Australian Computer Bulletin

VOLUME 2, NUMBER 1, FEBRUARY 1978

CONTENTS THIS ISSUE:

Figure 3 Enquiry stations (see story starting page 6)

Published monthly by: Australian Computer Society Incorporated, P.O. Box 640, Crows Nest, NSW, 2065

Registered for posting as a publication — Category B.

The Trials and Tribulations of an On-Line Computer Project

A CASE STUDY OF AN AUTOMATED LIBRARY LOAN SYSTEM

T. Alex Reid,
Assistant Director, W.A. Regional Computing Centre, University of Western Australia.

Abstract: The paper starts with a brief description of the system and its chief features, from the hardware, software and user viewpoints. It then goes on to trace the history of the project, from conception to its present state (currently the system has been in production for two years, but has not yet had all features implemented.) Some of the areas examined include equipment selection, performance and maintenance, software design, writing and supervision, and user reaction and acceptance.

In looking at all the things "that went wrong", an attempt is made to draw lessons which may be of help to others contemplating an on-line system, and to show that on-line systems are quite different in kind from batch systems — more different, in fact, than batch systems are from manual ones.

A. INTRODUCTION

The aims of this paper are fourfold —

(a) to give a brief overview of the principal features of the system, principally from a user's viewpoint;

(b) to give a history of the development of the project, giving both the hardware and software pictures;

(c) to make a brief comment on the file designs chosen; and

(d) to draw some conclusions and lessons from the project.

It takes the form of a case study — a study of what has been done, what things went wrong and why. Some of the lessons are perhaps only applicable to the author or to those directly involved in the project. Nevertheless, there is an extraordinary reluctance on the part of computer professionals ever to admit that things went wrong, and it is hoped, that by being a little frank about the writer's own performance, it may be possible to draw out some lessons which will be applicable generally. Hopefully, most of these lessons will become self-evident as the saga unfolds.

B. DESCRIPTION OF SYSTEM FEATURES

1. System Configuration

The basic configuration chart (figure 1) shows the system divided into four logical divisions. On the extreme left is represented the major computer installed in the WA

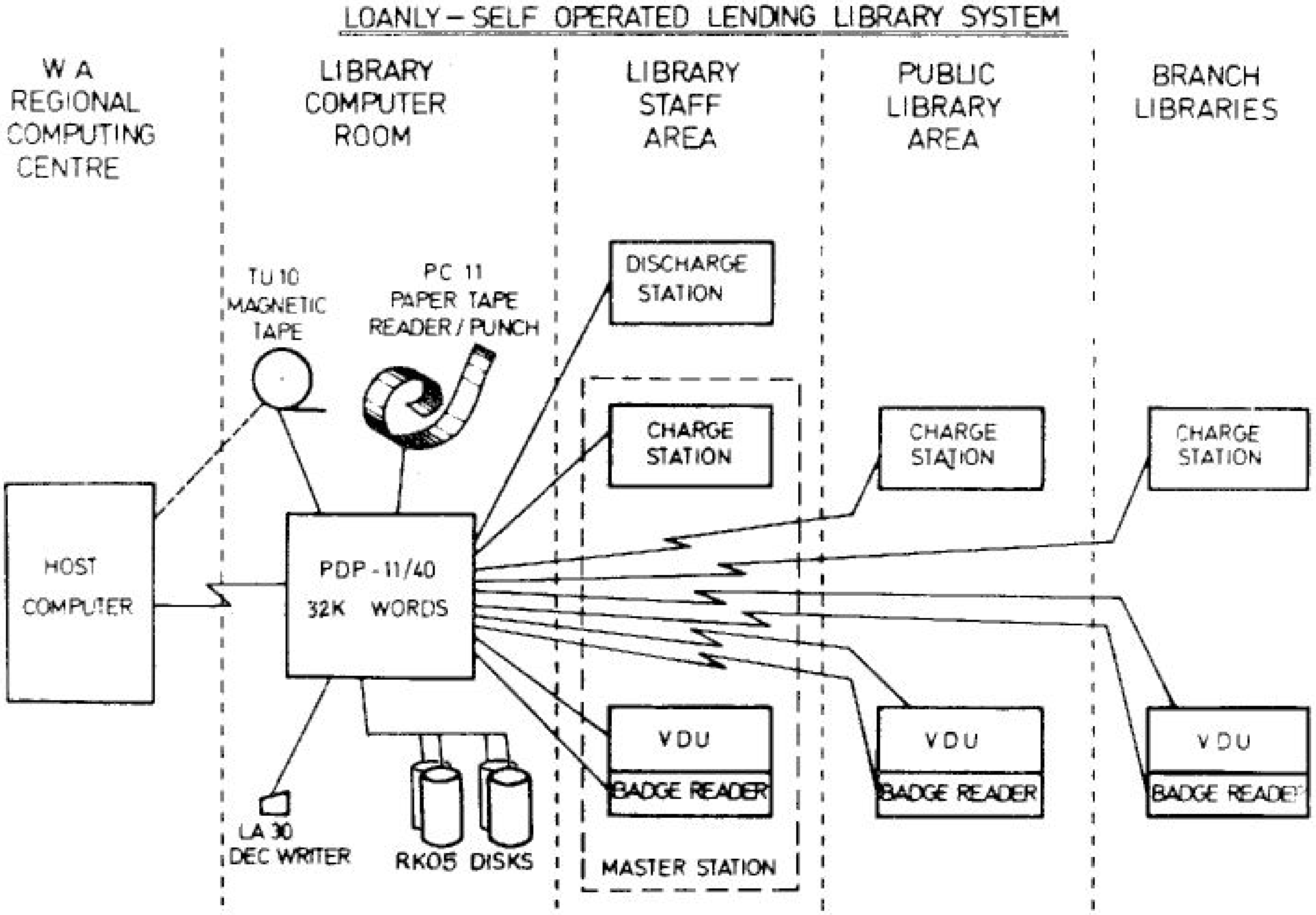

Figure 1 System configuration chart

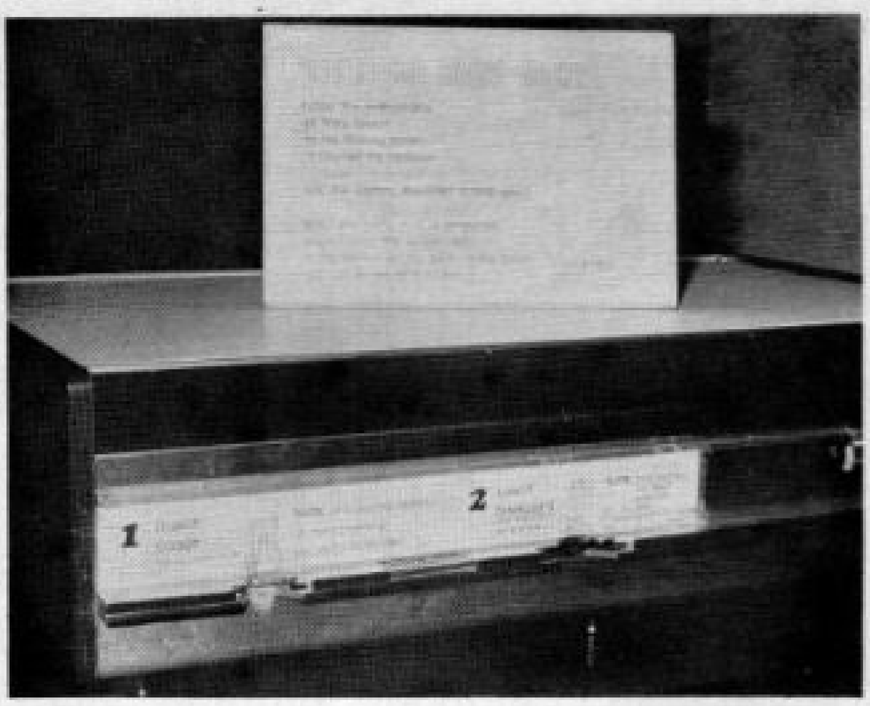

Figure 2 Charge terminal

Regional Computing Centre, to which the library system is linked. This link is not required for normal operation of the system, which is designed for free-standing operation. The next division represents the library computer and peripherals, which are located in the basement of the library. The next division represents the library staff facilities, which can be operated either as a separate charge station and enquiry station, or combined to form a "master station". At the next division are the public facilities, consisting of public charge stations and public enquiry stations. Finally, a stage not yet installed represents the facilities of a remote branch library.

2. Components of the Computer System

The heart of the configuration is a DEC PDP-11/40 mini-computer with 32K words of 16-bit memory, and a real-time clock. It has four RK05 cartridge disk drives, with 2.4 M bytes capacity each. These store the on-line files of the system, allowing for 18,000 borrower records and 24,000 loan records. A 150 1pm Tally line printer is used at present for logging transactions, each line representing one transaction on the system (enquiries are not included). The Decwriter console terminal is used for startup, error logging and other similar system functions. The system actually starts itself up in the morning and closes itself down at night, with the help of the real-time clock, and automatically performs various shutdown processing functions, the results of which are logged onto the Decwriter. Accordingly, the computer is left running 24 hours per day, 7 days per week. Connection of the eleven remote devices and the line to the DEC System-10 computer at WARCC is over asynchronous ASCII lines operated at 1200 or 2400 bps.

3. The User Terminals

(a) **Charge Stations**: There is a total of three charge stations in operation. These are used to record ("charge") loans (see figure 2). All are situated on or near the Loan Services Desk.

To effect a charge, it is necessary for the borrower to identify himself to the computer. This is done using a plastic 22-column hole-punched identification badge. The badge is pushed into a slot in front of the station as directed by an illuminated legend above the slot. The badge is clamped there, to prevent its removal until the end or interruption of the transaction is signalled, and another legend is illuminated to indicate the next step.

If the borrower may not use the terminal for any reason, this is detected by the computer and an appropriate legend illuminated. Otherwise, he must now identify the book he wishes to borrow. Book identification is contained on 80-column punched cards which are held in a pocket inside the back cover of the book. The information contained thereon includes the accession (identification) number, the Dewey class number, abbreviated author and title information, and other useful information such as publication date. Some cards are larger than the book, and rather than have them protrude beyond the top of the book, they are folded over. This can cause trouble if they are not unfolded before being fed into the terminal! All being well, the card is inserted into the terminal in the slot under the next legend to be illuminated.

Provided the card has been inserted successfully, a transaction slip ("charge ticket") is printed, guillotined and presented to the borrower under an appropriate legend through the third slot in the front of the terminal. The book card is ejected automatically at this stage, and the borrower instructed to return it to the pocket in the back of the book together with the ticket.

At this stage, a legend is illuminated which indicates that the borrower may insert another book card, if he wishes to make another loan, or may recover his badge to conclude the session. This latter is done by pressing the badge-release bar centrally located in the front of the terminal, which simultaneously signals to the computer that this has been done. This action, incidentally, may be taken at any stage of the transaction, should the borrower want to abort it.

Each ticket authorises the borrower to take a book in and out of the library at will – a unique requirement of the University and College libraries – and is a key component of the system. They come in several forms, depending on the nature and length of the loan. Normal loans do not have to be returned until the end of the current term at the University of WA (unless recalled earlier). Special loans are given extra prominence by blue and/or red colour stripes down the side of the ticket.

Of the three charge stations, one is for library staff use, and two are for borrower use.

(b) **Discharge Station**: The single discharge station is for library staff use only, and can double as an extra charge station when necessary. It is made into a discharge station by type-in at the master station, and turning a key on the front of the station. Alternatively, any station can perform temporarily as a discharge station when a special badge is used.

The tickets produced as a result of discharges indicate the following conditions:–

- i normal discharge;
- ii book had been recalled;
- iii book was overdue;
- iv book was not on loan;
- v a combination of ii and iii above.

For an overdue loan, the ticket is used as a basis for any fines which may be imposed. These are not calculated or recorded automatically by the computer. For a recalled book, the slip is held in the book, and the book shelved in a holding area until the recaller comes to collect it. He is notified of its availability automatically as a result of the discharge.

(c) **Enquiry Stations**: The enquiry stations are comprised of a badge reader and VDU (see figure 3 on front cover). The badge reader component is used for identifying the enquirer in the same way as is done at the charge stations. Again, a badge-release bar is used to retrieve the badge and to signal to the computer that the session is over.

The VDU is de-activated until an acceptable badge has been inserted.

On this occasion, it is possible to differentiate between various degrees of unacceptable enquirer; some may be barred all access to the enquiry station, while others may be permitted to enquire but not to effect a recall.

There are various options open to the enquirer upon identifying himself. Typically, he might select to enquire using class number as the key, and a screenful of loan descriptions is presented to him. He can opt to see the succeeding screenful, or to examine a particular loan.

Having examined the details of the loan, he can recall that book, revert to the screenful of books, or end that particular enquiry.

Any recall he may have requested will be initiated at that time automatically.

The session is terminated by pressing badge-release.

(d) **Master Station**: This station consists of a VDU and a charge terminal. These can be used either separately, or in a combined fashion. The VDU in its rest state offers all of the facilities available to the public enquirer, plus several others. One other aspect of the master VDU is that important system messages are displayed there, regardless of its status at the time (e.g. stolen badge in use at a named terminal).

One of the additional enquiries available at the master VDU is enquiry on a borrower. Having examined a borrower's status and loan and recall quantities, the library staff member has the option to alter his borrowing status, or record outstanding fines against him, etc.

The other principal use to which the master VDU is put is for examining system performance statistics and controlling the status of the terminals.

Any terminal can be made inoperative or changed from being, say, a charge station to being a discharge station. Any VDU can be made into an additional master VDU. In fact, should the need arise, a public enquiry station can be made into a master VDU by turning a key in the badge reader.

The charge terminal can be used, of course, for any charging or discharging operation. In addition, combined with the VDU, it can be used to charge books for which no badge is available. In this case, the borrower number is typed on the VDU.

(e) **DEC-10 Computer Link:** This inter-computer link (1200 bps) is used to enable a master VDU to be used as a DEC-10 terminal, or for the transfer of files to and from the DEC-10. At present borrower names and addresses are not held on the library system. Accordingly, the file of recall, availability and overdue notices which is accumulated during the day, is transferred to the DEC-10 daily for printing, and the names and addresses are edited on-line through the link.

(f) **Magnetic Tape**: The magtape is used for large-scale file transfer, for transaction logging (in addition to the line printer), and for regular file dumps for backup purposes.

All functions associated with the control of circulation are thus automated, and as long as everything goes well, the library staff and the users of the system are well satisfied.

4. The Use of the Logging File

The logging file, written to magnetic tape or disk, as well as to the line printer, serves many useful purposes, apart from the obvious one of providing a measure of protection against disk file calamities. The hard-copy log provides a continuous read-out of transactions in the system, which can be used to check just what has been going on, at the time it has occurred. This has had a most important impact on the degree to which the system has been accepted by the library staff. This was particularly useful in the early stages, though not so important now.

The most significant byproduct, however, of the log (more especially because it is available in machine-readable form) has been to provide a great variety of management information. For example, just from the hard-copy log, one can obtain a chart of borrowing activity throughout the day. Figure 4 shows loans made per half-hour period during a typical day, which reveals a number of interesting and useful facts about borrowing habits.

From a computer analysis of the loan file, one can obtain information on the distribution of loans among the institutions making use of the Reid Library (figure 5), the distribution of loans per borrower (figure 6), and the distribution of recalls per borrower (figure 7).

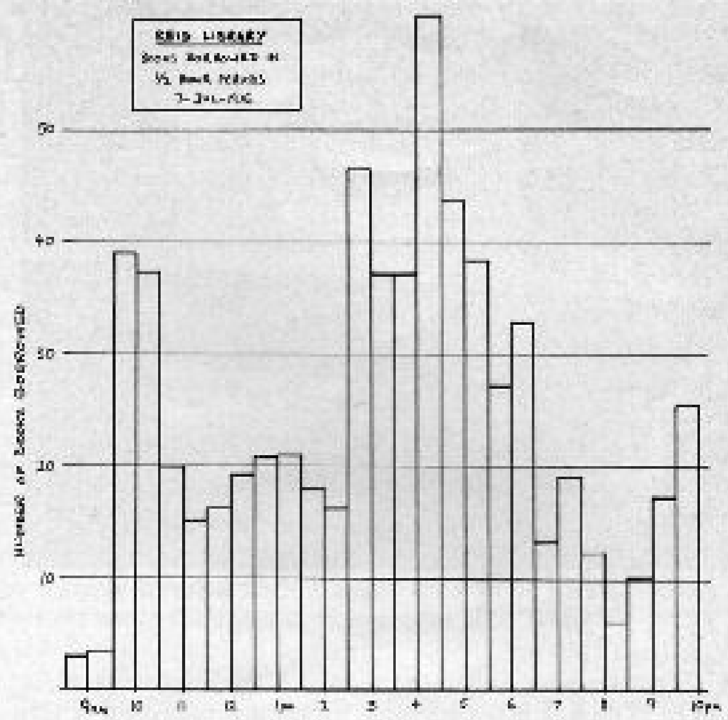

Figure 4 Loans by period of day

Indeed, there is no limit to the information one can extract, given the data in machine-readable form in this way, and suitable analysis programs.

Further information on the system features as originally planned can be found elsewhere[1,2].

C. HISTORY OF THE PROJECT AND SOFTWARE DEVELOPMENT

Figure 8 gives an idea of the key events in the development of the project as a whole, and the software development in particular.

As early as 1967 some form of automation was first considered for the library, and it was decided to move in stages, introducing a semi-automated punched card system in 1968. In this system, the punched card was held in the book until a loan was to be made, when it was removed and held by the library as a record of the transaction. The borrower wrote the date and his number onto the card, as well as onto a "transaction card" which was stamped by a library assistant and kept with the book as the receipt.

In July 1971 the Regional Centre first became involved, when it was invited to comment on a preliminary proposal received from IBM. It became clear that more work would have to be done by the University to devise a

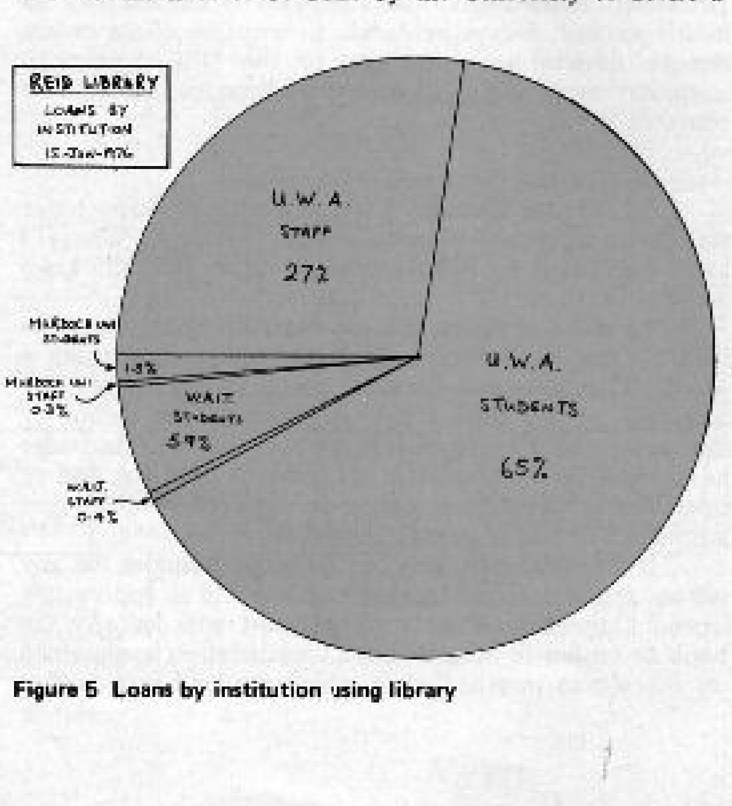

Figure 5 Loans by institution using library

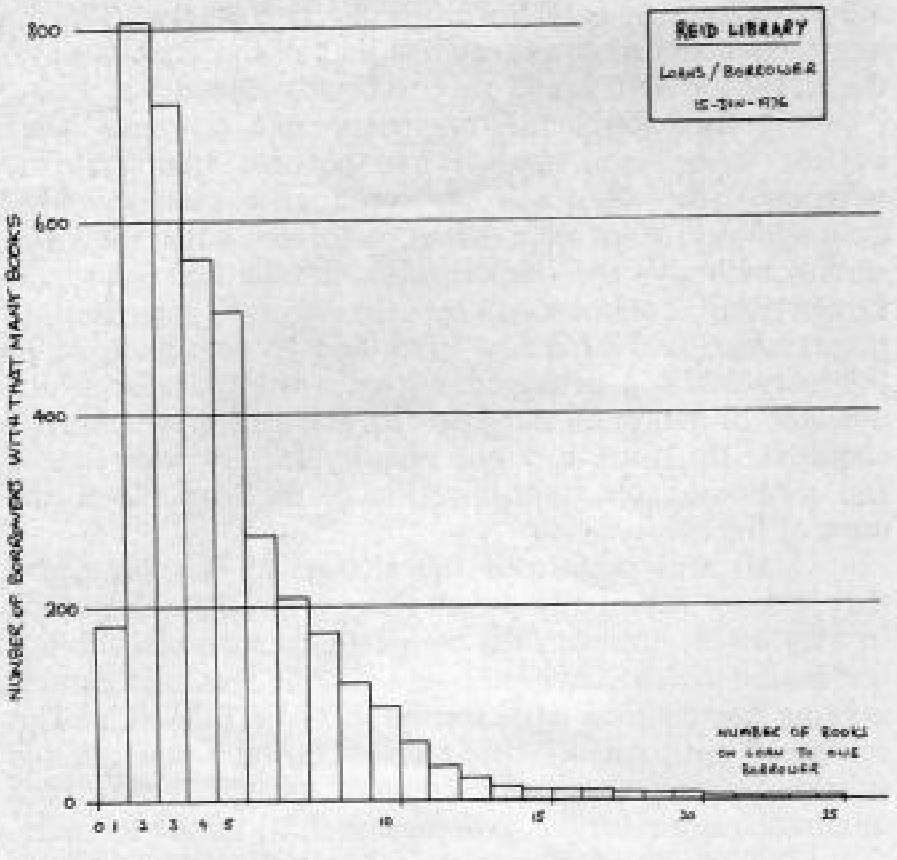

Figure 6 Loans per borrower

workable system, and the writer was given the job in October 1971 of advising the library on hardware and software selection.

The chief problem at this stage was in finding a suitable charge terminal which could be operated by the borrowers themselves. The library had become wedded to the principle of "self-charging" because it was here that the most dramatic staff savings could be effected.

No suitable terminal could be found. However, a firm (Automated Library Systems – ALS) was discovered by the Systems Librarian in a visit to England early in 1972. ALS was willing and, it seemed, able to design and build such a terminal. So it looked like a viable system could be put together. Consequently, specifications were drawn up and tenders called in August 1972. A combined DEC/ALS proposal was chosen and funding approved by the Australian Universities Commission.

It was agreed that the Computing Centre would write the software as part of the contract, and take about one man-year to do it. It was decided that a programmer should be recruited specifically for the job, on a temporary basis, and he duly arrived in July 1973.

At this stage, preliminary design only of the software had been done. A decision was quickly reached to use Fortran for the system because:

(a) a good compiler was available on the PDP-11;
(b) development could be carried out on any other large computer equipped with Fortran (specifically, the Centre's machines);
(c) the experience of the programmer and other Centre staff in Fortran;
(d) the desire for portability should the system ever be marketed.

This has definitely proved to be the correct decision so far as the Library is concerned. However, another decision made early in the piece, to use macro facilities, proved to be not so wise. The potential of macro processors is known and respected widely among academics but to date has gained little acceptance elsewhere. Essentially, it is very similar to the COBOL COPY statement, but with extensive nesting, recursion, conditional and parameterisation capabilities. Such facilities will be well-known to assembly language programmers where a macro processor is generally an integral part of the assembler.

Used selectively and wisely, such facilities can be most beneficial. They can provide a high degree of flexibility in one's programs, enabling changes to parts of the programs involving record layouts, buffer sizes, characters per word, etc., to be made by altering single parameter settings. This increases portability and adaptability. Very extensive use

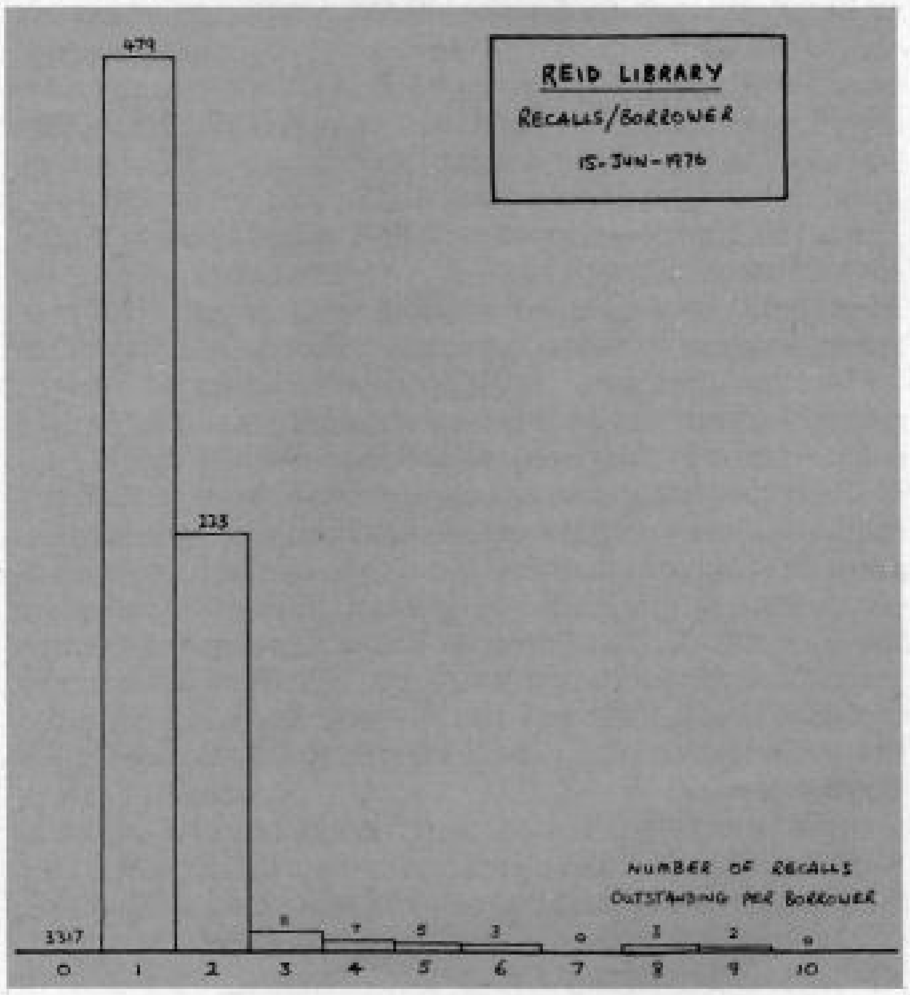

Figure 7 Recalls per borrower

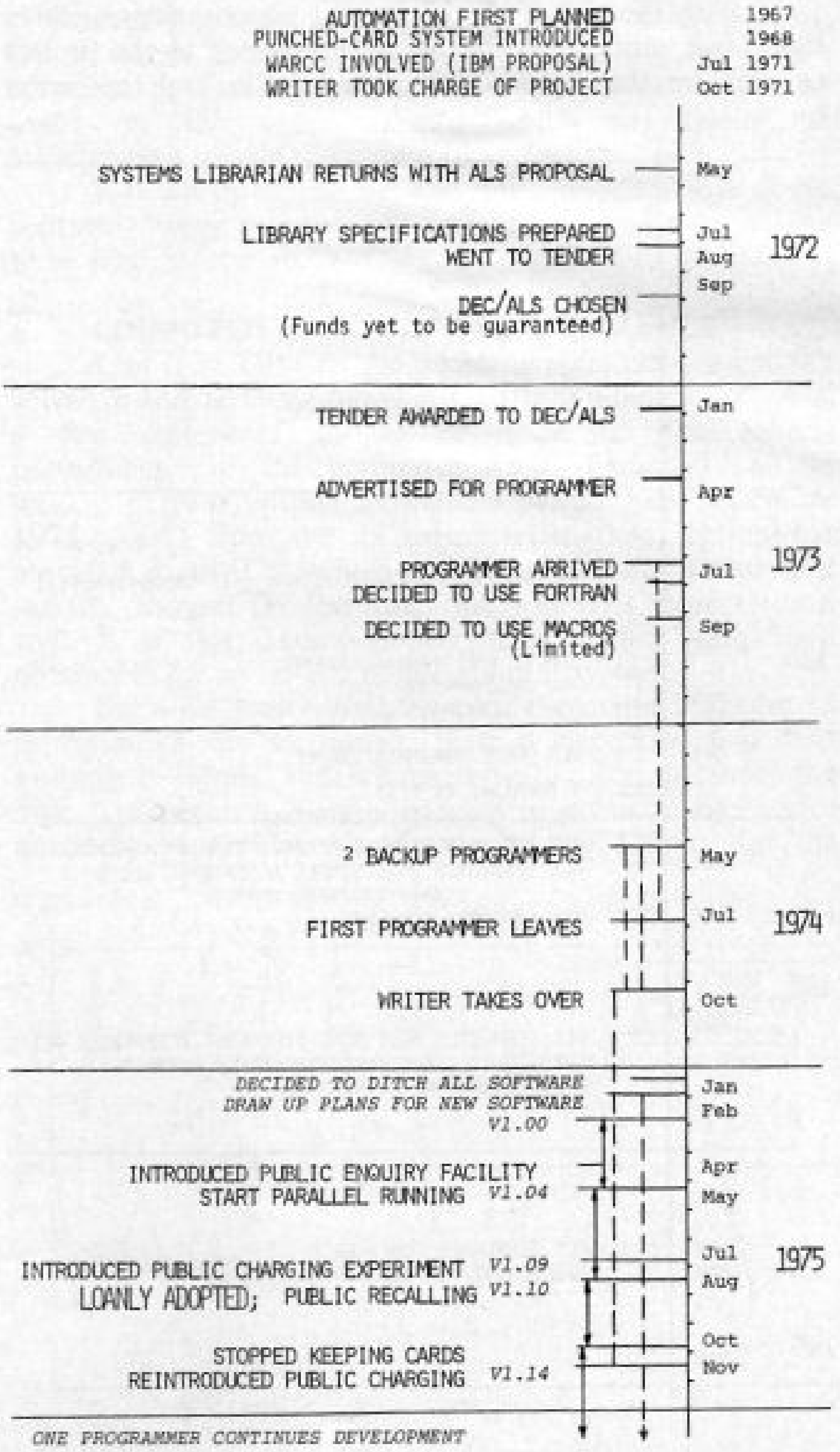

Figure 8 History of the project – general and software

was made of these facilities; so much so, in fact, that at one point it was claimed that we were close to a system in which a single parameter would determine whether a real-time library system, a real-time airline reservation system or a real-time betting system would emerge!

However, in this case a separate processor was needed (STAGE2 was used[3]) and apart from the problem of portability of such a system between programmers, macro processors are not known for their frugal use of computer time. Soon 24-hour turnaround was the best that could be achieved just to compile the programs written. This had a disastrous effect on the productivity of the programmer. The writer exercised inadequate supervision, and wrongly allowed the situation to continue until March 1974, by which time it had become obvious that almost no progress was being made.

To add to the difficulties, we had opted for a "grand-slam" approach, by which no part of the system was to be installed and run until the whole system was ready. This is in contrast to the more traditional "incremental-installation" approach, in which the system is first installed in a primitive fashion, and then gradually enhanced through later releases. The attraction of the grand-slam approach is that the user is not inconvenienced by a constantly changing system, and the problems of tinkering with a production system are removed. However, it unfortunately takes little account of the psychology of programming. On long projects, programmers are motivated by small successes along the way. Deprived of seeing the fruit of their labours on a progressive basis, they can lose a lot of their enthusiasm and drive.

With the end of the 12 months temporary position in sight, two more programmers were assigned to the project to pick up the pieces. This proved futile, and the writer moved in himself in October 1974. By January 1975 the inevitable conclusion was reached that all the work put into the existing system would have to be abandoned.

At this stage, the computer and terminals were waiting to be used, and it had become vital that the terminals be used in a productive environment. Consequently, plans were drawn up for new software which would embody the incremental installation approach. Design was commenced by the writer and another programmer on 29 January 1975, and 16 days later, on 14 February 1975, a debugged system was introduced which was able to handle all standard charges, discharges and staff enquiries. By April, a public enquiry facility was released and made available on a selection of the loan file to the users of the library.

Staff still performed the charges at this stage, and they did not necessarily put all charges through the system. In May 1975, however, the complete current loan file was transferred to the computer in a 5-hour run, and parallel running commenced with version 1.04 of LOANLY. This continued until August 1975, when LOANLY was adopted as the primary system. A public recall facility was simultaneously released, with version 1.10 of the software.

The Library continued to retain the book cards, as a safeguard against loss of their computer file, until they were satisfied as to the system's integrity, in October 1975.

A month later, in November 1975, with version 1.14 of LOANLY, public charging was adopted. A brief experiment had been undertaken in July, but adoption of this as the standard mode of operation was deferred for hardware considerations. These are covered in the next section.

Development of the software has continued since then, with just one programmer working on the system, and we are now with version 1.18. Regular bulletins are issued by the library as to the stage of implementation reached[4].

Further information on the method of implementation, at least as originally planned, can be found elsewhere[1,2].

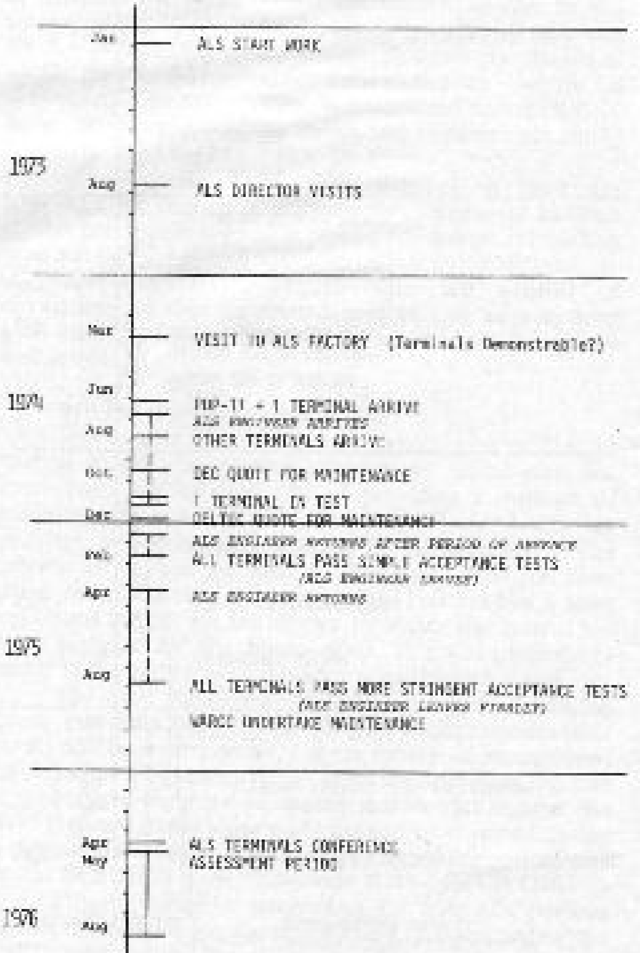

Figure 9 History of the project — hardware

D. HISTORY OF TERMINAL HARDWARE DEVELOPMENT

Figure 9 depicts the terminal hardware story, and can be laid alongside the project/software chart to see the interaction between the two.

Automated Library Systems of UK were the only company who were prepared and sufficiently experienced in library systems to design and construct terminals to suit the Library's needs. There were plenty of devices which would read identity badges or book cards or both, but none which provided a suitable ticket as a receipt. When they were awarded the contract in January 1973, they immediately commenced work on the detailed design.

The Managing Director of ALS visited Perth to attend the Library Conference in August 1973, and the opportunity was taken to establish what progress had been made and clarify some aspects of the design. By March 1974, the units were sufficiently well advanced for the writer to visit the factory and establish that they would indeed perform the functions specified. With hindsight, it is clear that greater benefit would have been reaped by sending someone thoroughly familiar with the design, construction and maintenance of terminal hardware, as these became the areas of greatest difficulty, and were largely overlooked at the time. The writer assumed that if a piece of equipment performed the functions it had been designed to do, then that was all there was to it. So it did not occur to ask what would happen if the terminal broke down.

In late June 1974 the first terminal arrived in Perth, together with the PDP-11 processor. Shortly thereafter, the ALS engineer who had designed and supervised the building of the terminals arrived. Other terminals arrived in August, but were badly damaged in transit. Because they had been specially built for the Library, replacement terminals could

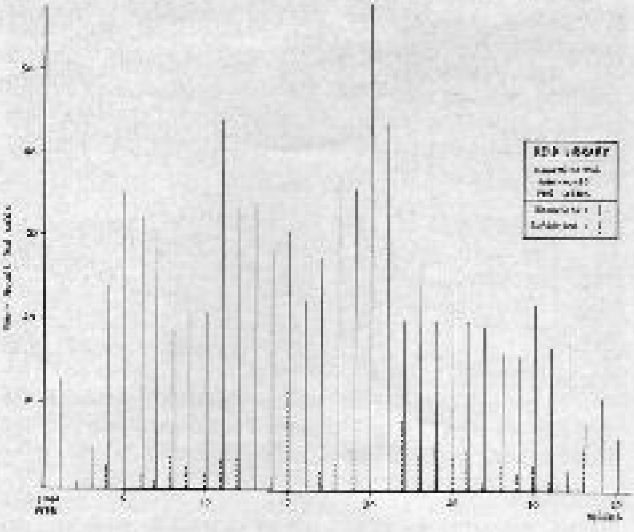

Figure 10 Terminal maintenance effort

not be ordered from the factory, and so the damaged ones had to be rebuilt in Perth (without the benefit of the ALS factory facilities). As a result, and because of modifications insisted on after their arrival, we ended up with four prototype terminals, rather than one prototype and three production models. This accounts for the long time that the ALS engineer spent on the site.

Meanwhile, we were anxious to obtain a maintenance service in Perth. A couple of firms quoted, but because the terminals were very much an unknown quantity, the best we could obtain was a "time and materials" quotation, at about $25 per hour. In view of the state of the terminals at that stage, this was quite unacceptable. Eventually, the Centre undertook to shoulder the risk, more of which later.

You will see that panic had started to set in with the software at this stage. Prior to this, there had been no real urgency in getting the software working, as there were no terminals to use with it. Now, however, it became important to be able to test the terminals under conditions approaching reality, hence the decision to adopt the incremental installation approach.

By February 1975, all the terminals passed some very simple acceptance tests, which did not reflect the situation which might occur under heavy usage. The verdict was "all terminals are working", and we allowed the engineer to return to England.

Unfortunately, all four charge stations became inoperable the very next day. Continuous attention by the Centre's engineers enabled us to limp along, but it became clear that much had to be done to the terminals before we could say they were really "maintainable". Consequently, ALS were petitioned to send an engineer back, and he returned in April. This time, far more stringent acceptance tests were laid down (under the guidance of the Centre's engineers) and agreed to by ALS.

The terminals duly passed these tests and the engineer left again. However, this did not signal the end to the Centre engineers' worries, as the terminals continued to need almost the full-time services of one technician. The difficulties experienced were particularly bad when the University resumed classes in March 1976, when the terminals received their first exposure to inexperienced operators in large numbers.

This culminated in a conference in April to decide what to do. It was decided to tackle the problem on several fronts including:

- (a) releasing one charge terminal every afternoon as a backup or for bench work;
- (b) carry out extensive progressive overhaul of each charge terminal;
- (c) eliminate on-the-spot repairs as far as possible;
- (d) improve the construction of the charge terminals in several respects;
- (e) improve documentation.

As a result of these steps there was a dramatic improvement in the performance of the terminals, as will be seen from figure 10; this shows the supervisor and technician hours spent on the system on a weekly basis from January to September 1976. Since May, there was a steady decline in the hours spent in reworking the terminals (bench time) and in correcting faults (restore service). Figure 11 shows the decline in the number of calls per week to repair the terminals.

This improvement finally succeeded in turning the tide of pessimism and defeat among Library staff evident early in the piece.

In order to give an idea of some of the problems facing the maintenance engineers (which have all been successfully overcome), it is instructive to note the contents of each of the charge stations. In each are the badge reader, card reader, ticket printer and guillotine assemblies, legend connectors, logic boards and no less than 5 different power supplies. Bear in mind also that each terminal is, in effect, a prototype, so the quality of the workmanship was less than perfect, and minor variations existed between terminals.

On the other hand, one must also bear in mind that no similar terminals had been built anywhere in the world, and almost no experience available on the likely effect of self-charging. Some of the problems which arose we could possibly have foreseen. For instance, the folding cards mentioned earlier have been periodically inserted into the reader in a folded state, and get jammed there. Releasing them requires the services of a technician, who runs the risk of damaging connectors on the card readers in the process. Other things, however, could not reasonably have been foreseen. For example, there is a critical adjustment to be made on the badge readers which has taken the maintenance people some time to perfect.

It is worth reiterating, also, the research nature of the venture — when you are first, you cannot expect everything to be straightforward.

E. COMPUTER HARDWARE PERFORMANCE

Lest it be felt that the problems were confined to the software and terminal hardware, both of which were novel, a few comments should be made on the general performance of the computer itself. Figure 12 shows weekly percent uptime during the period June-November 1975. Apart from the six calamities marked, uptime has averaged about 97%, which is fairly typical for a computer system, though far less than the Library's expectations, and, it is felt, rather poorer than should have been obtainable for an on-line mini-computer system.

Once the reasons behind even the major breakdowns in figure 12 are examined, then it becomes clear that nothing is simple and straightforward. For instance, the disk problems were possibly caused by poor airconditioning. The site chosen by the Library for the

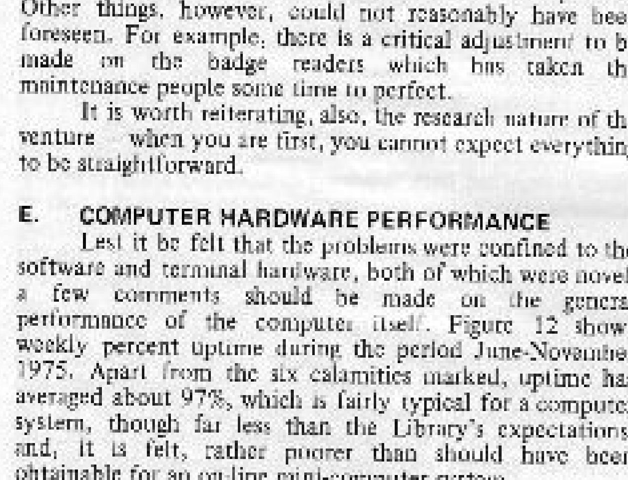

Figure 11 ALS terminal maintenance calls

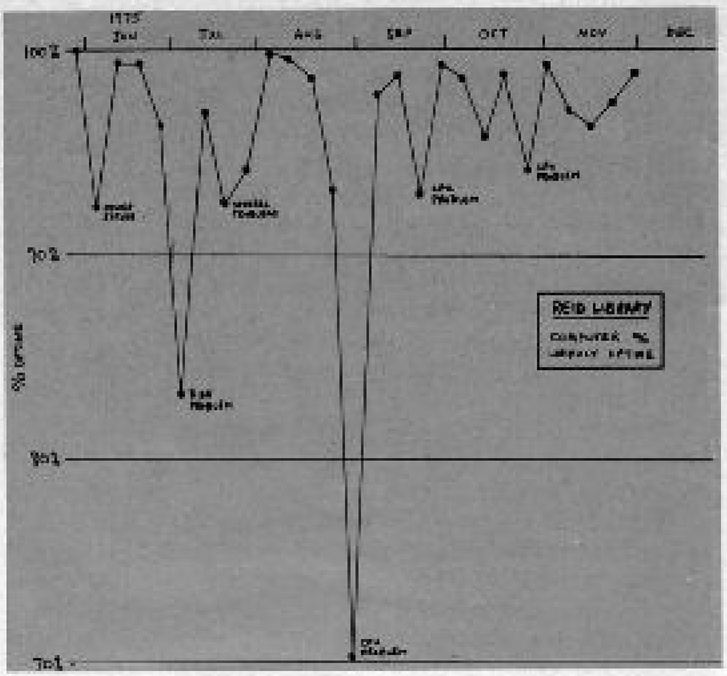

Figure 12 Weekly computer uptime

computer was in a "flat spot" in relation to the main airconditioning system; it was right next to the main return-air grille for that level of the Library. The poorness of the conditioning became obvious at an early date, and a separate unit was installed adjacent to the computer. Even that has not eradicated all problems – the conditioner has to work twice as hard, being near the return-air grille, and extracts copious quantities of water from the air. Because it is not close to an outside wall, this water is drained into a large copper tank from which the water is frequently bucketted out. The computer is effectively located in a passageway, adding to the airconditioning problems.

Examining the CPU problems reveals that the most likely cause has been a poor power supply (and, possibly, undue susceptibility to power fluctuations on the part of the CPU). Various parts of the computer expect differing voltage levels – 220, 230 and 240 volts compared with the nominal supply level of 250 volts in Perth. The Perth power supply is notoriously unstable; for example, on one occasion the supply reached 360 volts for a few milliseconds. Fortunately, earlier problems had led the Library to install a power regulator, which incidentally was burnt out on this occasion. Unfortunately, one of the computer cabinets was connected to a standard outlet, and extensive damage was caused. The computer supplier has since, but not because of the above, modified the disk power level protection mechanism.

The fact that equipment designed to run at 220 volts was being delivered 250 volts also contributes to the heat generated, and compounded the above heat dissipation problems.

An early discovery was the inadequacy of a standard "9 to 5" maintenance agreement for an on-line system. A reluctance to release the computer during business hours for preventive maintenance led to its complete neglect. It is a common problem with minicomputer suppliers, certainly in Perth, that an out-of-hours maintenance service is hard to obtain. Apart from the cost, which is designed to discourage, in some cases it is entirely dependent on the good-will of the engineer as to whether out-of-hours service is even possible.

Another problem, produced by this reluctance to release the system during the day, was that what preventive maintenance sessions were agreed to were frequently consumed performing tasks of a corrective nature which were not critical.

Finally, however, an agreement was reached whereby preventive maintenance was performed on a fortnightly basis on Wednesday mornings, with the alternate Wednesday available for any postponeable corrective maintenance.

An interesting, but not very relevant, episode followed this agreement. On the very first occasion of this specially scheduled preventive maintenance, a spanner was dropped into the works (almost literally) and the computer was down for a day and a half!

F. COMMENTS ON THE FILE DESIGN

The proposed file designs (see elsewhere[1,2]) were employed pretty well as described. That is, use of hashed-key addressing as the primary access method for the two main files (borrowers and loans), use of indexes for secondary accessing (e.g. enquiry on class number), and extensive use of chaining.

Although known to be the least efficient of the methods for handling collisions in a hash-addressed file, the linear probing method was chosen. The reasons were:

(a) its programming simplicity;
(b) it was a "known technique", and therefore could be implemented in a hurry (refer to the final implementation timetable);
(c) its stated performance, measured by average accesses for files of certain loadings, was adequate for the needs of the system at the time.

Unfortunately, what was not published in the literature[5] (and what to date has still been overlooked[6]), was what happened to the performance of this technique under a high insertion and deletion rate. What happens is that the average and maximum access times for an insertion steadily increase until they become equal to the time required to read the whole file. It didn't get as bad as that before the effect was noticed! A number of attempts were made to rectify the situation, until finally the use of hashing was abandoned in favour of a crude but effective index scheme, pending further investigations.

Needless to say, subsequent attempts to alter the file accessing methods have been preceded by healthy simulations. The log files (B.4) have been put to good use here also.

More information on the use of linear probing (in particular) for a volatile file is now being assembled for publication[7].

G. CONCLUSIONS AND LESSONS

Before eliciting some general lessons from this project, a few notes on the general success or otherwise of the project are worth making.

Firstly, the project as a whole can be considered a "success" in that it has indeed saved money, facilitated the borrowing of books, and provided additional information on borrowings. Despite some early apprehension, the system has gained thorough acceptance by the library staff and by the borrowers. The bulk of the borrowers find the enquiry facility particularly useful and easy to use, and generally prefer to use the automated system (attractive library assistants notwithstanding!). There are a few people, of course, for whom the system offers little advantage – these being mainly those who use the library very infrequently and so are never familiar enough with the terminals to use them with confidence.

The people using the terminals are essentially untrained, and this represents a quite unique venture; however, surprisingly few problems have been encountered, in spite of difficulties with the terminals, and enquiry dialogue which is far from perfect. The best indication of borrower acceptance is perhaps given by the 40% growth in borrowing that has occurred over the period 1975-1976, despite a static borrower population.

Not surprisingly, the Library is now planning the replacement of the charge terminals. It is expected that these will be taken out of service at the end of 1979. By that time, they will have seen 5 years of service, which is not too disgraceful for electromechanical devices in the hands of the public. The Library's confidence in Automated Library Systems Ltd is still sufficiently great to be looking to that firm to provide replacement terminals, which will almost certainly be their label-based equipment. These devices have no moving parts and do not even require opening of the book to have its identifying number read by the computer. The punched cards will be eliminated – a decision which, but for their existence in every book, and the lack of a suitable alternative at the time, should perhaps have been made at the outset of this project. Tickets will also be eliminated, which move is expected to substantially speed up the process of borrowing a book. The present purpose of a ticket is to validate the loan at the exit gate, and this function will be taken over by a book-reader at the exit gate, operated by the attendant there – so the computer itself will check that the book is on loan, and hopefully in less time than it presently takes to open the book and examine the ticket.

In this new system, greater consideration will need to be given to fall-back in the event of computer failure. Presently, it is a simple matter to revert to the previous manual system, in which the book-cards are retained by the library as its record, together with handwritten borrower identification, and a manually stamped receipt card issued to the borrower.

Overall then, the system can be said to have been successful, though far from perfect. And many of these imperfections, it is hoped, will be ironed out in the above improvements.

We will turn now to some of the conclusions and lessons which can be drawn from this project, many of which should have become obvious as the tale unfolded.

1. The specifications for this system developed as the project did. This has dragged out the project unacceptably. Nevertheless, in any project where new ground is being broken, it is to be expected that some decisions will be reversed as we see the result. In addition, it is questionable, for a long project, whether the specifications can really be frozen. A library is a service organisation, and must respond to the needs of its users; all improvement or change cannot be halted just because a computer project is under development.
2. Point 1 above begs the question as to whether the specifications were adequate to start with – whether the systems analysis had been done effectively. For any on-line system, because the user is so much closer to and dependent on it than in a batch system, it is vital that the systems analysis be done thoroughly.
3. Developing a system while it is in production has several serious disadvantages, especially when it is an on-line system. Most of these will be patently obvious, and require very careful procedures for implementing new systems (figure 13). The effect on programmer morale of having access to the computer only after 10 pm or at weekends can be rather severe!
4. Should systems be implemented all in one go ("grand slam") or bit by bit ("incremental")? Two contributions can be made to this debate: firstly, with the latter, the user might have to be forced to accept, for a time, something less than the facilities he had under his old system; secondly, programmer morale may need some external stimulus under the former. Other relevant considerations were raised under 1 and 3 above.
5. The matter of programmer morale raised in 3 above points to the importance of having access to some testing facility other than the computer being used for the on-line system. We had access to a DEC System 10, for which was built a simple but effective testing vehicle ("testbed"). Such a testbed, if set up for use in batch as well (as was this one), makes it possible to put each new version through a standard test package of transactions before being installed, with results checked automatically against the last successful run.
6. Over-ambitious programmers and negligible supervision make poor bedfellows. It is clear that a reasonable level of supervision is essential, even for professionals.
7. It is clear that macro processors make inappropriate developmental tools, because of lack of widespread knowledge of them, and because of high processing costs.
8. A computer project consisting of one

PRINCIPLES TO OBSERVE WHEN DEVELOPING A PRODUCTION SYSTEM

(1) Never introduce a new version without extensive testing in the off-line mode (the only exception to this is when something as trivial as a change of due-date is made – but see (2) below).
Note: testing should include all sorts of general operations which are not affected by the changes.

(2) Always run a few tests on the new version once installed live; these should include a few normal charges, discharges, enquiries, plus a few random operations relevant to the area modified.

(3) Always failsafe the disks before introducing a new version which either
(a) modifies aspects of accessing the disks, or data recorded on disk; or
(b) makes significant other changes.

(4) Every new version, no matter how trivial, must have a new version number:
(a) if the new version has no external (visible to user) changes, then only the version suffix (alphabetical) should be incremented;
(b) if there are external changes, then the visible version number (numeric) should be incremented (and the suffix blanked out).

(5) For a type 4(b) change, multiple copies of a full written description of the differences should be given to the user a few days before (for changes involving significant alteration to the user's procedures or operation, considerable advance notice – even if not so detailed – should be given), along with a statement such as "changes to be made within the next few days".

(6) For all changes (types 4(a) and (b)), warning should be given the day before any attempt to introduce the new version. Notification should then be left with all concerned stating that a new version has been installed, its version number, and a brief summary of changes made. (**Note:** For both types 4(a) and (b) changes).

(7) The previous version should be renamed to clearly indicate what version it was, and left on the disk; the new version should be so named that it will be used automatically.

(8) Written instructions should be prepared which indicate how to revert to the previous version should it prove necessary. This is particularly important where file incompatibilities may exist.

(9) Arrangements should be made prior to the change for at least one officer equipped at least with the information in (5)-(8) above, to be on stand-by at the time the new version is first used, and the user should be notified of who this is.

Figure 13 Steps to observe when changing a production system

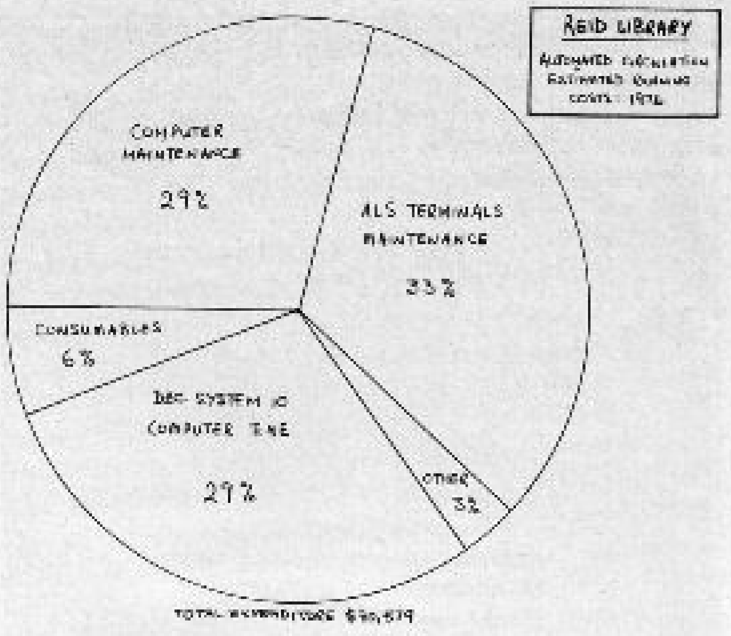

Figure 14 Circulation system running costs

analyst/programmer suffers the least from communication breakdowns, and ought to have the best chance of success (given the task is not too great). However, if just one programmer leaves, the project is thrown into chaos, particularly if non-standard or novel techniques have been employed. However, this experience has failed to shake the writer's faith in the superiority of one "superprogrammer" over a "Mongolian horde".

9. Where a project breaks new ground (has a research content), then all parties must be thoroughly prepared to back down on requirements and/or produce more resources when things go wrong.
10. Arising from point 9 is the observation that when there is only one supplier prepared to offer the goods, then you must be prepared either to compromise on your requirements, or to abandon your plans.
11. Both points 9 and 10 are compounded when the supplier of custom-built goods is 10,000 miles away. It is not feasible to continue a thorough dialogue over that distance as specifications are refined and progress monitored.
12. Because the project was to lead to a fully productive system, it was embarked on only after a cost justification had been done (before the Computing Centre was involved in the project). One might be forgiven for wondering if the project was ever properly cost-justified; certainly, the trade-offs were never kept before all parties, so that decisions on consuming extra resources or cutting losses were never possible. The writer personally solved this difficulty by assuming the project was largely a research one (encouragement to do so was received from both the Computing Centre and the Library). Research projects are not necessarily designed to make a profit, but to contribute to the discovery and dissemination of knowledge (which purpose is partly served by this paper).

    The main justification for the project was in the staff savings which could be achieved. However, one feels that the costs of running (as opposed to implementing) the system were badly underestimated. One tends to talk in terms of (for example) "a computer system costing $100,000 will save 5 people per year". The running costs are in fact quite high (figure 14) totalling over $30,000 per year at present.

    The actual staff savings have been variously estimated by library staff at from 2 to 7, though no official statement has been made. In fact, all of the affected staff are being used to augment other functions in the library. Computer-caused redundancy is, of course, a dirty word.
13. Finally a point, which has been highlighted in the above points, is the overwhelming complexity of on-line systems, from all aspects, compared with manual or conventional batch systems.

    The care required in specifying an on-line system, was mentioned (point 2); and the special provisions which must be made for testing on-line system releases was discussed (points 3 and 5). The problems of providing for hardware maintenance in an on-line system were covered (section E). Mention was made (section B.4.) of the value of some kind of hardcopy output; humans typically have had an attitude towards any kind of system set up to streamline their function (be it computer or manual), where some tangible documents are at the heart of the system, that "in the final analysis", they could go back to the documents and "work it out by hand", albeit somewhat tediously. This fall-back option, held in the back of the mind, has tremendous psychological benefit in providing people with a sense of security. On-line systems remove this hard-copy reference point, and can induce all sorts of insecurities unless introduced carefully.

    Another very important safety factor, which is often overlooked, is the ability, in a batch system, to "rerun the job" if it fails for some reason. Frequently, the user need never know about the failure. The user may, of course, develop a much too optimistic view of computer systems, which will not stand him in good stead when he is given an on-line system. There is no room for such errors here, because the transaction information is transient, and must be recorded correctly inside the system at the time of the transaction or it may be forever lost. Many more transactions may occur before the problem is noticed or can be fixed, so a complex back-track may be necessary (a long way from the simple father-son and transaction tape files). And the user will not only know about the failure, he will know about it a long time before the computer person.

    An on-line system thus requires a much greater dependence on the computer than has heretofore been required, and can lead to all sorts of new situations.

    One failing that many computer people have, and which is worsened by an on-line system, is to tend to believe that the system is "the computer", or at best "the computer system". Humans are still the most important ingredient, and more so with an on-line system where the system and the human are wedded more closely. Computer people are fond of saying that the computer is "just another tool" and while the computer *is* certainly just a tool, it is also important to recognise that there is such a difference in magnitude between it and any other tool as to constitute a difference in kind. On-line systems underscore this difference, and should only be attempted after very careful analysis and planning.

H. REFERENCES

1. NOEL, D.G. and REID, T.A. "System and File Design in the LOANLY Real-time Circulation System". *17th Library Assoc. Conf.*, August 1973, p479.
2. REID, T.A., NOEL, D.G. and GREAVES, C.P.R., "LOANLY – A Real-Time Library Circulation System". *6th Aust. Computer Conf.*, May 1974, p45.
3. WAITE, W.M., "The Mobile Programming System: STAGE2", *Comm. ACM*, 13, July 1970, p415.
4. "LOANLY Loan Control System Summaries", published periodically by D.G. Noel, Reid Library, University of WA.
5. MORRIS, R., "Scatter Storage Techniques", *Comm. ACM 11*, Jan. 1968, p38.
6. SEVERANCE, D and DUHNE, R., "A Practitioner's Guide to Addressing Algorithms", *Comm. ACM 19*, June 1976, p314.
7. REID, T.A., "A Cautionary Note on Randomised Files", in preparation.