of the computer, unlike the human, every calculation does not need to be checked, but at least one calculation of each conceivable type must be checked.

MR. JAMIESON: As I mentioned earlier, I am a strong believer that every area in a life company operation must be properly managed, even the actuarial department. When computing services are decentralized using time-sharing, minicomputer or both, the responsibility for part of the software development shifts from the data processing manager to the actuarial manager.

Several years ago few standards existed in this area. Perhaps the time-sharing vendors realized that user cost over-runs would mean bigger cash revenues for the vendor in the short-term future, so that little help was volunteered. However, I found several responsible vendors who believed that everyone's long-range interests were best served by operating within well-defined guidelines. With their help, and with the valuable ideas of my associates in our company, I came up with a set of Documentation Standards for time-sharing which have proved valuable.

We have found it necessary to have control of both tape cartridges and minicomputer usage, and have had to schedule minicomputer time in advance because usage averages in excess of 40 hours per week per machine. If anyone is concerned about the possibility of over-control, I'll be happy to address that point during the discussion. Meanwhile, let me state that our prime objective was to make interactive data processing attractive to as many potential users as possible, while retaining sufficient documentation to assure ourselves of both validity of output and continuity in the event of staff turnover. The results have been quite satisfactory, as many of our applications have proven to be useable and understandable by a number of users besides the originator of the software.

MR. KLEINBERG: I was asked to discuss the management of data processing resources in the consulting firms. This is a difficult subject since very few consulting firms actually manage their data processing facilities.

The problem is acute with multi-office consulting firms. Basically there is very little central management of the data processing function within the consulting field. Each office is permitted to reach its own conclusion as to the best method to provide for its data processing needs.

In many cases each office is independently developing its own computer software. In the most extreme situations, each office could have its own valuation program written for its own machine and written in a language which is different from the languages which are used by the other offices.

This situation has developed as a result of the historic independence of individual consulting offices. This independence, which is a positive factor in most areas of a consulting firm's operations, acts as a detriment to the proper development of data processing within the consulting firms.

Clearly, there is the possibility of substantial cost savings if the development of software is centralized. Further, the quality and sophistication of the software could be substantially improved without corresponding increases in expenses. The centralized development of software would place some hardware restrictions on individual offices, but these restrictions should be manageable.

In order to improve the management (and hence the quality) of data processing, the consulting forms must meet several objectives: first, the software development must be reasonably financed: second, the legitimate needs of the user must be satisfied and, lastly, the software must be designed and developed by the actuaries who will use the systems. I do not think that a consulting firm can solve its data processing needs by relying on EDP specialists.

In summary, the data processing facilities within the consulting field have developed careful planning. However, because of the importance of data processing and the dollars spent on hardware and software, the consulting firm that fails to properly manage its data processing facilities will find itself at a competitive disadvantage.

MR. MAGYAR: Storing data on line for immediate access is comparatively expensive if you are using outside time-sharing services; it can typically run to over 20% of monthly costs.

Unfortunately, storage costs are probably the most difficult to control. Costs can be kept down, however, if you keep your programmers aware of the expense they are incurring in this area.

In an APL time-sharing environment, each user of a time-sharing system has his own individual series of workspaces assigned under a unique control number. Storage charges build up as these workspaces become filled. Insist that your people clear out any data files or workspace areas that were filled during a terminal session if the data is not going to be used again.

FORTRAN time-sharing systems are file oriented. Some systems are dynamic, i.e., file size is determined as you execute, so that you are assigned and charged for only the storage actually needed. Other FORTRAN systems require that file size be determined and specified beforehand. Since it is

a fairly difficult procedure to estimate how much storage a program and associated data will require, you may find that you are defining and paying for more storage space than you really need. On the other hand, a job run on such a system will "bomb-out" if you have not specified sufficient file space and you will face the additional expense of re-running.

If several users are working with the same program, see that each doesn't store the program in his own workspace or files; rather store it in a central area accessible to all users.

Ask the question, "must this be stored on-line?" Most services enable the user to dump his programs and data onto tape for storage at the computer center via instructions given at the terminal. If the programs or data are to be used infrequently, it may be cheaper to pay the low tape storage charges and remounting charges for manual reloading onto the system when requested, rather than keep the information in "on-line" disk files. Some vendors will be able to mount your tapes within an hour or so after you send a request over your terminal or by telephone. Other vendors may need overnight to respond.

Monthly bills usually show storage charges separately for each user as they do for connect and central processor unit charges. Unusual usage of storage will be apparent and can be analyzed and rectified if necessary.

MR. JAMIESON: I also am a strong believer in control and documentation. It has to be done by the management of the department. In our case we require, for time-sharing at least, that it be done even for APL. We require the code to be written out ahead of time and debugged at the desk. The programmer must do thinking work at the desk, not when he's connected with the machine. In the case of a small error for which the machine gives a diagnostic and which can be corrected right away, he is permitted to continue at the terminal; but we do not permit a person sitting at the terminal to try to figure out what they've done wrong. Such a person is bumped off the machine for a variety of reasons, one of which is somebody else is waiting to use the machine productively.

MR. JEFF ROBINSON: With regard to storage, Joel, have you got any recommendations on the use of back up? Most of the time-sharing vendors provide back up and I know we have really not relied on that and purged our files.

MR. MAGYAR: It is generally available. Also, most services will keep several years of a certain time of day status of the core and the associated files as back up so you can ask them to put the system back up at some prior date and thus save storage charges that way. You're taking a chance, because it's under their control, not under yours. I would not recommend it as sole back up source.

MR. CHRISTOPHER H. McELVAINE: I'm a firm believer in controls and documentation. We have an APL terminal which is used by my actuarial department and I struggle to determine how you differentiate between interactive programming and interactive actuarial thinking. I would like to find a way of getting our actuarial students to sort out their actuarial problems first and then apply their interactive approach to the programming rather than the actuarial thinking.

MR. PERROTT: My function is actually managing a data processing area and we use interactive debugging. We use COBOL because our applications are more data processing. We strongly encourage the programmers to code their programs, get item keypunched and loaded into the system before they ever go near the terminal, then to use the inter-active power when they're actually debugging it.