ICL TECHNICAL JOURNAL

Volume 6 Issue 1 May 1988

Published by
INTERNATIONAL COMPUTERS LIMITED
at
OXFORD UNIVERSITY PRESS

ICL TECHNICAL JOURNAL

The ICL Technical Journal is published twice a year by International Computers Limited at Oxford University Press.

All correspondence and papers to be considered for publication should be addressed to the Editor.

The views expressed in the papers are those of the authors and do not necessarily represent ICL policy.

1987 subscription rates: annual subscription £32 UK, £40 rest of world, US $72 N. America; single issues £17 UK, £22 rest of world, US $38 N. America. Orders with remittances should be sent to the Journals Subscriptions Department, Oxford University Press, Walton Street, Oxford OX2 6DP, UK.

Printed by H Charlesworth & Co Ltd, Huddersfield ISSN 0142–1557

Contents

ICL TECHNICAL JOURNAL

Volume 6 Issue 1

Foreword

The first six papers in this issue all deal with various aspects of support and maintenance of computer hardware and software. This is a vitally important service and is one that does not always get the credit and attention that it deserves. In today's conditions computer service presents an exceedingly demanding task, both organisationally and intellectually. A modern computer, even what is now considered a small system, is really a very complex and sophisticated device, made even more so by the fact that it is often required to do very complicated things with the data given to it. Users very reasonably expect to get the very best out of their systems and any fault that does occur to be corrected very quickly. In many cases the continuous error-free working of the system is so vital to the organisation that it serves that anything but the highest standard of reliability is not tolerable.

ICL has many thousands of systems installed all round the world, of all sizes and used for all manner of purposes. To meet the users' needs for reliability it must have an organisation and the physical means to gather, analyse and classify information on the performance of these systems, on any malfunctionings and on how faults have been corrected. Also it must exploit all the techniques and technologies – and be always on the lookout for new ones – to improve the process of inferring the reason for a fault from whatever was observed. Modern equipment is really remarkably reliable, but the degree of interworking, particularly via networks, results in the reason for a failure being often not at all obvious: hence the need for skilled diagnosticians supported by powerful tools. The aim of these papers is to give an idea of the organisation that ICL has set up to meet these needs and of the tools that are being used.

J.M. Proctor
ICL Director of Services

ICL Series 39 Support Process

R. Allison
ICL Southern Regional Support Centre, Reading, Berkshire

Abstract

The ICL Series 39 mainframe computers were designed from the start to be maintained entirely by a remote support process. The paper describes this process and the ICL organisation that operates it. Key technological components are special hardware built into the machines, the Node Support Computer, that monitors all activities continuously and collects diagnostic information, and through its Watchdog facility can ensure that the VME operating system is functioning correctly; and special software permanently resident in each machine, SAM – Support And Maintenance – that, among other activities, processes and interprets this information. In "system dead" situations the NSC can be accessed by special software available to the service. This software, VISA – VME Inoperable System Access, runs in either a remote 2900 or Series 39 mainframe and enables investigation and diagnosis to take place. The support system can deal with both hardware and software faults.

1 Historical background

The first commercially produced computers, whoever the manufacturer, were maintained by a team of engineers who lived on the customer's site and were responsible for solving whatever problems arose and keeping the system running. They would carry a stock – often a large stock – of spares; they could call on specialist help, for example in connection with peripherals, from the manufacturer's relevant area base.

In the mid-1960s machines were becoming more complex and were being designed in compatible "families", such as the ICL 1900 range of which first deliveries were being made in 1965. Also, they were being installed in much greater numbers: when ICL was formed in 1968 by the merger of ICT and English Electric-Leo-Marconi it had an installed base of over 1000 1900-series mainframes. These machines provided more information to help diagnose faults, but because of their greater complexity the maintenance process had to be separated into a number of specialist activities. The on-site service, covering the system on a broad basis, became known as the "first line" support and dealt with the problems that arose most often – and which therefore the site engineers could deal with themselves; the more specialised services were provided by area or otherwise centralised "second line" support units.

The second line support staff had to decide if they could diagnose a fault, and give its cure, from the information given to them by the site engineers, or whether they needed to go to the site for more investigation; and if the latter, what spares, and possibly further test equipment, they needed to take with them. They became very skilled in this, but with the increasing size, complexity and installed numbers of the systems this procedure became increasingly costly and impractical. There was thus an increasingly strong drive to automate as much of the process as possible; in ICL this culminated in the features designed into the Series 39 machines and in the support organisation and process described in the rest of the paper.

2 Series 39 forerunner: The ICL 2900 and the ADEMS software

A great deal of self-monitoring and diagnosis was built into the machines of the 2900 range, both in the first P-series and its successor (and current) S-series; and a number of software products were developed to use the information provided as an aid to maintenance. This culminated in a system called ADEMS – Automated Diagnostic and Error Management System – which resided in the customer's machine. This could be used to provide very detailed information on the behaviour of the installation over short periods, or summarised information, for example in the form of bar charts, over long monitoring periods of up to 40 weeks. It included a system diary in which all load, dump and fault times were recorded automatically. The information was presented in a very easily readable form and the system could be, and was, used by the customer's staff as well as by the ICL engineers. Also, many customers installed modems to connect their systems to ICL support centres; and the latter, through a network of ME29s, could monitor the welfare of the systems thus linked and provide a remote support and maintenance service.

ADEMS was very well liked, was undoubtedly successful and gave valuable experience in remote support. But it was passive in the sense that it almost always left the support engineers to make the diagnosis, only in very few cases attempting to resolve a problem itself. Its successor SAM – Support And Maintenance – goes much further, as will be explained.

3 The ICL Series 39: SAM and VISA software

The ICL Series 39 family of mainframes was launched in 1985 with the first deliveries of the then smallest member, Level 30: the May 1985 issue of this journal (Vol. 4, No. 3) is devoted to this machine, which incorporates many very novel architectural and engineering features. The main processing unit in a Series 39 system is the Node, consisting of an Order Code Processor (OCP) which executes the basic machine instructions, a store of from 8 to 64 megabytes of random-access information, an Input/Output Coupler which, through other interfaces, communicates with disks, tapes, printers, other peripherals and communications links, and a Node Support Computer (NSC) described below. A Series 39 system can consist of from 1 to 4 nodes.

The NSC is described by Ashcroft in the journal issue referred to above as follows.

> The node support computer (NSC) has two main purposes, initial program load (IPL) and diagnostics. At switch-on the NSC, after self-test, loads the microcode for one of the High Speed Couplers (HSC) and then, through this coupler, establishes communication via a High Speed Disk Controller (HSDC) with a disk containing the remainder of the microcode. After loading the microcodes into the various units the NSC loads a 2900 IPL program into the main store, which when activated loads the full VME system. All the node registers incorporate loop-spinning registers which enable them to be read or written to by the NSC for maintenance and fault finding. A telephone line connection is provided for diagnosis by a remote VME system in a diagnostic centre, running the diagnostic software VISA (VME Inoperable System Access).

This description makes clear that the ability to do remote diagnosis, at a very detailed level, was designed into the fundamental structure of the series.

Corresponding to ADEMS for the 2900, the SAM software for Series 39 resides in the customer's machine. This:

- records and analyses all the usage and error information reported by the system
- checks the performance of each unit against pre-set thresholds and generates a Maintenance Service Request (MSR) on any unit that does not meet the threshold conditions
- compresses the information relevant to a MSR into a "fingerprint" of 255 bytes, establishes communication with the relevant ICL support centre, usually over the ordinary telephone system (the PSTN – Public Switched Network System), where the request is dealt with as described in Section 5 below
- receives, analyses and acts on the response to the MSR, if need be generating and sending a request for further action or information.

As will be explained, the handling of and responding to the MSR at the support centre is normally completely automatic, and the necessary corrective action is often given within a few minutes.

Figure 1 shows this linking, and what is located in the customer's machine and in the support centre respectively.

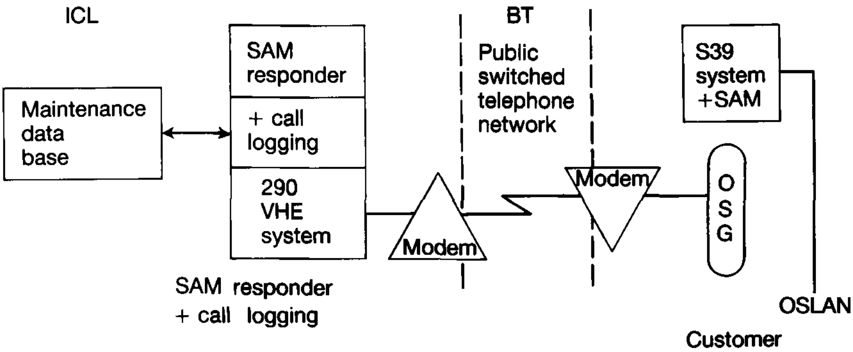


Fig. 1

This is the procedure when the system in the field has VME and SAM operational. If it is severely crippled or completely dead a different one has to be used, for which the VISA software, already mentioned, was written. This resides in the ICL centre and enables the diagnosticians there to access and investigate the field system over the telephone connection; Section 6 describes this.

4 Series 39 Support Organisation

Figure 2 shows the present structure: this has been developed over several years in response to changes in demand, and as experience was gained. The primary division is into the two Regional Support Centres, Northern (NRSC) and Southern (SRSC) serving customers in the corresponding halves of the UK. NRSC, which serves continental Europe also, is located at West Gorton (Manchester), SRSC is split physically into locations at Reading and Elstree but operates as a single unit. These centres act as both first and second line support, currently dealing with about 70% of the MSRs and all of the VISA calls. Linked to the RSCs are the Branches, which are in fact the

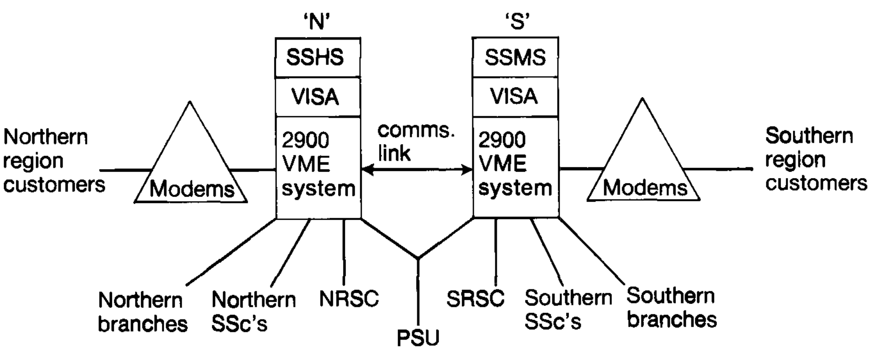


Fig. 2

customers' point of contact with the service. A Branch will provide some of the first line services, doing the initial checks on the MSR; it is also the interface during normal working hours for manual calls and provides on-site expertise to replace parts, this last usually under the direction of more specialised units in the organisation.

The RSCs, with their Branches, are backed up by System Support Centres (SSCs) and Product Support Units (PSUs), which provide more specialised services as required; the former specialise in communications and superstructure software products, the latter being responsible for any problems that the second line units cannot solve and acting in effect as third line support. There are SSCs at Bristol, Putney and Reading in the south and in Manchester (Arndale House) and Wakefield in the north; several PSUs are located at West Gorton, with additional specialised units at Stevenage and Bracknell. Any problem that a regional centre cannot solve is passed on to a PSU.

The processing power needed to operate the service is provided by a VME installation at West Gorton (2 × 2966 at the time of writing, but shortly to be changed to a Series 39 system), to which all centres and branches are networked.

The support process itself falls into two distinct operations:

(i) resolving the problems raised in the MSRs when the customer's system is basically operational
(ii) dealing with "system dead" situations, requiring use of the VISA software.

The second is clearly more serious and of course is given the highest priority; we describe this next, and then the first in Section 6.

5 "System dead" situation: VISA software

The VISA software resides in the mainframes at West Gorton dedicated to the support service. As already explained, it enables an engineer in a support centre to access and investigate a customer's system when it is effectively or actually inoperable. He does this through the Node Support Computer, and can read from and write to registers in the node. The VISA link is shown in Fig. 3. The NSC cannot interrogate the controllers on the fast optical-fibre Macrolan link to the node's disks and tapes, nor the slower peripherals linked by the slower 10 mb/sec Oslan; so the diagnostician has to be helped by someone on the site, to report displayed information that cannot be seen at the centre.

VISA is clearly a very powerful and sophisticated tool, and is for use by experts only; in fact it is used only in the second and third line services. It was designed mainly with hardware failures in mind, but it can be very effective in dealing with software problems also, when a customer has corrupted both

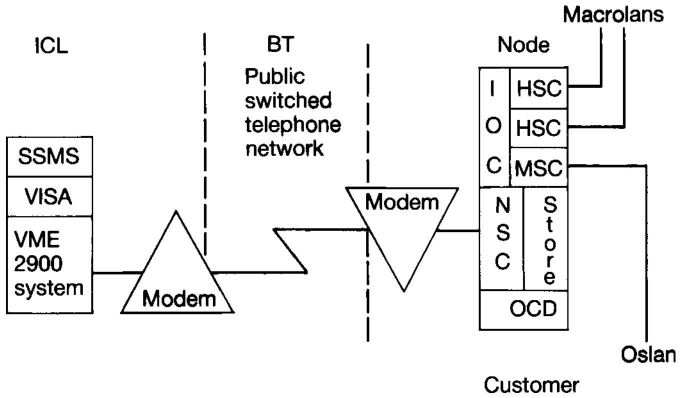


Fig. 3

the loadsets of his system and VME will not load on either; an example is given in Section 6.2.

A typical sequence of events in a VISA call is as follows; usually the system will be inoperable, so the initial contact has to be made by a manual telephone call – the mechanism for handling calls in the support centres is described in Section 6.1.1.

1 The customer cannot load his system. He telephones his Series 39 Branch, who log the details and establish if the problem is trivial, such as no power on a unit.
2 If the Branch cannot solve the problem the call is passed immediately to the Regional Support Centre who allocate a diagnostician.
3 The diagnostician calls the customer and discusses the problem.
4 If the diagnostician decides that a VISA connection is necessary he allocates a telephone number to the customer.
5 Each node has a switchable "VISA cable"; the customer must switch this to the position that connects the NSC to his modem – the other position makes the connections required for transmitting MSRs, so that one modem will cover both needs.
6 The customer calls the number given, and when the ICL modem answers switches his modem to DATA and his node to REMOTE.
7 The diagnostician checks that the connection is made and the link established. If it is, he can now load (IPL) the remote system and monitor the loading sequence. If the load succeeds the VME regime can be established and the system's progress monitored, all the node lights being displayed over the VISA link. Hardware or program counter stops can be set so that if the load fails the state of the node at the point of failure is preserved for investigation.
8 If the link is not established the first corrective action is to replace the single board that carries all the node's communications capabilities.

9 If the diagnosis is that there is a hardware fault and a new board is required, the diagnostician will inform the Service Desk who will arrange for an engineer to go to the site and fit the board.

It is worth while adding here that the NSC has extensive built-in self-checking features. There are back-to-back tests which check the logic up to the modem. Once the NSC is running it can pull in a general purpose diagnostic package DGEN, residing on the Initial Program Load disks, which checks all the hardware within the node. There are two versions of DGEN available: a reduced version which runs automatically when the system is powered-on and the full version that can be called from the node or via the VISA link, as required.

6 Dealing with maintenance service requests

Before discussing this we must say something about the resources used by the support service.

The key resource is the Maintenance Data Base (MDB). This is located in a mainframe at Hitchin and is used by the support units to handle hardware and software problems. Each time an incident report is received the problem is identified as either a new error or a repeat of a previous one – a "Known Error" – and a Known Error Log (KEL) is held on the data base, recording details of each Known Error by the symptoms of the problem and the solution to be applied. Microfiche copies of KELs are supplied to customers on a regular basis, to enable their own support staff to check such errors.

A copy of the MDB is held within the SSMS support system described below and is known as the Maintenance Knowledge Base (MKB). This is the data base used to do the fingerprint look-ups described in para. 6.1.1 below.

In the early days of the service all calls from customers were logged manually, but obviously this could not continue and a system called SSMS – SAM System Maintenance Server – was developed, having three components:

SRCS SAM Request Communications Server
SKBS SAM Knowledge Base Server
SRMS SAM Request Manager Server

whose functions will become clear in the following account.

For the MSR, in the customer's installation there is SAM, running under VME and monitoring all the activities – tape loadings, errors in magnetic peripherals, communications errors, system reloads ...; and in the support centre is SSMS, ready to receive communications from SAM. When SAM decides that some information should be sent to ICL it generates a MSR and transmits the information to SSMS. How this is dealt with depends very much on

(a) whether it concerns hardware or software
(b) whether or not a "hit" is found in the Knowledge Base
(c) whether or not the situation is covered by the customer's contract with ICL.

We consider hardware events first, then software.

6.1 Hardware events

Associated with the MSR will be the "fingerprint" of up to 255 bytes, giving the relevant information. Raising the MSR will cause a message to appear on the site operator's terminal screen, typically

> SAM REQUEST/S AWAITING TRANSMISSION TO ICL (VIA ADxx) TRANSMIT NOW (Y/N)

after which the MSR text will be displayed on the SAM communication screen, ready for transmission. The operator should check that this has not been generated by mis-operation or invalid use of the system and could assign a priority, although this latter is not usually done for hardware MSRs. Then to transmit the message he calls a specified telephone number (this is in fact the first of a "hunting group", which is searched automatically for a free line) and when the connection is made types TELL SAM, COMMS at the operating station, getting a response SAM/ICL COMMUNICATION COMMENCING.

The MSR is transmitted over an Application Data Interchange (ADI) link in what is called FINGERPRINT LOOKUP (FPLU) mode. At the ICL end the Communication Server software captures the information and causes the fingerprint to be "looked up" in the Maintenance Knowledge Base (MKB). If a hit is registered this means that the Knowledge Base has a solution to the problem, and this is transmitted to the customer over the same link. It is then processed by SAM, whose next action will depend on whether or not this solution has already been supplied to this site.

If this is the first time that this problem has occurred at this site, SAM will now raise a Corrective Action Request (CAR) to prompt ICL that a solution is available. If the solution sent had already been tried and found unsatisfactory, or if no solution had been found in the Knowledge Base, SAM will raise a Resolution Action Request (RAR), which is in effect a request for support.

At the end of the transmission the customer site will receive a prompt

> SAM/ICL COMMUNICATION ENDED. PLEASE ACKNOWLEDGE

which is acknowledged and the SAM link broken. ICL will now validate the Corrective Action and arrange for it to be done.

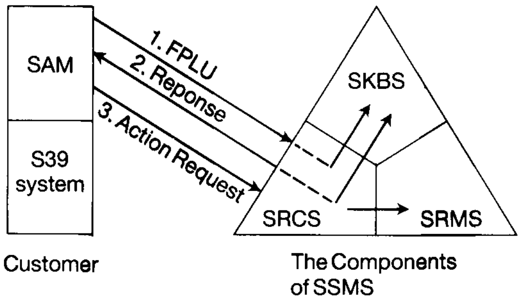


Fig. 4

All this will have taken about one minute, and has gone on without any human intervention other than making and breaking the link. At the end of the session the details of the MSR and the results of the look-up, along with the CAR or RAR, will have been logged in the SRMS data base.

Figure 4 shows this interaction.

6.1.1 Call logging and escalation. The above account has left untreated a number of administrative points concerning the handling of calls for support. We now discuss

how the call is handled in the Request Management system
how the data is held in the SRMS database
how the data is scheduled to a diagnostician
how the data is passed between the support units

all of which are very important to the smooth running of the service.

Taking the hardware example as typical, the data sent over the ADI link will be written immediately to a disk file called RTL – Request Transaction Log. This is a serial file containing a record of all transactions between the SAM Communications Server and the customer's SAM system.

The SRMS software reads the RTL file made by the Communications Server, creates an entry in its data base, generates the next number in the sequence (the RMS identifier, the Request Manager sequence number in the data base, providing the most commonly used method for identifying calls between units) and associates it with a screen containing the customer's details, the MSR number, the fingerprint, the priority and other relevant information. This screen can be accessed by ICL staff, using a command DISplay DETails from within SRMS. The mechanism that generates the screen also creates the initial entries in the diary associated with this RMS identifier.

In the diary the call is allocated a STATUS, with four fields and displayed as follows:

PROGRESS STATE/PROGRESS QUALIFIER/LOGICAL ACTIONEE/ACTUAL ACTIONEE
PROGRESS STATE is fairly self-explanatory; among the possibilities are:

ALLOC:	the MSR requires allocating within the unit
OFFLSUP:	the problem is being dealt with off-line
APPLYCA:	apply the corrective action (recorded in the diary)
EXPORT:	deeper analysis required – the customer has been requested to send diagnostic information, usually a dump, to the centre
CLEARANCE:	the call has been cleared and is being monitored

PROGRESS QUALIFIER can be used to give additional information on PROGRESS STATE
LOGICAL ACTIONEE usually means a group or team: thus SSC BRS means the Bristol System Support Centre.
ACTUAL ACTIONEE usually identifies the individual allocated to the call; it can be used also to direct calls to specialist groups, e.g. NODE for a node or system diagnostician, CONT for a controller specialist.

For the hardware example discussed, the first entry generated by the system would be

LUREP/NO QUAL/SKBS/NONE

meaning that the MSR has been received by the Communications Server, is being looked up in the Knowledge Base (SKBS), there is nothing further to say about its progress state and a reply to the look-up (LUREP) is awaited.

After the reply has been received from the knowledge base and sent back successfully to the site the MSR status changes to

LUREPSENT/NO QUAL/REPLYHK/NONE

If there was a hardware error and SAM accepts the response and sends a Corrective Action Request the next status is

ALLOC/NO QUAL/NONE/NONE

Up to now all the entries have been made by the system, by SRMS in fact. The progress state ALLOC now indicates that the MSR must be allocated to some person or group in the unit, and the relevant servicing Branch must look at the MSR data and either resolve the problem or pass it up to the next level of support – the second line.

From now on most of the changes to the status will be made by the users of the system, for example Service Desk operators, technical specialists, diagnosticians; each time they make a change they must put an entry in the diary, giving their reasons.

If the Branch wishes to pass this MSR to the next level, which is the Regional Support Centre, all it has to do is to change LOGICAL ACTIONEE to either NRSC or SRSC as appropriate. To assist in scheduling the handling of the problem they would show in the QUALIFIER field the type of service unit required, so that for a disk controller problem the status could now read

ALLOC/HSDC/SRSC/NEW

and finally the team leader at the centre would allocate an individual to the problem and record this in the ACTUAL ACTIONEE field.

Figure 5 shows this routing and escalation process, and Fig. 6 shows the actual records made by the system for a typical hardware problem.

6.2 Software events

Suppose the Series 39 system has detected a software error and has dumped and reloaded itself, or has experienced a communications incident; SAM will pick this up and generate a MSR. The actions that follow immediately are the same as in the case of a hardware event: a prompt to the operator,

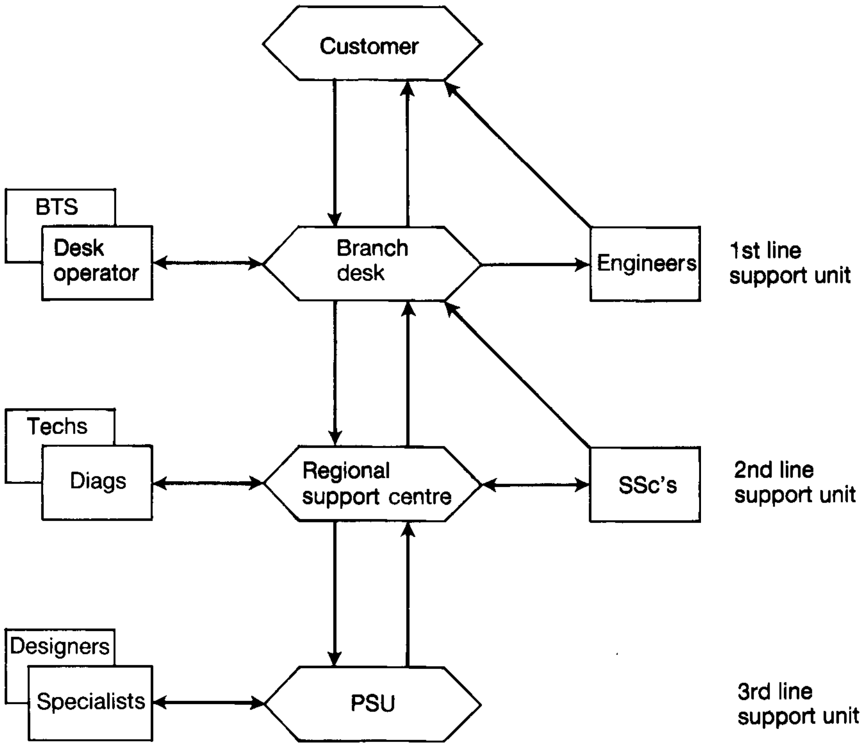


Fig. 5

```
                    * DETAILS OF REQUEST JP0046SR00087 ( 1208 ) *        ONLY PAGE

Progress state      : CLEARANCE            Date raised       : 1988/01/27 16:02:54
Progress qualifier  : CLR905               Last updated      : 1988/02/05 14:06:09
Logical actionee    : CLEARANCEHK
Actual actionee     : NONE                 Short name        : CGW
Priority            : X                    Site name         : 27068
                                           System ID         : JP0046
Service unit        : FD49
Service unit type   : FDS2500              Branch            : FEL01
Error class         :  4                   Management unit   : 0186/FEL01.S

Fingerprint : AC:4100:UT:FDS2500 :DE:MECH. ERRORS - BUS OUT PARITY ERROR OR TA
or          : G OUT SEQ ERROR:
Request     :
Summary     :

Last communication : 1988/02/05 14:06:07  REQUEST ENTERED CLEARANCE
Last diary entry   : 1988/02/05 14:06:09  DTLS:FSHY
Last entry text    : CLEARANCE      CLR905      CLEARANCEHK   NONE        X

COMMAND:
```

```
                    * DIARY DISPLAY FOR JP0046SR00087 ( 1208 ) *        FIRST PAGE

  DATE   |TIME |TYPE:NAME| DIARY TEXT
88/02/05|14:06|DTLS:FSHY| CLEARANCE      CLR905      CLEARANCEHK   NONE        X
88/02/01|11:22|NOTE:FSHD| SEE ENTRY ON RMS 1207.
88/02/01|11:22|DTLS:FSHD| APPLYCA        HD4A        FEL01         FRC19825    X
88/01/29|15:47|DTLS:FBLA| OFFLSUP        MANCA       FEL01         NONE        X
88/01/29|13:59|NOTE:RMAN| ************* CORRECTIVE ACTION FOR FD49 *************
        |     |         | CHANGE G476 PCB, P/N 88804476, IN HD4A POS'N A4B.
        |     |         | NOTE THIS IS SAME C.A. AS MSR 86 ON RMS 1207.
        |     |         | ***************************************************
88/01/29|13:59|DTLS:RMAN| ALLOC          MANCA       FEL01         NEW         X
88/01/29|12:25|DTLS:RMAN| ALLOC          FDS2500     SRSC          RMAN        X
88/01/29|11:31|DTLS:FROL| ALLOC          FDS2500     SRSC          NEW         X
88/01/28|10:48|NOTE:FBBL| CSAR CALL NO: 19825.
88/01/28|10:48|DTLS:FBBL| OFFLSUP        FDS2500     FEL01         NONE        X
88/01/27|16:04|DTLS:SRMS| ALLOC          NO QUAL     NONE          NONE        X
88/01/27|16:03|DTLS:SRMS| LUREPSENT      NO QUAL     REPLYHK       NONE        X

COMMAND: +

SESSION :SSMSSV.RA (DOE3) STARTS AT 1988/02/10  15:19:02
```

Fig. 6

transmission of the MSR, look-up in the knowledge base and response to the site – all of which can happen in less than a few minutes. The response will generate a print-out on the line printer at the site; if a "hit" has been found the necessary software repairs will be printed and also will be recorded in SAM's resource database. Whether these are to be applied by ICL or by the customer depends on his contract with ICL; but the general effect is that the system could have crashed, the ICL centre informed and a fix obtained within a minute, without calling in any human help except to dial-in the line and this service is available 24 hours a day, every day of the year.

If a hit is not registered in the knowledge base then the details of the MSR, including the fingerprint, are printed out at the site; a software fingerprint is usually in the form of a Procedure Name and a Displacement (within that procedure), for example:

ML:05:TY:01/SGSE/PE:PN:DHEXTENDFILETABLE:DI:X00174:
PE:05/01:

This indicates that there has been a Program Error of type 5.1 (PE:05:01) in Procedure DHEXTENDFILETABLE at Displacement hexadecimal 174.

Again, whether this is handled by ICL or by the customer depends on the contract: let us assume that it is his responsibility. He will then have a copy of the Known Error Log (KEL) on microfiche, which he will scan to look for a match against the fingerprint. If he succeeds he can request the repair listed there, either from his local System Support Centre or, by using a recently introduced facility, directly from the Maintenance Knowledge Base. After receiving and implementing the repair he will close the MSR in SAM.

If no match is found in the KEL, whether by the customer or by ICL, he will log in to SAM and cause the MSR to be retransmitted to ICL, with a priority assigned. It will then be dealt with as in the case of a hardware event, if possible by the Branch but usually at second line, where this support request will be actioned by a Support Centre diagnostician who will contact the customer and request a link to enable him to investigate the problem. Once the RSA (Remote Session Access) link has been established the diagnostician will look at the software dump interactively, and when he has solved the problem the Corrective Action is put into the SAM diary for implementation. Should this action require a software repair to be sent to the customer this can be done by the Support Centre, the repair being sent over the line and placed in a library on the customer's system to be installed when convenient to him.

The process is illustrated in Fig. 7.

7 Security

The system clearly involves ICL staff in gaining access to customers' installations, and the company recognises very clearly that this must be rigidly controlled. There is therefore strict control, by passwords and password changes, over who is able to access SAM and VISA; and ICL is allowed this access only with the agreement of the customer, who initiates the connection and can break it at any time.

8 Impending developments

As would be expected, the support system is not static but is being developed as needs grow and experience is gained. The following will have been made by the time this paper is published.

1 A Resource Retrieval Facility will be provided, by means of which customers can obtain, electronically, resources directly from the Maintenance Knowledge Base. These resources are

```
                  * DETAILS OF REQUEST JF1353SR00057 (   880 ) *        ONLY PAGE

Progress state       : CLEARANCE              Date raised        : 1988/01/22 16:49:03
Progress qualifier   : CLR 002                Last updated       : 1988/01/28 14:22:35
Logical actionee     : CLEARANCEHK
Actual actionee      : NONE                   Short name         : ASD
Priority             : C                      Site name          : 16059
                                              System ID          : JF1353
Service unit         : KABLS SYST02
Service unit type    : KABLS 0003             Branch             : FEL01
Error class          :  5                     Management unit    : 0750/FEL01.S

Fingerprint  : ML:08:TY:02/IDLE:PN:SEIMHANDLECABINETKEY:DI:X00034:
or           :
Request      :
Summary      :

Last communication : 1988/01/28 14:22:33  REQUEST ENTERED CLEARANCE
Last diary entry   : 1988/01/28 14:22:35  DTLS:FBBL
Last entry text    : CLEARANCE     CLR 002      CLEARANCEHK    NONE        C

COMMAND:  +

SESSION :SSMSSV.RA (DOE3) STARTS AT 1988/02/10  15:19:02

                  * DIARY DISPLAY FOR JF1353SR00057 (   880 ) *        ONLY PAGE

~~~~~~~~|~~~~~|~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  DATE  |TIME |TYPE:NAME| DIARY TEXT
________|_____|_________|_________________________________________________________
88/01/28|14:22|DTLS:FBBL| CLEARANCE     CLR 002      CLEARANCEHK    NONE        C
88/01/27|17:49|NOTE:RSTR| NCMSILHEADERTABLE POINTED TO A FAIL IN LSHDISCONNECT-
        |     |         | INGSTATETIDY+X182, RC:77557. REP KD/5/1232.3 SENT.
88/01/27|17:49|DTLS:RSTR| ALLOC         CLR002       FEL01          NEW         C
88/01/26|11:48|NOTE:RSTR| SAM IS DEAD, WITH RPER
88/01/25|15:18|NOTE:RSTR| INSUFFICIENT SPACE IN ICLMAINTCT FOR SAM TO ARCHIVE,
        |     |         | SO SAM NOT RUNNING.  SITE WILL TIDY MAINCT UP AND I
        |     |         | WILL LOOK AT DUMP TOMORROW WHEN HOPEFULLY SAM WILL BE
        |     |         | RUNNING.
88/01/25|09:12|DTLS:RKIL| ALLOC         KABLS        SRSC           RSTR        C
88/01/25|09:01|NOTE:FROL| CSAR CALL NO 18972
88/01/25|09:01|DTLS:FROL| ALLOC         KABLS        SRSC           NEW         C
88/01/22|16:53|DTLS:SRMS| ALLOC         NO QUAL      NONE           NONE        C
88/01/22|16:49|DTLS:SRMS| LUREPSENT     NO QUAL      REPLYHK        NONE        C
88/01/22|16:49|DTLS:SRMS| LUREP         NO QUAL      SKBS           NONE        C

COMMAND:  +

SESSION :SSMSSV.RA (DOE3) STARTS AT 1988/02/10  15:19:02
```

Fig. 7

news items: a NEWS data base will be held on the Maintenance Knowledge Base (MKB) and will be available to all customers. It will be indexed by NEWSBYDATE and NEWSBYSUBJECT; external customers will become aware of the total NEWS content only by enquiring of these indexes

software repairs, either singly or as a package. This has been mentioned in para. 6.2; it was available to some sites in late 1987 and will be available to all sites in February 1988.

2 The system will be moved from the 2966 machines to a multi-node Series 39 Level 80 installation early in 1988.

3 A more sophisticated communication network to allow software dumps (exported on magnetic tape) to be handled centrally. These dumps will be

loaded on the system running the service at Hitchin and the output spooled directly to the relevant support centre at Reading, Elstree or West Gorton. This will greatly reduce the time the dump is in transit.

8 Conclusion

From June 1986 to December 1987 the Series 39 customer data base grew by a factor of 5, with a corresponding doubling of the workload in the support centres: the support process successfully kept abreast with this increase. The move to Series 39 hardware, and the release of further developments in VME, will enable us to face the future with confidence.

The lessons learned in the early days have enabled us to refine the logging, scheduling and escalation facilities into an easily controlled fault management system within ICL. The power of the combination of VISA and the Node Support Computer has been proven and has given us the ability to bring the highest skills within the company to bear on a problem quickly, whether this has arisen in Manchester or in Melbourne.

Acknowledgements

My thanks to my colleagues in the Southern Region Support Centre for their comments on this paper.

The ICL system support centre organisation

J. Young
ICL (UK) Southern Region, Regional Technical Support, Reading, Berkshire

Abstract

The role of a System Support Centre (SSC) has changed radically over the past few years; whilst the term SSC is still in regular use it no longer carries the earlier meaning of a single purely diagnostic unit. The paper sets out to describe in some detail the make-up of and functions performed by the modern SSC, based on the organisation in the ICL (UK) Southern Region.

1 SSC organisation

The overall term 'SSC organisation' is used to describe three organisations which work closely together:

System Support Centre
Responsible for providing Support and Services on VME Base and Superstructure Products.

Network Systems
Responsible for providing Support and Services on Networking and Communications.

Regional Control Centre
Provides administration and control facilities to the above units. Control Centre staff work within each of the units as Administrators.

The SSC itself is made up of several teams:

Senior Consultants
Superstructure
VME Support & Services
VME Maintenance

2 SSC responsibilities

The SSC's responsibilities fall into three distinct areas:

The first is to provide software support and maintenance.

The second is to provide backup support to ICL field engineering and to contribute to pre-sales issues identified by account support staff.

The third is the provision of chargeable software services and technical consultancy.

The following sections of this paper cover the responsibilities in more detail.

3 Software support and maintenance

The SSC is responsible for direct support to customers as committed in their VME software licence. This involves telephone assistance as well as the traditional SSC role of 'filtering' product problems submitted by customers on Product Reports.

3.1 Product reports

If a suspected error is found in a software product and a solution cannot be found by customer technical staff, then a Product Report Form (PRF), Known Error Worksheets (KEW), Diagnostic Error Worksheet (DEW) and any necessary supporting evidence should be sent to the SSC.

The Known Error Worksheet provides a scratchpad when searching the Known Error Log (I will explain the KEL later). When attached to the Product Report Form, it indicates to ICL staff the level of analysis already attempted and eliminates duplicated effort by serving as a record of successful and unsuccessful KEL searches.

The Diagnostic Error Worksheet indicates, for different types of error condition, the particular evidence and additional information that is required either to be recorded on the worksheet or provided separately. There are worksheets for VME, CME, IDMS (on VME) and TPMS, each differing slightly in approach but with the same overall purpose.

When used and completed accurately KEWs and DEWs enable a higher level of problem resolution, on site, by customer staff, thus avoiding unnecessary delays. It will also ensure that when a problem is passed to ICL, all necessary evidence will be passed with it and again avoid unnecessary delays.

Additional assistance in collecting evidence for problems may be obtained from the SSC or from documentation such as the 'VME User Guide'. Detailed procedures for submitting Product Reports are also available via the SSC.

All Product Reports received in the SSC are logged by the Regional Control Centre (Admin) onto a central ICL database known as the Maintenance

Database (MDB) as soon after receipt as possible. Details of hardware, software, software version numbers, date of failure, component, priority, report number and evidence received are logged. Once these transactions are complete the Product Report and evidence are delivered to the appropriate team for investigation.

Each team will investigate the problem and aim to meet the 'SSC Service Measures'.

What are Service Measures? Before considering Service Measures it is important to understand the priorities attached to Product Reports and the classification of resolution or 'Closure Categories' as they are better known.

3.1.1 Product Report priorities. Each Product Report is assigned a priority by the customer. The priority in the range A to E corresponds to the impact with which the problem on site is perceived. One example of each priority is listed below.

Impact	*Priority*
System not usable for any purpose	A
Frequent system or communications failure (> 1 per day)	B
Less frequent system failure (between 1 per day and 1 per 7 days)	C
Non-critical malfunction	D
Information only, Usability suggestion or Documentation	E

3.1.2 Closure categories. The closure category is a numeric character and may be any one of the following:

Category	*Meaning*
1	*Product Error – No clearance planned* An error has been recognised in the issued product but there are no plans to provide a solution.
2	*Product Error – Source clearance scheduled or repair available* An error has been recognised in the issued product and a repair is available or a source clearance of the error is planned.
3	*No fault in ICL product* The cause of the problem has been identified and does not stem from a fault/deficiency in an ICL product.

4 *Documentation Deficiency*
The problem was caused by incorrect, misleading, incomplete or ambiguous documentation.

5 *Hardware Fault*
The reported problem was caused by a hardware malfunction not of a design nature.

6 *Unpublished Known Error*
The reported problem is known to the support group closing the incident but not available to the originator.

7 *Withdrawn by User*
The originator no longer wishes the problem to be considered.

8 *Unresolved*
Work cannot continue on the reported problem on the evidence available, and further evidence cannot be reasonably obtained.

9 *No Error – Consultancy/training required*
The problem reported is not an error in an ICL product. It is a problem which can be better resolved by use of a consultancy or training service.

10 *Published Known Error*
The reported problem is available in the known error file or other relevant documentation, and it is reasonable to assume this is available to the originator.

11 *Enhanced Request*
Existing evidence indicates that the product performs according to its specification but the user meets a problem in putting it to legitimate use and wishes to extend the specification.

12 *Administrative Closure*
The problem has been incorrectly entered onto the Maintenance Database.

13 *Further Evidence Required*
Work cannot proceed on the basis of the submitted evidence.

3.1.3 Service measures. When dealing with Product Reports, SSC performance is measured against the following targets:

Filtration: Target 80%. SSC's are allowed to close Product Reports using categories 3, 5, 7, 9, 10 and 13. All other categories are reserved for Business Centre usage except in some special circumstances.

The filtration effectiveness of the SSC is measured using the closure categories of reports transferred from the SSC to the Business Centres. In outline it is the percentage of the reports that the Business Centres close in 'non-SSC-closeable closure categories' (which are categories 1, 2, 4, 6 and 8. NB: 7, 11 and 12 are ignored in the calculations). In simpler terms it means that the SSC is measured on the number of non-product errors that it fails to diagnose.

Timescales: Target 80%. A target was agreed with the VME User Group that 80% of all reports would be closed in n-days (where n = 4 days for B priorities, 10 days for C priorities and 40 days for D and E priorities. Priority A reports are closed immediately, i.e. continually worked on until resolution).

To meet this:

Procedures are in place to deal with the reports in the SSC within half life (n/2) to allow the Business Centres to meet the overall target unless the report can be resolved in the SSC within full-life (n days).

Bug factors: Target (in-house) BF/Bug < 1;(transferred) BF/Bug < 4. The bug factor is a combined measurement of an individual report's age and priority (the oldest, high priority reports carry the highest bug factor).

The target is to maintain the in-house bug factor/bug (for all reports retained at the SSC) to less than 1.

In addition the bug factor/bug for all transferred reports has a target of less than 4. This involves progressing and chasing transferred reports and providing assistance where necessary to enable the Business Centres to maintain this target.

Backlog: Target less than 1 week's average input. To meet the other targets above there is an additional target of reducing the Backlog/Average Weekly Input to less than 1 (i.e. one week's worth of input in the backlog). When this occurs the backlog is really work in progress.

3.2 Prescan

On arrival at the appropriate team the first action to be performed is the registration of the Product Report within a local control mechanism used to track the problem throughout its life in the SSC. This can be a computerised system but most teams still prefer the visibility of a T-card rack.

The colour of the T-card denotes priority:

B = Pink, C = Yellow, D = Blue, E = Green.

Each T-card is marked with the customer's reference, priority, product area, date of input and date of half life. Once this action is complete the problem is prescanned according to its priority bearing in mind the Service Measures. 'A' or 'B' priorities, those known to be critical and reopened Product Reports will be dealt with immediately.

Prescanning may be performed by product specialists or by a specific team of 'prescanners'. The function is to ascertain whether enough pertinent information has been supplied or if the problem is already known to ICL. During this phase a prescanner will complete a 'worksheet' to record his investigation. The purpose of this worksheet is to record any notes that may be of relevance to the next investigator/specialist, should the problem not be resolved at this point. A Product Report is then closed, transferred to a particular Business Centre or passed to another specialist.

3.3 Diagnostic tools

Throughout the diagnostic process, whether in the initial prescan or during the further in-depth analysis, specialists have a range of tools available to them. The most important of these are the Known Error Log, Project Log and Diagnostic Guide.

3.3.1 Known Error Log. All software errors known to ICL are logged on the Maintenance Database (MDB). A set of microfiche detailing Known Errors, and derived from the MDB, is produced and distributed, by ICL Group Information Systems, weekly to all registered VME customers and support staff.

This set includes:

VME Known Error Log (KEL)

Superstructure KEL(s)

Communications Products KEL

Recommended Repair State/Star KEL fiche (issued when required).

The Recommended Repair State/Star KEL fiche is a collection of repairs to errors considered of vital importance, e.g. data corruption or system break.

Each KEL consists of all, or some of three parts:

The introduction, which describes amongst other things how to use the KEL.

The cross index, which is the means of finding an entry in the KEL.

The Known Error Section detailing all the Known Errors; this section is divided up by product on some KELs.

The KEL enables specialists to see if a reported problem arises from a product error which is already known about and shows whether repairs or avoidance actions have been identified.

The KEL may also be interrogated on-line using CAFS facilities. This allows specialists to use 'fuzzy matching' on textual or numeric strings rather than on well defined keywords.

The SSC's play a significant part in improving the KEL and maintaining it up to date. KEL entries are raised by SSC staff to:

- Document new problems – Normally this will accompany a transferred Product Report Form but on occasion may be the result of practical experience.
- Amend existing KEL entries. These will be raised when an existing KEL is seen to be incorrect or inadequate.

Although SSC's raise KELs they must be forwarded to Business Centres for authorisation.

3.3.2 Project Log. The Project Log exists for the entire VME Operating System. It comprises microfiche S3 source listings of the individual procedures within each System Version of VME.

The Project Log is usually filed in Chapter and Subsystem order. Since the SSC supports more than one System Version at any one time, there are normally a large number of fiche cabinets resident within each unit.

The first chapter is in many ways the most important, for it contains a miscellaneous collection of useful information. It is split into subsections and contains items such as:

Result Code Descriptions
Message Texts
Macros
Interactions
Holon Trees.

The remaining chapters hold in holon order all the S3 listings. This large volume of code is in daily use for it is this that enables the specialists to identify the exact line number of the failing piece of code.

3.3.3 Diagnostic Guide. The Diagnostic Guide is designed as an aid to those with diagnostic experience of VME. The Guide is organised by Chapter and Subsystem similar to the VME Project Log. Each section gives a short description of the subsystem followed by a more detailed explanation.

The first chapter contains an index to the Project Log, which lists in order all the VME chapters with their subsystems, and an index to the Guide, which lists all the items from the Project Log that appear in the Guide.

In addition the first chapter of the Guide contains the Diagnosticians Handbook which is a miscellaneous collection of useful information that could not be placed within a Project Log Subsystem, e.g. a list of program error types.

All other chapters of the Guide deal with the respective chapter of the Project Log. Within each chapter are sections, each of which deals with a subsystem of the relevant chapter of VME.

These sections contain:

Summary of any changes in the subsystem for previous releases of VME.

A general description (overview) of the subsystem.

A detailed description of the workings of the subsystem including table layouts and description of procedures and variables within them.

3.4 Specialist investigation

Product Reports requiring further investigation, if necessary, will be queued in priority order within product area, awaiting the attention of a specialist. The corresponding T-card will be placed in the 'unallocated' section of the appropriate team's T-card rack.

T-card racks are organised in sections:

Unallocated
Allocated
Closed
Transferred.

A specialist will either work on a Product Report allocated as 'immediate' by the appropriate Team Leader or will choose a report depending on his area of specialisation and skill, and according to priority in the queue.

Once chosen he will move the T-card to the 'allocated' section under his name. When his investigation has been completed he will either close or transfer the Product Report and move the T-card to the appropriate section in the T-card rack. If he cannot complete his investigation due to some other urgent assignment he will notify his Team Leader for appropriate action.

3.5 Product Report transfer

In simple terms the aim of the SSC is to resolve as many problems as possible, but in some cases they have to transfer the Product Report to a Business Centre or another unit.

The following are examples of the types of problem that are transferred:

- New product problems, i.e. product reports thought to document previously unreported bugs. In this case the Product Report must be transferred to the Business Centre whether or not it can be solved in the SSC, as the SSC is charged with informing the Business Centre of new bugs, and only they have the authority to decide on resolution.

- New usability or documentation problems.
- Products where the skill is not held at the SSC.
- Product Reports which the SSC staff are simply unable to solve within reasonable timescales.

When transferring a Product Report to another unit, a Transfer Slip is completed with today's date, name and telephone number of the transferrer and a location code of the new destination.

If the destination is a Business Centre then an Interim Response will be sent to the customer stating this fact.

Before further movement takes place the Product Report, worksheets and Transfer Slip are collected together and checked for Quality by either another specialist or the Team Leader.

The purpose of this check is to ensure:

All the evidence is attached.

The worksheet(s) and Transfer Slip are fully and properly completed and give sufficient detail for the next investigator.

The transferrer has put in sufficient effort in dealing with the report.

In this manner Quality can be monitored and maintained so that the unit may retain its right to 'fly' the BSI Kite.

Finally the Product Report is returned to Admin where the Movement History on the MDB is updated before the Product Report is put into the internal post for delivery to the appropriate unit. This service guarantees next morning delivery.

3.6 Product Report closure

Every member of the SSC who closes Product Reports is responsible for:

Providing a typed response which is technically accurate and concise.

Giving guidance that would help the customer prescan more effectively (e.g. Known Errors that should have been found and how).

Telephoning the customer in cases of requests for further information on high priorities. 'A' and 'B's.

When a specialist completes his investigation he must select the appropriate closure category and type a response on the in-house Response system. This system provides a wordprocessor-like facility for producing legible responses rather than one with dubious quality handwriting.

Before the response is printed, the responder will ask another colleague (preferably one with a skill in a similar area) to check his response.

The response is checked for:

completeness

the English

all the questions on the Product Report have been answered

the correct closure category has been used.

If the Quality check is OK then the response is countersigned and the T-card moved to the appropriate section in the T-card rack. A final quality check is carried out by the Team Leader before the Product Report is returned to Admin. At this point Repair(s) are attached to the Product Report, if a solution has been identified, before being returned to the customer, i.e. closure category 10.

3.7 Telephone assistance

The SSC provides a telephone service direct to customers and ICL staff. It has an objective to respond to calls within 24 hours for contracted TELSUP customers or 5 working days for all others. (TELSUP contract is explained in section 5 – Professional Services.)

The Admin unit take the initial call and will ask customers for their User Prefix (unique three or four character site identifier), name, telephone number and brief details of the query. These details are entered into a Call Logging System after which the customer is given a Call Number and the call passed to a specialist, or if no suitable person is available, 'stacked' for the customer to be called back. In the latter case the call number provides a reference to the initial call.

The specialist has the responsibility of ensuring that he meets the timescale objective and that he gives a quality service. He will deal with the call, completing any research necessary, and when a satisfactory response has been given, he will enter brief details of the response into the Call Logging System and close the call.

The Call Logging System provides an audit trail of all calls received, and prevents calls being lost. It is also used by management to monitor the achievement against target.

In the case of customers who have not contracted for TELSUP or a service which includes it, the SSC will only accept calls in the following categories:

Querying a Product Report Response

Clarification of an existing KEL entry

Warning of an urgent imminent Product Report

Repair Request.

3.8 Repair request

Where a solution to a problem found by a customer on the KEL is a repair number, the repair can be requested from the SSC by post or by telephone. If a repair is restricted (untested) then a telex is required, accepting responsibility for the repair on site.

The request is dealt with by a member of the Admin unit who will enter details of the request into the Call Logging System. Repair number, KEL number plus the release/version required must be given.

Repair issuing is dealt with as soon as possible but if a request is urgent then priority can be given to the request if required.

The SSC will accept a request for a maximum of ten repairs over the telephone otherwise a letter, telex or fax is required. This helps to minimise delays and mistakes.

If large numbers of repairs are required at any one time, the SSC can put these on tape. However, a charge will be made for this service.

In addition to the above, Series 39 customers may make use of facilities for the electronic retrieval of repairs. 'Automated Resource Retrievel Facilities' are now available within the SAM Product (Support and Maintenance). The facilities enable customer systems electronically to obtain repairs from an ICL database, called the Maintenance Knowledge Base (MKB).

So much for the traditional support and maintenance role, now let us consider the SSC's other responsibilities.

4 Pre-sales

The SSC provides pre-sales assistance to Sales Branches in three main areas:

Risk Appraisals

VME presentations in support of sales campaigns

Proposal Clinics.

4.1 Risk Appraisals

Product and service offerings made to customers must be commented upon in such a way as to help the salesman achieve the sale but avoid system combinations which, in the view of the SSC, do not represent long term 'good business'.

The Risk Appraisal document details such offerings and provides a mechanism whereby relevant specialists may make comments on areas within their specialisation.

The SSC provides resources to technically vet all Risk Appraisals in the area of large systems mainframes and VME. The products quoted and the services offered are vetted to ensure they are technically viable and provide a sound basis for a VME solution.

Unit management receive all VME Risk Appraisals from the Regional Control Centre and are responsible for providing a response to the Control Centre for each Risk Appraisal before the specified deadline. Suitably skilled VME Consultants are involved and asked to consider and comment on issues such as:

- Does the software offered provide an adequate product set to support the proposed workload?
- Are there any special considerations to be taken into account for the particular customer?
- Have appropriate services been sold? If not are there any known reasons for this and possible future implications?

The Risk Appraisal will then be marked High, Medium or Low and returned to the Regional Control Centre for appropriate action.

Risk Appraisals are also used for forecasting purposes. New prospects are noted and the Sales Branch contacts regularly polled, so that the SSC can plan services accordingly.

4.2 VME presentations

The SSC from time to time receives requests from Sales and Account Teams to provide 'State of the Product' presentations for senior customer management and their technical staff.

Presentations must be designed in such a way as to present the benefits of VME in a manner appropriate to the technical awareness of the target audience.

On being assigned to Pre-Sales events consultants are expected to:

- Discuss with the Sales Team the overall objectives of the event.
- Summarise the main benefit statements and check that they are appropriate to the event.
- Understand the technical level of the audience and develop an appropriate presentation.

After the event the consultant will seek feedback from the Account Team as to the effectiveness of the presentation for that particular market or event.

4.3 Proposal Clinics

Proposal Clinics have a similar objective to Risk Appraisals, but take place before the products and services are proposed to the customer. The SSC provides resources at the clinics to technically vet Account Teams' impending offerings to customers and prospective clients.

5 Professional services

5.1 Packaged services

The SSC provides the skills necessary to develop, enhance and deliver a number of standard packaged services designed to meet the demands of the Large and Distributed Systems customers. These services are defined in Product Descriptions which form the basis of a services contract, and are delivered in accordance with Service Guides and workbooks provided by ICL Services.

These services include:

- Planning and installation services, e.g. VME START-UP Service.
- VME Upgrade Planning and Implementation Service.
- Software Maintenance services, e.g. Super 29 Service (MAINPI), Super 39 Service.
- Telephone Consultancy Service (TELSUP).

Product Descriptions are as the name suggests, documents produced to enable the customers to satisfy themselves whether the Product/Service would be suitable for their needs. Rather than append these documents I have summarised them below.

VME START-UP Service. This service provides for the start-up of customers moving to Series 39 VME. The service comprises:

- Formal training for a member of the customer's staff. Training will instruct the attendee on the administration of the VME service to be built on the customer's Series 39 System.
- The production of a viable VME System Build Plan in conjunction with the customer's staff. The plan, agreed by the customer, will cover such topics as cataloguing, user structure, filestore, security and operation.
- The provision of an operational VME Service built according to the System Build Plan.
- Documentation of the System Build activities.

VME Upgrade Planning and Implementation Service. This service provides for upgrading of one loadset of an existent VME service to a specified release. The service comprises:

- Production of a plan sufficient to upgrade the VME service to a level of operation where the workload may continue to be run.
- The upgrade of one of the VME loadsets together with all licensed options currently in use.
- Any installation tests provided as part of the product will be run.

Super 39 Service. This service provides a customer who has limited or inexperienced VME staff with an ongoing software maintenance and periodic upgrade service. The service comprises:

- Control, planning and implementation of all VME Base software and Options for which the customer is licensed to an ICL defined release level.
- As above for ICL Superstructure products (i.e. excluding Applications).
- Advice and guidance on the change control, application and testing of ICL supplied software repairs and amendments.
- Telephone Support.
- Management of regular meetings held with customer staff to review progress, plans and problems.

Additionally, where a customer uses the TME Option on Series 39, this service provides for:

- Control, planning and implementation of all Software Maintenance Files (SMFs).
- Advice on new developments in TME base and superstructure software.

Telephone Consultancy (TELSUP). This service entitles a single customer site to telephone ICL for technical advice and guidance. Questions may be on the usage or exploitation of any of the operating systems, communications or superstructure products on DME, VME, CME or CME* systems. The service does not however cover Group 4 (Applications) products.

The service normally provides immediate consultancy, however it may be necessary, on some occasions, for ICL staff to call back within 24 hours.

Customers using this service should have sufficient knowledge of the problem area to understand a reasonably explained response.

5.2 Service delivery

Once a consultant has been assigned to a service and briefed on the basic details of the service delivery, it is his responsibility to make contact with a member of the Account Team, who can provide him with a further briefing on account specific issues.

A meeting with the customer is then arranged to check the customer's understanding of the service and agree the timescales. The service will then be delivered in accordance with the methods laid down in the Service Guide.

Service Guides describe in detail the methodology of service delivery. They also provide a comprehensive checklist of tasks to be undertaken.

On completion of the service the consultant will produce a Service Report, the production of which serves several purposes including that:

> To ensure that the customer is informed as to the actions performed on site. If, for example, part of the service was not performed, then the report should indicate the reason.
>
> To inform colleagues about problems encountered.

Copies of the report are passed to the customer and the Account Team.

On completion of each VME Start-up Service a questionnaire is sent to the customer. The purpose of this questionnaire, called a 'Service Assessment Form', is twofold:

> To enable the SSC to ascertain the customer's perception of the service delivery.
>
> To enable ICL to continually refine the service to meet the needs of customers.

Answers and comments returned are regularly reviewed by management. In this way the quality of the service and its deliverables can be constantly monitored and any adjustments made accordingly.

5.3 Ad-hoc consultancy

The majority of services are provided in a standard form at a fixed price but because of the variations in size and complexity of the systems which can be constructed with VME it is necessary for some services to be individually assessed and quoted.

The services will normally be specific short term technical tasks or short periods of consultancy sold on a time and materials basis. Good examples of this type of service are:

> Ad-hoc consultancy required to migrate a 2900 Series workload to Series 39.
>
> Consultancy tailored to the user's needs during a filestore re-organisation.

5.4 Technical workshops

The SSC designs and runs a series of VME technical workshops as and when product enhancements make them commercially viable. These are advertised by means of a letter sent to every registered VME customer twice a year (generally August and December/January). These workshops are aimed at customer technical support staff and cover such topics as:

Filestore
Performance
Loading Environment
TPMS Recovery.

5.5 Other services

Information and Operations Notices (IONs). The SSC is responsible for the production and distribution of Information and Operations Notices. These provide information on various subjects as the need arises. IONs are divided into several categories.

Procedural	(P)
Management	(M)
VME Base	(B)
Superstructure	(S)
Series 39	(L)

The information contained in an ION can be of major importance, particularly when printed on PINK paper. A 'Pink' ION gives advice which will avoid the possibility of data corruption. These should be read carefully and filed together for easy reference.

6 Conclusion

The paper has attempted to give the reader a brief insight into the working of a System Support Centre. There are at the moment five such units in the UK, all of which perform similar functions. However, they do so in slightly different ways, and for that reason deliberate generalisations have been made throughout the paper. The objective has always been to provide a highly responsive, high quality support service to customers and this will continue; but the nature and scope of these services, and the means by which they are provided, will doubtless continue to change in response to changes in scale, needs and underlying technology.

Acknowledgements

Grateful thanks are given to those who have contributed, knowingly or unknowingly, to this paper.

ICL Services Product Centre

M.D. Griffiths
ICL Mainframe Systems, West Gorton, Manchester

Abstract

Services Product Centre was established in the summer of 1987 as one of seven Product Centres in ICL's Mainframe Systems division. It has a key role to play in the provision of services for VME systems worldwide. It provides software tools for the support process, defines and packages services including the support services and provides the information needed to maintain and exploit VME systems and services. The paper describes the Centre's role in the provision of services and the techniques used in their development.

1 Introduction

1.1 The Services Product Centre

During 1987, Mainframe Systems development teams were organised into a series of seven Product Centres which fit together in a structure designed to harness the natural flow of the development process. Figure 1 shows the arrangement of these Product Centres; it depicts the flow of development of mainframe systems from the enabling technologies that are used by hardware

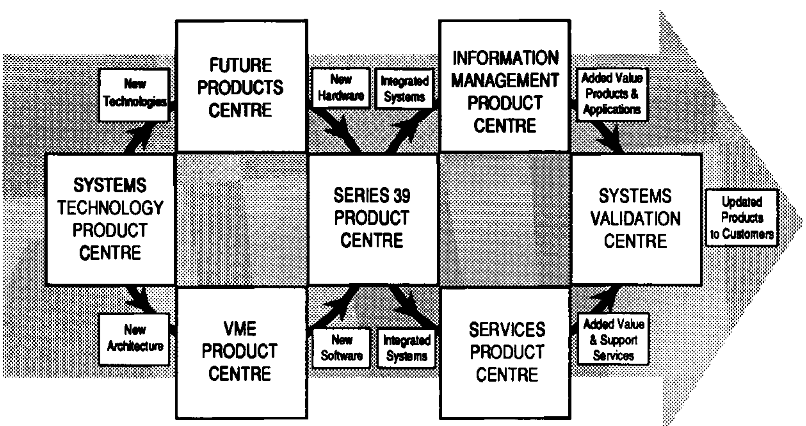


Fig. 1. Product Centre Structure

and software system development, through an integration process, with the addition of added value software and services and finally via a dedicated validation centre.

Each Product Centre has a clearly identified role to play in the provision of ICL's range of mainframe-based systems. Services Product Centre is one of three Product Centres that, combined, form Mainframe Systems Software and Services Directorate.

The other two Centres in this Directorate are:

Information Management Product Centre which produces fourth generation language products, application tools and applications that run on VME.

VME Product Centre which produces the operating system software, VME, and the transaction processing and database software, TPMS and IDMS.

The role of the Services Product Centre is to provide information and services to complement the software produced by the other Product Centres and the hardware developed by the Systems Engineering Directorate, which contains the remaining four Product Centres.

The Services Product Centre (SPC) has three production units, as follows:

Customer Information Project, which is managed by myself. This project is responsible for most of the technical information provided for Mainframe Systems. It also manages the channel into Manufacturing Division for software products and the related literature.

Systems Management Development: This project produces the service management software products SAM and SSMS that are the major components of the support process. It incorporates a team, *Services development unit*. This team produces the packaged services, which are described in section 2 of the paper.

Finally, the *Systems Introduction Unit* has the role of enabling new Mainframe Systems products to be introduced smoothly into the operations devisions and hence into the customer base.

1.2 Services and information

All of SPC's production units contribute in some way to the support of ICL's mainframe systems. In this paper I have chosen to concentrate on two key activities – the production of services and of information.

It is worth stating at the outset that I consider 'support' for our customers to be wider in scope than simply fixing problems. ICL has a strategic objective of being a 'solution supplier' and therefore differentiates itself in the market place from commodity suppliers. Support and maintenance for business solutions involves helping the customer to get maximum benefit from his investment in ICL products. The spread of activities may, therefore, range from installing and setting up a system to collaborative ventures with major customers to develop new methodologies for support: Central System Teleservice, described below, is an example of the latter.

The different categories of support and the different types of service are described in section 2 below.

My view is that three related disciplines must combine to provide the support needed; these are

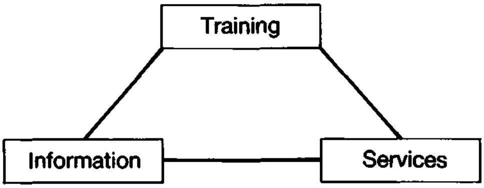


Diagram A

There is a great deal of synergy to be gained from having specialists on the production of Services and Information in one Product Centre. We shall consider later the idea of a "Packaged Service"; this, essentially, is defined in a document – the Service Guide – and the remaining 'evidence' of the service on the customer's desk is also a document.

Tentative moves have already been made to explore how the developing technology of computer-based training can play a part in the provision of significantly enhanced by the provision of embedded computer-based training. The inclusion of such training would have the advantage of standardising the basic learning elements of the service, while allowing the consultant delivering the service to concentrate on those aspects which require most tailoring to meet individual customers' needs. However, the potential use of computer based training in this way is still in the investigation phase and so it is not described further in this paper.

Section 3 of the paper describes some of the current strategic thinking behind the provision of information to mainframe customers. The typical design process that modern computer publications are subjected to is presented and examples of its use are selected from the suite of Series 39 support information.

2 Services

2.1 What services?

A traditional strength of ICL lies in its provision of hardware services, but these are only one aspect of the complete Information Technology Services industry. The industry has three growth sectors:

Software Services: these are concerned with requirements related to technical products, for example transition services from one system to another or software update services. Knowledge of the customer's business is not normally required here. Such services can include technical workshops for customer staff. Within ICL these services tend to be known as Customer Services; many are provided by System Support Centre (SSC) staff, as described in the paper by Young in this issue.[1]

Educational (or Training) Services: this refers to customer technical training on specific ICL products such as VME, TPMS, Quickbuild, etc.

Professional Services: these are concerned with management or business consultancy directed at the managerial processes and their use of information technology. Professional services are frequently related to business solutions and may be joint ventures in the case of our largest customers.

The table below shows the predicted growth patterns in the services industry.

Table 1 Predicted Growth Patterns in Service Industry

SERVICE SECTOR	PERCENTAGE OF TOTAL SERVICES REVENUE 1985	1990	ANNUAL GROWTH RATE
Hardware Services	83%	64%	4%
Software Services	10%	23%	31%
Professional Services	5%	9%	25%
Educational Services	2%	4%	20%

A significant customer trend is to prefer to purchase a comprehensive range of services from one supplier (known colloquially as 'one stop shopping') and it is this requirement that ICL must satisfy. A key role of Services Product Centre is to provide a portfolio of service products and support services to meet this need.

2.2 Packaged Services

Services Product Centre (SPC) has the major role to play in the production of specifically purchaseable packaged services. However, SPC staff do not themselves *deliver* the service to the customer. SPC's role is perhaps best described by reference to the three 'P's of Services: People, Process and Physical Evidence.

People. The consultants who deliver services to a customer are owned, within ICL, by the operating companies: ICL UK Ltd and the International Operations. The role of SPC here is to train and prepare consultants on the service itself and in the skills needed to deliver a service. Skills transfer workshops are regularly held to enable ICL consultants to qualify as specific service deliverers. Basic skills training, right down to the level of appearance and how to introduce oneself to a client, are covered, as well as the technical aspects of the particular service being taught.

Process. The process to be followed by a services consultant for any particular service is designed, developed and validated by the staff of Services Product Centre. The process is indeed the critical quality driver of a service – it is the process which is constantly updated in the light of feedback and experience.

A typical service will have several phases, for example:

Planning, much of which is done by the consultant away from the customer site,
Implementation, actually carrying out the work on a customer's system, and
Handover, which often involves a formal customer presentation and issue of a complete customer workbook.

For each service the process is defined in a consultant's *Service Guide*. This incorporates a description of all the tasks that have to be undertaken as a part of the service and guidance on how long each task should take. Essential checklists and specific pro-formas are included, as are sections on ICL's and the customer's responsibilities according to the service contract. Reminders of skills learned on SPC workshops are incorporated where they are most relevant to each particular phase. For example, the Pre-Service Preparation checklist may contain a reminder to contact the customer's sales support person for a briefing on any current problems before the consultant visits the site.

Physical Evidence. Different services will of course leave different physical evidence. A VME start-up service, for example, will leave a working VME system. However, it is important to leave some evidence that relates to the service itself rather than the actual end product such as a VME system. For each service, SPC produce a skeleton workbook that is completed in conjunction with the customer during the consultancy period. This workbook is then presented as a lasting reference document for the customer; it contains details of any key decisions arrived at jointly by ICL and the customer during the consultancy.

The importance of the workbooks as a lasting impression of a service cannot be overstated and SPC staff are currently considering how the appearance of the workbooks can be significantly enhanced.

Each service is formally documented by a Product Description, which is in the same format as for all ICL's software products. The Product Descriptions are produced by Software and Services marketing staff, not by Services Product Centre staff themselves.

Benefits from packaged services. The provision of services of all kinds to a customer presents many opportunities for the mutual benefit of ICL and its customers. While each individual service has its own recommendations, the following benefits potentially accrue from any service offering:

1. The customer will normally get a *higher utilisation* of his system which has been established using a proven process. The customer is therefore likely to obtain a higher business benefit from his products.
2. The provision of services, especially the continuous 'Keep you going' type (see below) provides an extra *interface* between the customer and ICL, which can provide a channel to facilitate planned growth of the customer's system.
3. ICL gains a double benefit. Obviously the services themselves provide income, but there is also the reduced cost of support. ICLs engineers time becomes more cost-effective (sorting out real problems) because fewer user-induced problems occur on the customers' systems.

To obtain such benefits, ICL obviously has to deliver a *quality* service that conforms to the customer's agreed requirements. Investment in the process is the key method to ensure such quality, hence the critical importance of the contribution of the unseen Services Product Centre staff. We do not, of course, forget that the customer perception of a successful service can hinge very much on the professionalism and expertise of the service deliverer.

2.3 The Services Marketplace

VME installations can be grouped into the following three broad classes:

1. *High VME experience:* Customers with large multi-system operations typically supporting complex networks.

   Customers in this segment are working on innovative developments, which frequently extend the capability of existing products and are the originators of many developments undertaken by ICL. Their systems are usually highly stressed and they are willing to invest in a support infrastructure for their own operation.

2. *Medium VME experience:* Generally customers with mixed production and development workloads, but without high levels of complexity and innovation.

   Systems are consequently not as stressed as in the segment above, but special considerations often have to be catered for, such as high security.

3. *Low or zero VME experience.* This is the most competitive and cost-conscious segment of the mainframes marketplace. Customers typically run a single system with a stable workload. They have little inclination to invest in support costs nor do they see why they should have to.

Three categories of service have been identified. These are:

1. *Get You Started* (Professional services). Services designed to help the customer size, select and plan his system, and to provide a well engineered VME base system on which the customer can build successfully.

2. *Keep You Going* (Hardware and software services). Services designed to remove from the customer the distractions of managing and maintaining his system, to allow him to concentrate on deriving the benefit from his investment.

3. *Take You Further* (Professional and software services). Services designed to help the customer exploit his system to the full, deriving business growth for both the customer and ICL in a controlled, planned manner.

The table below maps these service products on to the customer segments; the entries in bold type are the subjects of the examples described next.

Table 2 Services and Information for VME users

VME Segments	Service category needs	Service Products	Information Products
High VME Experience	Keep you going	**Central System Teleservice**	Diagnostic Guide
	Take you further		RSI optional manuals
Medium VME Experience	Keep you going	System Teleservice and specialised variants	**SEE Guide System Management**
	Take you further		System Exploitation System Security
Low VME Experience	Get you started	**Startup Service** SAMP	'Ease of Use Guides'
	Keep you going	Super 39 Service	**SEE Guide** Prompts Guide

2.4 Examples of services

Two very different services are now described. One is a service currently under development, a 'Keep you going' service targetted at a High VME experience segment; the other is the generally available Series 39 VME Startup Service, which is a 'Get you started' service offered primarily for Low VME experience users. I hope that my choice of services from the opposite ends of the spectrum will give readers some appreciation of the scope of the services arena.

Both services are described under the headings:

Overview
Description – what the customer receives
Process – how it is achieved
Components – deliverables from Services Product Centre.

2.4.1 System Teleservice. The Series 39 maintenance strategy for equipment service and software error correction is based on the use of teleservice for both call-logging and diagnosis. The support process allows customers to make a direct telecommunications connection 24 hours a day, 7 days a week to an automated call-logging system to register their problem. As soon as the problem is registered, a database of Known Errors and Repairs is scanned for a match. The process is fully automatic and customers can dial in for each problem, or collate problems and dial in at a time convenient to themselves. This process is fully described in the paper by Allison in this issue.[2]

The system that controls the administration of the process is known as *Request Manager*. Each Customer Service branch is being equipped with terminals connected to this, so that problems can be viewed by the branches as soon as they are logged. Problems go through a number of stages after they have been logged, before they are closed. Each stage is identified in Request Manager as the problem progresses, and Customer Service branches can monitor this progress. All Customer Service desks and Support Centres are on-line to this ICL system.

2.4.2 Central System Teleservice. The development of a centrally controlled System Teleservice (CST) is an added dimension to the Teleservice operation. With Central system Teleservice a customer version of Request Manager is used by the customer himself to maintain central control of all his systems and so provide a single interface to ICL. Such a product is not yet

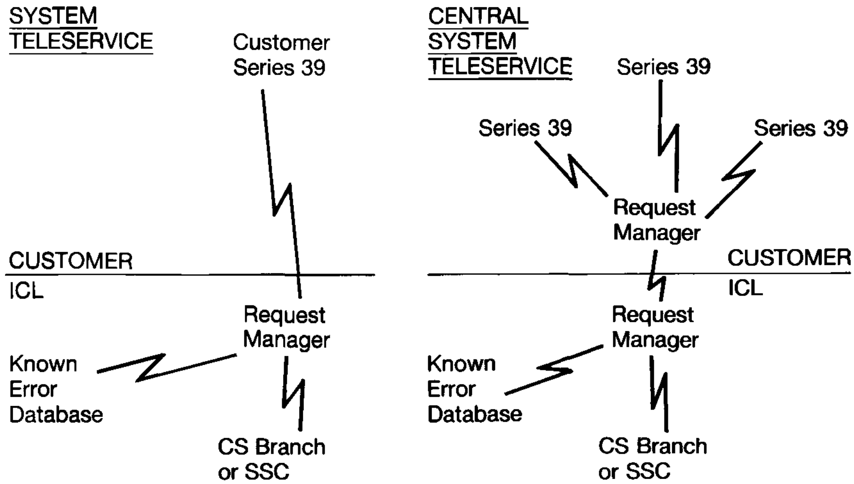


Diagram B

available from ICL but is in its late stages of development in Services Product Centre. Diagrammatically, the difference is as shown in Diagram B.

Description.
Central System Teleservice (CST) will facilitate the support-management of a number of Series 39 systems (and potentially others) from a single location on the customer site.

It will provide all the facilities necessary to control and co-ordinate the service and support aspects of a system.

The service requires the establishment of a 'Service Desk' within the customer organisation to act as a focal point for communications concerning support for the customers' end-user community. This is then seen as a valuable service offered by the customers' computing systems department to their customers.

The strategy for Central System Teleservice introduces processes which facilitate central management of service actions, particularly in the area of:

- Problem recognition and reporting
- Central help desk
- Co-ordination of problem resolution activities
- Problem tracking, retaining historical records
- Problem allocation to specific organisations or people
- Statistical analysis and reporting on service performance.

The service package provides for all the necessary pre-delivery assessment and consultancy, as well as assistance with installation and ongoing support.

The introduction of CST into an existing organisation creates the possibility of major organisational change. Management consultancy is therefore equally as important as technical consultancy. The analysis of the customer's required support infrastructure can lead to change of geographical location and functional distribution of both systems and manpower. The intention is for ICL consultants provide information to assist the customer who has to make decisions on any organisational change required to establish a tailored system.

The package also includes training for the customer staff, who will be involved with the service: operators, technical support staff, system management and DP manager. The training is designed to ensure that the customer's staff understand the service functions for which they are responsible and the support interfaces with ICL.

Following the realisation of the full operational capability, post customer handover, the service will provide for ongoing, regular review meetings with the customer to discuss: problems, future service developments as well as advice and guidance of the operation and daily use of the service product(s).

Normal product support is also provided including bug fixing and preventative maintenance.

The Process.
A full description of the processes involved in System Teleservice and Central Teleservice is given in the paper by Allison already referred to; the section that follows describes the software produced in Services Product Centre to facilitate the provision of these services,

Components.
Figure 2 shows a top view of the system. The shaded areas are the components produced by Services Product Centre in the Systems Management Development Project.

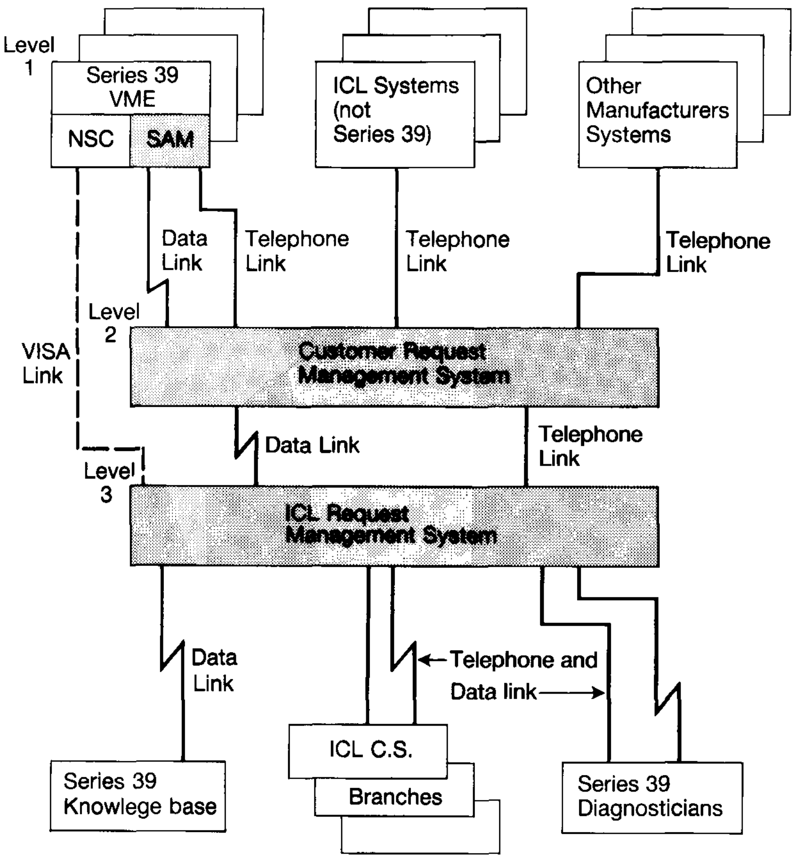


Fig. 2 Central System Teleservice: Top view

The major components are:

- SAM ("Support and Maintenance")
- Request Manager (SSMS) on the customer's site
- ICL Request Management System.

SAM. SAM software is provided for all Series 39 customers as part of System Teleservice. The changes required for Central System Teleservice are mainly procedural, although product tailoring may be required to support the customer's required options. The SAM software product is specific to Series 39 – that is, it is not available on other ICL systems.

SAM runs in the customer's machine and continually monitors the behaviour of that system. Should it detect a potential problem it aggregates data relating to the incident and generates a "fingerprint", which is a coded representation of the incident. The fingerprint may be the accumulation of several incidents that have been monitored and for which data has been collected prior to the problem becoming visible to the customer. The number of incidents that are allowed before a problem becomes visible is determined by a threshold, preset for each particular type.

SAM also enables a user to input manually the details of any malfunctions recognised by customer staff. The customer is guided in the level of information required by product-specific menus that ask for certain fault identifiers, which he may have access to at the time the fault is manifest. The result of this dialogue may be the raising of a fingerprint or the attachment of additional information to an existing fingerprint. The generation of the fingerprint gives rise to a request for action, known as a Maintenance Service Request.

Request manager or SSMS. SSMS (Sam Service Management System) as provided for Central System Teleservice is a derivative of the software provided for the ICL Support Centre. In the configuration provided for CST, SSMS contains two major components:

SRCS is responsible for managing electronic communications between SSMS(s) and SAM servers.

It logs all transactions between SAM servers, other SSMSs and its parent SSMS in Serial files known as Request Transaction Logs (RTLs). There is one RTL for each external SAM or SSMS.

SRMS is provided to enable the logging and progression of Service Requests. It comprises an IDMSX database with update and retrieval applications.

SRMS reads the RTL files produced by SRCS so that all electronically received requests are automatically logged in the SRMS database. Manual logging of requests into the SRMS database is also provided for. Comprehensive database search and update capabilities are provided.

Progress in servicing a customer request may be recorded manually, but it is automatically logged at several progression stages. All types of Maintenance Service Request received (e.g. Fingerprint Lookup Requests, Support Requests, Corrective Action Requests) are logged, as are the responses to them. Such logging and progress recording provides the users of SRMS with a comprehensive Request handling service. Visibility of problems and their progression is given from inception through to solution and clearance.

ICL Request Management System. This contains the same products as the customer system but in addition has access to ICL's Series 39 Knowledge Base which contains known errors and relevant repairs. This database is built up with *predicted* information for hardware components, whereas information on software problems is entered as a result of *experience.*

2.4.3 Series 39 VME Start-up Service. The outline description of this service, taken from the Product Description[3] is as follows:

> This service provides for the start-up of customers moving to Series 39 VME. The service comprises formal training for a nominated member of the customer's staff, the production of an agreed viable VME System Build Plan and the provision of an operational VME Service built according to the System Build Plan.
>
> Provision of the Start-up Service is dependent upon the customer having contracted with ICL for the installation and test of the customer's Series 39 equipment.

There is a 'sister' service provided specifically for VME/CME* customers who are upgrading from an ME29 system and wish to emulate the TME operating system. Both services are clearly targetted at the low VME experience user. They have been developed by SPC in such a way that ICL consultants worldwide can professionally deliver a standard service to all customers in this target market segment.

Description.
From Start-up Service the customer receives a significant amount of *training*; he is provided with a viable *System Build Plan* and an *operational VME system* built to his requirements in accordance with the system build plan. The system is developed to the stage where the development of applications programs or other services can reasonably begin.

The training consists of a series of formal training modules for a selected member of the customer's staff, covering all the fundamental topics needed by a system manager who is responsible for running a VME service on a Series 39 system. A Training Consultant is nominated for each instance of Start-up Service, who provides specialist advice and support to enable the customer to get maximum benefit from his allocated training period.

To maximise the benefits to the customer of the formal training, this is normally carried out before the planning for a system build is begun. Frequently, therefore, the training precedes the delivery of the actual system and the consultancy. However, during the consultancy period the customer receives further training via demonstrations, tutorials and appropriate documentation on all the key activities that he will need to undertake when he is responsible for provision of the VME service to his end users.

The system build plan is produced in conjunction with the customer's staff including, of course, the member who attends the training course. The plan identifies the work to be done jointly by the ICL consultant and the customer to establish a suitably tailored system. The plan itself, like the training, is usually completed before the delivery of the Series 39 system. For these who know VME, the plan includes decisions that are taken in the following critical areas:

- Hardware Cataloguing
- User structure
- System and User Filestore requirements
- Workmix definition
- Filestore Backup methods
- System Support and Maintenance
- System Control Language (SCL) standards.

The operational system. Once the hardware has been installed on the customer site and an initial VME system established, work on the provision of an operational, tailored VME system can begin.

All the system build activities are documented in the Workbook, which is handed over to the customer on completion of the service. The Workbook will contain a reference definition of the customer's system, itemising all system design decisions and their justifications.

If the customer has requested ICL to maintain his equipment, perhaps by System Teleservice, then the Start-up Service additionally provides demonstrations and tutorials on support procedures:

- The initiation and testing of the Telediagnosis and Telesupport routes with the appropriate ICL Support Centre; the provision of procedures for ongoing use of Telediagnosis
- Customer's Servicing and Problem Determination procedures
- Procedures and information necessary for the customer to contact ICL with requests for service.

Process.
The process by which the consultant is able to deliver a satisfactory service to the customer is documented in the Service Guide provided by SPC. ICL consultants who are asked to deliver the service normally attend the relevant

service training workshop. However, it is SPC's intention to enable this service to be delivered by less experienced staff than are required by other services. The service has been designed in such a way that by following the procedures in the Service Guide, ICL consultants in all countries can deliver a quality service in a standard, professional manner.

The basic skills required by a potential consultant are documented for Customer Service unit managers in the Service Planning Guide for the Start-up Service, which is produced by the Systems Introduction team of Services Product Centre.

The process for delivery of the Start-up service is divided into three phases: the Planning Phase – the Installation Phase – the Starter Phase. The objectives of the respective phases are as follows.

The planning phase. For the consultant to familiarise himself with the customer's requirements and to map these onto a system build plan. The consultant also has to formulate and agree a plan for conducting the remaining phases of the service. Typical duration of this phase is 10 days over a period not exceeding 3 elapsed weeks. The task-checklist in the Service Guide identifies 22 separate tasks for this phase. Each task is described in the Service Guide and, frequently, references are made to the workbook which contains technical information that remains with the customer after the service.

The installation phase. To establish a working base VME system from which the customer's tailored system can be produced. The major activities of this phase are:

- Installation and commissioning of the hardware
- Installation of VME system software
- Extension of the hardware and software configuration
- Optionally, installation of the System Administration Prompter.

Typical duration of this phase is 3–4 days.

The starter phase. To establish a customer usable VME system (with CME* where required). During this phase the customer receives advice and guidance on VME usage and facilities by means of demonstrations, tutorials and discussion documents. On completion of the phase, the Workbook of design decisions, build documentation and proformas is assembled for handover to the customer.

Typical duration of this phase is 8–10 days.

Components.
The deliverable components of the service produced by Services Product Centre have all been mentioned un the text so far. In summary they are:

- *Product description* (produced by Software and Services Marketing): a legal requirement for any contract
- *Service Planning Guide:* contains information for Customer Service units worldwide explaining the marketing strategy for services and giving guidance on the resources needed to be deployed
- *VME/CME* Start-up Service Guide:* defines the process
- *VME/CME* Start-up Service Consultants Workbook:* contains reference information about VME and documents the design and build process
- *Training Workshop:* familiarises consultants with the Guide and Workbook and describes the process in detail.

3 Information

The production and issue of information is the second of the three related disciplines (Services, Information and Training) that is the responsibility of the Services Product Centre. The range of information needed to support a family of computers such as the Series 39 is vast. The Customer Information project has from 25–30 staff working primarily on Series 39 documentation: in a typical year 8–10000 pages or screens of text are issued by the project. One team of 7 people is dedicated to support documentation; the information it produces is provided for ICL's own support staff and for customers engaged in the support and maintenance of their own systems.

The following sections indicate the scope of the information required for mainframe systems products and then describe the process by which the documentation is produced.

3.1 Scope of mainframe systems information

At the most fundamental level a range of computers such as Series 39 has an absolute requirement for the following types of information:

1. *Sales* documentation: to sell the product
2. *Legal* documentation: to form a contract
3. *Technical guidance:* to enable customers to use the product
4. *Reference* documentation: to enable experts (e.g. software houses, VME experienced customers) to exploit the product
5. *Support* documentation: to enable customers, ICL and occasionally third-parties to maintain the product
6. *Training* documentation: to enable 'new' users to quickly become competent and to enable experienced users to develop specialist roles
7. *Technical journals* and/or notes for devotees are not, strictly speaking a requirement, but such documents are normally available for technocrats
8. *Sales support* documentation: to assist the salesman who is evaluating a customer's requirements.

None of the above requirements are unique to Series 39 systems but, because of the nature of an operating system, categories 4 and 5 are necessarily more extensive than for an application.

Responsibility for the production of these documents in Mainframe Systems division is summarised in the table below.

Table 3 Mainframe Systems documentation types

TYPE	PRODUCED BY	EXAMPLES
LEGAL	Marketing and Legal Services	Product descriptions
SALES BROCHURES	Business Strategy and Marketing Managers	Series 39 brochures
TECHNICAL GUIDES FOR SOFTWARE AND SERVICES	Services Product Centre	VME: System Management (Series 39)
REFERENCE	Services Product Centre	VME: RSI Manuals. Command Specifications
TRAINING	Training Services	Training Course Manuals
SUPPORT DOCUMENTATION	Services Product Centre	S39 Service Guides. System Event Guide
SALES AIDS	MS Marketing Staff	Series 39 Handbook
JOURNALS AND NOTES FOR DEVOTEES	Individuals	The Story of VME. ICL Tech. Journal papers.

It will be seen that Services Product Centre is the main producer of technical information for Mainframe Systems. The Customer Information Project within SPC works to a strategy which is summarised in the next section.

3.2 Market segmentation and VME documentation architecture.

When a documentation structure is being considered the numbers of customer staff involved in the running of the system and their level of expertise are crucially important. The following paragraphs summarise those attributes of customer installations that have a critical bearing on the level of information that is appropriate.

1. *The small system user:* either a small data processing department, whose purpose is solely to provide a support for a small commercial or professional enterprise, or a satellite station of a very large user of the type described in 3 below. The DP department usually consists of a DP manager, a 'technical expert', a system administrator, from one to five operating staff and up to two – but often none – applications developers.

   This user is typically a recent ME29 user who is used to the smaller scale TME operating system. Such a user may well have used the Start-up Service described earlier to establish his system.

2. *The medium system user:* the traditional data processing department supporting a medium-to-large commercial or scientific organisation with system programming, application development and operating departments each consisting of from 5 to 25 individuals with a fair amount of VME experience.

3. *The large system user:* probably operates a collection of large single and multi-node mainframes with complex networking and comunications systems usually driving smaller satellite systems. The installation supports a large national or international organisation and, in addition to a full complement of system management, programming and operating staff on several sites, also has its own internal group of VME sizing, tuning, development and diagnostic experts. Such a customer is, of course, likely to consider the Central System Teleservices offering described earlier.

Customer Information Project's strategy is to cover the documentation needs of these three bands of VME customer by structuring its products so that graded levels of publication are produced.

For segment 1 there are *task-orientated or tutorial* guides giving step-by-step instruction to new or inexperienced Series 39 small or medium users. These publications are intended mainly to supplement the menu-driven system administration facilities and will satisfy the expected day-to-day needs of the small VME sites.

For setments 2 and 3 there is a suite of *guides and reference manuals* appropriate to professional computer staff; that is, the type of traditional manual currently comprising the bulk of mainframe documentation. These will continue to be produced in paper form but may also be offered in on-line form facilitating CAFS based fast searches in the near future.

In addition, specialist support guides are produced for VME experts on customer sites. A current example is the VME Diagnostic Guide.

The documentation structure is intended to allow the requirements of each type of user to be satisfied by the usage percentages in the table below:

Table 4

		PUBLICATION CLASS		
		Task-orientated Guides	Reference Manuals	Specialist Guides
USER BAND	Small	90%	10%	0%
	Medium	50%	50%	0%
	Large	0%	80%	20%

3.3 The process

I have indicated that Series 39 publications are not produced independently, but that each one is part of a structured suite of information. This section examines the *process* by which individual manuals are produced within this target architecture I shall concentrate on the design process rather than the production process, though I shall refer to the latter where it affects the design process itself: as most people are aware, there have recently been drastic changes to the traditional print production process as a consequence of the availability of "electronic publishing" equipment and techniques.

I believe that the major improvements in computer documentation have come about as a part of a more general change of orientation of computer manufacturers from being technology led to being customer led. This change in culture manifests itself in computer documentation in the form of role or task-oriented guides rather than functional or product-oriented reference texts. The switch in emphasis is from '*how* it works and *what* it does' to 'how *you* (the reader) can use the system to reap the benefits'. The reader, not the product, is the target.

So, how are modern computer-user publications designed? I consider that there are four phases required for the production of instructional text; they will generally overlap but for simplicity, they can be considered here in sequence.

- The research phase
- The creative phase
- The validation phase
- The production phase.

With the advent of electronic publishing, a critical element of the last of these four stages (composition) is performed in parallel with the creative and validation stages and perhaps it should nowadays strictly be labelled the replication stage.

3.3.1 The Research phase. This is the essential pre-design stage. It is concerned with scoping the document (or set of screens for on-line text) that is to be produced. It is a stage of planning and making decisions, decisions about the scope and purpose of the document that are crucial to its success. The author needs to identify who is going to use the document and *exactly* what for.

In a management overview to technical publications standards[4] the pre-design tasks are grouped under the heading of a 'Requirements definition process' and this is exactly what the research phase is concerned with. The key task is to target the publication as carefully as possible onto a particular type of user carrying out a specific set of tasks. For this purpose the researcher must define the target audience, not only in terms of level of expertise that can be expected from the customer (see above on market segmentation), but also in terms of the customer's expectation of the usability of the product.

To give an example of what I mean by the latter point, consider the Series 39 publication "VME: System Event Guide"[5] (I will use this manual, known as the SEE Guide to illustrate several of the points in this section). This publication is designed to be the first reference point for a VME system manager or operator if an unpredicted event occurs. This means that when there is a fault in the system, and the customer's likely expectation is that he should quickly be able to diagnose the problem so that remedial action can be taken: the SEE guide is his first point of reference, usually before SAM is examined. The inference for the design stage of the SEE Guide is that fast accessibility of information is crucial. The reader is unlikely to be in a position to 'browse read' until he finds what he wants, he is more likely to be under some pressure to take remedial action (in itself possibly quite straightforward) fast, because his users are affected. How this requirement is tackled in the case of the SEE Guide is shown in the design section.

The research stage, therefore, needs to determine the type of user and the environment in which the publication will be used. It then has to go a stage further and identify the purpose or objective for which the information is needed. This may require a hierarchic task analysis to be performed, even if it can only be done theoretically at this juncture. The analysis can, and should, be validated at the later validation phase of production.

A level of analysis should be performed on any product, system or service requiring a publication. As I have stated previously, the solution to a customer's business problem may involve the integration of a number of products, or perhaps a number of facilities within an operating system. An example of a task requiring interaction between a number of products would be "Managing a database".

At the research stage any production constraints that will affect the design process should be considered. There are frequently constraints such as the need to produce a publication quickly in time to support the launch of a new product, and budget constraints may limit the freedom of the designer. The potential need for translation and the method of replication are two important considerations. If a document is to be translated it is particularly important to consider the factors which can make accurate translation easier – the obvious one being the extensive use of diagrams when possible. The replication process has to be considered at this early stage. If a document is likely to be photocopied there is little point in the designer using colour for emphasis or highlighting: for example, warning statements printed in red will not show in the photocopied version and it would be much better to use, say, white space to highlight the warning text.

The output from the research stage is likely to be a publications plan, which documents the reasons for a proposed publication's form and stipulates the resources to be allocated to its production. Some plans are, in fact, produced at a slightly later stage and may include a synopsis of a manual. However, the synopsis is properly an early output of the design phase, because it may include a sample layout.

3.3.2 The Creative phase. I consider the creative phase to include essentially the design and first draft process. Key outputs are therefore a synopsis, as mentioned above, and the first draft. The stage from the issue of the first draft for comment, and hopefully usage, onwards is the validation phase.

There are two fundamental design elements of any technical publication, Content and Presentation. The design process involves the following tasks:

Content

- Selecting the information that is appropriate to the needs of the reader (and discarding that which he does not need)
- Writing the content in an appropriate style, for example step by step tutorial or descriptive text.

Presentation

- Presenting the material so that it can be assimilated and applied by the reader
- Structuring the information for ease of access.

For the content, the key task of the technical author is to select and re-orient the source information (usually provided by a very proud designer) so as to make it appropriate to the target reader. It is essential for the content to be *organised* in a way that aids the effectiveness of the text. Chapters are not simply vehicles for splitting up the text into a group of more or less equal numbers of pages. They add more value if they are used to describe a particular function, task or related group of tasks. The organisation of VME System Management[6] is an example of a manual with a carefully structured content.

This manual was created after a hierarchic task analysis process had been performed and the structure reflects the analysis that was done at the research stage. For example, the design of a VME system is split into three main tasks:

STAGE A The design of filestore and the user structure
STAGE B The design of services
STAGE C The design of work scheduling.

The first three 'chapters' of the manual are devoted to each of these generic tasks, and the sections within the chapters describe the component tasks that make up the generic task. An extract from the contents page shows chapter A containing sections:

A1 Designing your user structure
A2 Establishing your basic filestore requirements
A3 Planning your disc usage
A4 Planning your magnetic tape usage

and so on.

An extract from the hierarchic task analysis is given in Fig. 3 and the structure of the manual follows this analysis tree.

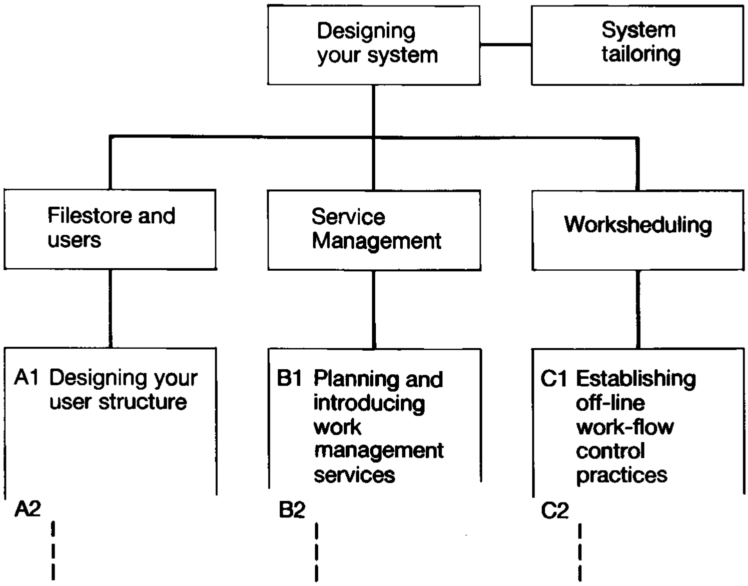


Fig. 3 An extract from the Tasks Analysis of VME System Management

Moving on now to discuss *presentation*, this is the most obvious and most subjective aspect of the design stage to a casual reader: even a well-structured, well-written manual is unlikely to be used, except reluctantly, unless it has an attractive appearance. Good presentation is therefore an essential component of a good manual. It is also difficult to define exactly what is good presentation. As I have said, this element can certainly be subjective. Aspects of presentation include: paper-size, typefaces, space and structure, lists, use of diagrams, colour and many other considerations.

Choice of page size, for example, warrants a whole chapter (and chapter 1 at that) in Dr. James Hartley's classic book "Designing Instructional Text"[7] Hartley points out that the selection of page size limits the later choice of such items as typesize, column arrangements and line length. It should therefore be done positively rather than by default. Hartley considers that the most important factor to consider is how the document is going to be used. An opened A4 manual has quite a large footprint (the area of space consumed by the manual on a desk) and would therefore be inappropriate for use in a cramped environment (for example, an airplane cockpit). But other factors which affect the choice of page size, such as production costs, can outweigh other considerations. Currently ICL's corporate documentation standards allow A4, 2/3 A4 and A5 manual sizes.

Because effective presentation styles require specialist skills to design them, a team of specialists within ICL are working to put together a compendium of existing styles that can be used by authors who only need to select a style that is appropriate to their particular requirements.

One aspect of design that I mentioned earlier is the presentation of information so that it is easy to find. This is necessary for all manuals and essential for some. I indicated that the System Event Guide[5], as a problem identification guide, needed a design that was particularly suited to fast look-up of information. To this end, the SEE Guide, as it is known, contains checklists on pages which open out wider than the normal footprint so that users can see the list when referring to other parts of the guide. And, as a very practical touch, pages that are likely to be removed from the binder frequently (for example, to take over to a peripheral device) are produced in card so as to be particularly durable; also the sections are separated by tab cards. Thus every effort is made to help the reader quickly find the information he needs.

Finally it is worth mentioning that electronic text needs to be subjected to an equally diligent design process. I shall not discuss this aspect here because it would require a paper of its own to do the topic justice. But I will leave you with the thought: if the text is merely reproduced on a screen it takes approximately twenty VDU screens to display the same amount of information that can be presented in a double page spread of an A4 book.

3.3.3 The validation phase. The third part of the process for producing customer information is the *validation* phase, whose purpose is to prove the accuracy and usability of the information: the latter – usability – is now becoming more clearly recognised as an essential part of the validation process. In a lecture arranged by ICL's Human Factors Group, Patricia Wright, whose work for the Medical Research Council has led to a keen interest in documentation, proposed a more 'scientific' approach to validation. She believes that different evaluation techniques are appropriate to each aspect of information. So content can best be assessed by field trials and interviews with users whereas usability can often be assessed by in-house 'laboratory' experiments. Her conclusion is that no one method of validation is acceptable on its own, and the appraisal of documents by experts (for example, the software designers) is seldom sufficient on its own.

John Williams of the ICL Literature and Software Operations has produced a useful checklist[8] that could be used when validating documentation. For each piece of information this recommends that the validator asks the following questions:

- Was it easy to *find?*
- Was it easy to *understand?*
- Was it *sufficient* to solve my problem?
- Was it *all necessary* in order to solve my problem?
- Was it *accurate?*

Our task in Customer Information, and elsewhere in ICL, is to set in motion the procedures necessary to validate our documentation objectively across the whole range of quality criteria. Not just technical accuracy but also: appearance, accessibility, maintainability, applicability, etc. This validation needs to be positive not passive, that is we must encourage field-trial users and other validators to use checklists such as the one above. And we must ensure that feedback obtained is reflected in documentation produced in the future.

Summary and conclusion

In the key markets in which ICL operates, it is essential to have added value offerings to differentiate our mainframe products from those of our competitors. The provision of quality services and appropriate information to our customers is one of the ways in which we can gain a competitive edge. This paper has, hopefully, given some insight into how the Product Centre approaches this task and meets its requirements.

Acknowledgements

Many thanks are due to Trevor Mills for providing the opportunity to write this paper; to Ray Sayers, Ken Hoyle and Nick Donelly for technical assistance on the Services aspects; to Joyce and Elaine for deciphering my handwriting, and to Jack Howlett for sympathetic cajoling.

References

1 YOUNG, J. (1988): The ICL System Support Centre Organisation
2 ALLISON, R. (1988): ICL Series 39 Support Process. ICL Technical Journal, **6** 1, xxx.
3 Product Description; Series 39 VME Start-up Service. ICL MPD/096, Ed 1 (1987).
4 Technical Publications and Standards Guidelines: Management Overview. ICL R06207/01 (1985).
5 VME: System Event Guide. ICL R00483/03 (1987).
6 VME: Systems Management (Series 39). ICL R0466/03 (1987).
7 HARTLEY, J. (1987): Designing Instructional Text. London. Kogan Page.
8 WILLIAMS, J. (1986): How to validate documentation. ICL ASPR040 (1986).

Knowledge engineering as an aid to the system service desks

J.D. Mitcalf
ICL Knowledge Engineering Business Centre, Manchester

Abstract

This paper describes the development of a prototype expert system to assist in initial problem diagnosis in a System Service Desk. The prototype diagnoses faults in the ICL 8801 word processor and was written in ADVISER, the ICL expert system shell. The paper seeks to show the applicability of Knowledge Based Systems in this field, but concludes that the classical rule-based expert system can be problematic when used in this role.

1 Background

Services make a major contribution to ICL's revenue and profit. Increasingly we face fierce competition from OEM suppliers of services, usually attempting to win business in our most profitable areas. In response to the growing need to reduce service costs and improve our image in customer services there is now a real requirement to release human experts from the chore of 'routine' problem identification. This project examined the potential of Knowledge Based Systems (KBS) to the application of problem identification in a System Service Desk.

The project was undertaken against a background of existing work on the use of expert systems in the field of computer system fault diagnosis [6,7,8] which reflects the growing interest by the major computer manufacturers in automatic fault detection. In ICL the DRS range was the first distributed system to incorporate on board local area network diagnosis. More recently extensive automatic fault diagnosis has been incorporated in the Series 39 telediagnostic facility which offers remote access to a dedicated diagnostic co-processor [15]. The provision of on-line fault information in particular represents an opportunity for automatic remote diagnosis which has not been fully exploited. The practicality of this approach has been demonstrated recently in the TXE4A expert system which diagnoses faults in STC telephone exchanges [14].

2 The service desk function

At the introduction of a new product there are inevitably some initial teething troubles and a relatively high number of experts are assigned to its support. This initial few months of product introduction is a demanding time for the experts. Customers are unfamiliar with the new product and there is a real need to distinguish genuine faults from errors of operation or installation. Communication between the experts is particularly important in these early days to ensure that when a new problem is identified it is immediately known to all the relevant service personnel wherever they may be based. Over a period of time it becomes possible to classify problems according to their cause and symptoms. A body of stable knowledge is accumulated which for the most part remains locked inside the heads of the experts involved. An experienced diagnostician can quickly identify a known fault type usually on the basis of very scant information. This is often the point at which experts begin to leave the project to support new and more interesting products, taking their expertise with them. The net effect is that as the product matures and its failure modes become better understood the quality of support does not necessarily improve and may in fact deteriorate if staff turnover is high. For a mature product the proportion of reported problems in the category of 'routine' can be anywhere between 30 and 60% depending on the product.

When a fault is reported to the service desk the call is taken by a Call Receptionist who records the basic details in a database of outstanding calls. Support personnel, known as Desk Specialists, are organised by product and geography. Each Specialist selects from the list of outstanding calls according to priority and may return the customer's call to ask for more details, to give immediate assistance and advice or with an ETA for the visiting engineer. Should a site visit be necessary the call is placed in a list to await allocation to an engineer by a Field Resource Controller together with details of appropriate spares and tools the Desk Specialist has indicated should be taken to site. To assist in the diagnostic process the expert has on-line access to a database of known errors and lists of installed hardware at each of the customer's sites. The average call throughput rate, based on the total handling time for a call on the desk, is recorded and monitored on a daily basis. The usual target is around 10 minutes.

It is considered that the 'call laundering' process provides a number of opportunities for Knowledge Engineering. However it was decided to confine the activities of the pilot project to call reception and initial problem diagnosis. At present the call reception process is responsible solely for recording the details of the fault. A natural extension of this function is to perform early diagnosis of routine faults. In the pilot project an expert system embodying sufficient knowledge to identify routine problems is used to structure the dialogue with the customer and ensure that the maximum of relevant information is obtained. If the problem is sufficiently simple a diagnosis can be made. A comparatively shallow diagnostic knowledge will suffice for an expert system in this role, the major requirement being rapid response to the customer's input. It is important that the customer is not aware of long pauses or perceives the conversation as unduly long. For this reason an arbitrary limit of 6 questions was placed on the dialogue. The potential benefits are that between 30 and 60% of calls are not passed on to the human expert, the customer is given an immediate response to his enquiry in most cases, and the quality of information is improved for those calls which require human diagnosis. A further advantage is that it falls well within the scope of available technology in expert systems and is readily amenable to implementation in ADVISER, the ICL proprietary expert system shell.

3 The model

ADVISER was chosen in the first instance as the expert system shell with which to address the area of initial problem diagnosis. The chosen area of expertise was the 8801 word processor which is a sufficiently compact body of knowledge to allow the development of an expert system in a relatively short time. The project was greatly assisted by the existence of a set of support manuals which had been written specifically to help comparatively unskilled personnel perform routine problem identification. The manuals had proved too slow and difficult to use in practice and had not been a great success, but they represent an existing body of up-to-date knowledge in a form readily transferable to ADVISER rules. The development of the pilot expert system proved to be very rapid, about 2 man months, being largely an exercise in transcription. In the event two versions of the system were possible in the time available, based on the two modes of reasoning, forward and backward chaining.

The forward chaining model was written largely using the DEMON construct of ADVISER whilst usual RULEs were at the heart of the backward chaining model [11]. ADVISER is not primarily a forward chaining shell; it is capable of forward inference but is not generally used in this mode.

The forward chaining approach more closely resembles the format of the manuals and therefore arguably the mode of reasoning used by the experts who wrote them. The model infers forwards from the symptoms using the rules to gradually piece together the facts until a conclusion can be derived. The model has no pre-conceptions as to what the fault might be but eliminates fault possibilities as information is supplied until the inference can proceed no further. Rarely does the inference resolve to a single conclusion.

The backward chaining model attempts to prove a pre-defined hypothesis (goal). At the highest level this may be simply to prove that a fault exists in the system. The high level goal breaks down into sub-goals, for example, that there is a fault in the disk system, the CPU board, or the memory board. The sub-goals can in turn be sub-divided down to the level of specific faults. The goals and sub-goals represent a decision tree, at each node there is an implied decision as to which goal to prove first. If the system fails to prove a given goal it backtracks to the last decision point and tries the next most likely goal until the solution is found.

The two models described showed clearly that a combination of forward and backward chaining is most appropriate. Backward chaining alone leads to a highly structured dialogue entirely under the control of the system. It is difficult to offer information outside the strict sequence of questions generated by the inference process. Potentially a great many redundant questions can be asked as the system explores inappropriate branches of the decision tree.

A purely forward chaining system on the other hand gives a passive interface with little or no dialogue structure in which the customer is left to volunteer the information he considers relevant. There is no impression that the consultation is moving towards a conclusion and in fact there is every possibility that none will be reached should some vital fact be omitted. Clearly this is a poor model of a service desk call in which it is essential to steer the dialogue to obtain a maximum of relevant information and eliminate inconsistent or irrelevant data. For this reason the forward chaining model remained in the laboratory and was never demonstrated to the service desk personnel.

The final prototype employed a combination of forward and backward chaining. The initial mode of inference is forward to establish the sub-system containing the fault. From then on backward chaining strives for a specific diagnosis and drives the consultation. If the initial deduction proves incorrect and a diagnosis can not be made in the chosen sub-system then forward chaining is again tried and if possible another sub-system is found which matches the known symptoms.

Appendix 1 shows the system diagnosing a fault in the display pcb of a DRS 8001 word processor. In the example session the correct sub-system, the video, is selected first and a diagnosis quickly made. In the event that the system could not establish a fault in the video, a second forward inference would be tried to derive a new area of investigation. The user is kept informed of the current focus of attention by a caption at the top left corner of the screen.

Neither of the two prototype models employed fuzzy reasoning. The manuals which were used as the knowledge source are based on a binary tree approach, i.e. a true/false decision is required at each branch. It was felt that the addition of probabilities would do little to aid the diagnostic abilities of the system at the level it would be expected to operate. As a general principle, when the customer is not certain that a given symptom is present it is better to ask that the fact be confirmed rather than record his uncertainty.

The size of the prototype models is approximately 3500 lines of ADVISER source which is equivalent to 70% of the 8801 Laundering Manual.

4 Future

The pilot system described addresses the problem of initial problem diagnosis and is only required to perform fault diagnosis to a comparatively shallow level on behalf of users who are not themselves diagnosticians and do not wish to challenge the advice given. In the future Knowledge Engineering will be of potential benefit to support the human experts. Here the expert system is characterised by a deep understanding of the products and its fault patterns. It will remind the expert of faults encountered in the past having similar symptoms and present a range of possible diagnoses for his appraisal [3]. The expert system adopts a subservient role to the expert who retains the authority to make the final decision. A key feature of this type of system is the co-operative nature of the relationship between man and machine. To take full advantage of this the expert must have immediate update access to the knowledge base when a new fault is identified or more ambitiously when the expert system encounters new information it updates the knowledge base automatically [1,9]. There must also be a move towards a centralised knowledge base accepting the contributions of all the available experts. The prime benefit is uniformity of diagnostic expertise, all the human experts are up to date with the current fault level. It is envisaged that ultimately this will lead to greater consistency of fault diagnosis and a consequent reduction in unnecessary site visits.

Expert systems of this nature are still in the realm of research, mostly in the medical field [4,5] and more generally are part of a growing body of research concerned with decision support expert systems. Despite the abundance of work in this field the problems are such that it is unlikely that fully operational systems will be available in the immediate future.

5 Conclusions: the next phase

In so far as the prototypes were intended to demonstrate the feasibility of using KBS as part of the function of a Support Desk the project was a complete success. However the classical rule-based system, even though richly endowed with both forward and backward chaining facilities, was not well suited to the task of developing an operational system. The reasons for this decision reflect the limitations of generic rule-based expert systems in a diagnostic role and can be summarised as follows:

- Response time, although acceptable for the majority of applications, was not adequate in this case. An unusually fast response is needed which is perceived by the user as near instantaneous. Even before the prototype was written it was considered unlikely that this requirement could be met.
- The user interface offered by the prototype was felt to involve too much typing for a Call Receptionist. A significant outcome of the pilot is the

decision to move away from keyboard input in the next phase of the project and use mouse selection accompanied by pictorial displays.
- Maintenance of a rule based system is problematic where many experts are involved all of whom require update access to a large model.

To combat some of these problems the next phase of the project will be based on a development of GUIDE from Kent University to be called LOCATOR. The system is implemented on a Sun workstation and will be given an operational trial at selected Service Desks in the Spring of 1988. The interface is superficially quite similar to that of ADVISER in that it consists of a series of menus. There are two important differences; mouse selection of menu options, which considerably improves the system's ease and speed of use and the inclusion of digitised drawings and photographs in question text. LOCATOR will address the HCI limitations of the prototype but maintainability is still likely to be a problem. In the long term a move to intelligent access to a database of fault symptoms, similar to the ACE expert system [2], is the only real solution.

Acknowledgments

I should like to thank George Rouse who contributed to the writing of this paper, Mark Poley and Neil Davis who were instrumental in setting up the pilot project. Thanks also go to Lou Hang Kong seconded from Singapore University who wrote all the models and whose industrious services were essential to the success of this project.

References

1 GINSBERG, A., WEISS, S. M., POLITAKIS, P.: 'SEEK2: a generalized approach to automatic knowledge base refinement'. Research report of Rutgers Univ. Dept. Comp. Sci., New Brunswick, NJ, Report No. CBM-TR-148.
2 VESONDER, G.T., STOLFO, S.J., ZIELINSKI, J.E. MILLER, F.D., COPP, D.H.: 'ACE: an expert system for telephone cable maintenance', in Proceedings of IJCAI-83, Los Altos, CA, Kaufman 1983.
3 RECTOR, A.L., NEWTON, P.D., MARSDEN, P.H.: 'What kind of system does an expert need?', in People and computers: designing the interfaces 1985, Johnson, R., Cook, S. (eds.), Cambridge U.P. for the BCS.
4 DODSON, D.C., HARRISON, M.J., RECTOR, A.L.: 'A prototype knowledge-based medical treatment planner', in Expert Systems 83, Churchill College, Cambridge, BCS Expert Systems Group.
5 FOX, J., MYERS, C.D., GREAVES, M.P., PEGRAM, S.M.: 'A systematic study of knowledge base refinement in the diagnosis of leukemia', in 'Knowledge Acquisition for Expert Systems: a practical handbook', Kidd, A.L. (ed.), New York, Plenum Press.
6 LATHAM, B., SWARTWOUT, M.W.: 'CDx – crash diagnostician for VMS', in Expert Systems and Knowledge Engineering 1986, North Holland.
7 WILSON, M.D., PAVELIN, C.: 'Experience with acquisition for expert system in construction', in Proceedings of SERC workshop on knowledge acquisition, Abingdon, July 1987, Report No. RAL-87-055.
8 ROLSTON, D.W.: 'A multiparadigm knowledge-based system for diagnosis of large mainframe peripherals', in Proceedings of the third conference on Artificial Intelligence applications 1987, Kissmmee, Florida, IEE Computer Society Press.
9 PAZZANI, M.: 'Refining the knowledge base of a diagnostic expert system: an

application of failure-driven learning', in Proceedings of AAAI-86 1986, Fifth National Conference on Artificial Intelligence, Philadelphia, AAAI, Vol. 2, Engineering.
10 CIN, M.D.: 'Distributed diagnosis for computing networks', in Microprocessing and microprogramming 1984, 14 (3,4), 139–144.
11 'ADVISER User Manual', ICL Publication R30008/02 1985.
12 BARTLETT, C.W.: 'S39XC', in ICLTJ, Vol. 5, Issue 3, May 1987.
13 SMALL, M.: 'The application of knowledge based systems to computer capacity management', in ICLTJ, Vol. 5, Issue 3, May 1987.
14 THANDAFFERI, M.: 'Expert systems application for TXE4A exchanges', in 'Electrical Communication, the technical journal of ITT', Vol. 60, No. 2, 'Expert Systems Technology'.

Appendix 1 – Operation

The system is based on keyword selection. Initially the customer is asked to describe the fault in general terms while the receptionist listens for one of a possible set of keywords. The receptionist selects the appropriate option from a complete set of keywords in the main menu. The keywords have been defined by investigating the most common words and phrases used by customers when describing fault symptoms. There is often one or more keywords for each of the possible failing sub-systems in the product. The customer is then asked a series of increasingly more specific questions. At each step the receptionist chooses the most closely matching keyword or phrase from a menu. The dialogue proceeds until the system either profers a diagnosis or logs the call.

Logic analysers for system problem solving

Brian Parker
Consultant, ICL UK HQ Technical Support, Stevenage, Hertfordshire

Abstract

Logic analysers are powerful, sophisticated digital test and measuring devices that enable the activities in a computer system to be investigated at the hardware function level. Commercially produced portable instruments have been available since 1978 and have been used since then by ICL Customer Services for investigating problems that are intractable by less powerful methods. This paper seeks to indicate the environment in which they are used and some of the techniques that have been developed for problem solving with their aid, illustrating this with examples of real-life applications.

1 Introduction

Every computer systems supplier has the occasional problem which proves exceptionally difficult to resolve. These problems fall into a number of categories but one frequently occurring feature is that the problem is of intermittent nature and cannot be easily provoked.

A number of techniques have been developed to isolate and resolve these problems, such as both hardware and software tracing and software dumping and analysis. The logic analyser has been developed to complement these techniques, and provides very powerful fault location facilities where other methods are inadequate.

Since a piece of test equipment must have a performance in excess of any unit to be diagnosed, the logic analyser often represents the 'state of the art' for the industry. Instruments currently available are the result of up to four 'generations' of evolution in the space of less than ten years, and contain the very latest that technology can offer.

2 Problem types

Applying a logic analyser to a problem is rarely an easy task. It requires detailed knowledge of both hardware and software, and physical access to the internal workings of the system. It follows therefore that they will be used only after the normal diagnosis methods, e.g. pcb replacement, dump analysis, etc., have failed to resolve the problem. By implication, this often means that the problem is critical for the user.

Several different problem types fall into this category:

2.1 Maintenance faults

This means faults in a unit or system that was functioning correctly but is not doing so now. This type of problem is usually, but not exclusively, associated with hardware failures. These may be genuine intermittent hardware faults, or can be solid failures which require a unique and infrequent set of conditions to become evident. In these circumstances, software traces and dump analysis are often misleading, or unable to provide the resolution required.

2.2 Hardware design faults

Despite design simulation, extensive validation and live testing it is not unknown for the odd design fault to show up after a product is released and in use. To slip through the quality control net means that these design faults require a rare and very specific set of circumstances, or precise timing of events to provoke the failure.

Initially this type of problem will be treated as a maintenance fault, using the procedures of software dump analysis and part replacement. However, the software dump only shows the state of the system at the time the dump was taken, and will not show any simultaneity conditions or timing at the hardware level which are often the key to this type of fault.

2.3 Software design faults

Again validation will usually eliminate the more obvious faults, so those that occur after general release are often obscure. Very often the failure will be associated with a particular customer's workload, frequently under maximum load conditions. Since any hardware error detection built into the system can rarely check for software functionality errors, it may be many thousands of instructions later before the original fault causes a failure.

Here the normal technique of dump analysis is at a severe disadvantage, as much of the original trace information will have been overwritten before the error is apparent. Hence isolating the faulty conditions and code can be very difficult. A classic case is the identification of code and/or data corruption.

2.4 Demarcation disputes

These are problems where two separate but connected modules fail to complete the required action. The suppliers of the modules may be different companies, or even different divisions within the same company. The common factor is that each is innocent until proven guilty; this is particularly true when the module is established and well known.

Very often the underlying cause of the failure is in interpretation of the interface specification at the very detailed implementation level. Software tracing is inadequate since it does not provide the details of exactly what crossed the interface, with associated timings. To get the problem resolved it is necessary not only to identify which module is in error, but also to be able to give the reason why in sufficient detail for a correction to be made.

3 Logic analyser functions

Over the recent years the facilities in logic analysers have been developed and enhanced in response to feedback and statements of requirements from users, but the main principles of operation have remained much the same. These fall into four main groups, with most commercial instruments providing the opportunity to mix two or more of these functions together.

3.1 The 'state' analyser

The first commercially available instruments provided only 'state' analysis.

The state analyser provides facilities for the capture of sequences of events (or 'states'), based on timing (or 'clocks') from the system under test (SUT), so offering a hardware trace facility. Figure 1 gives a block diagram.

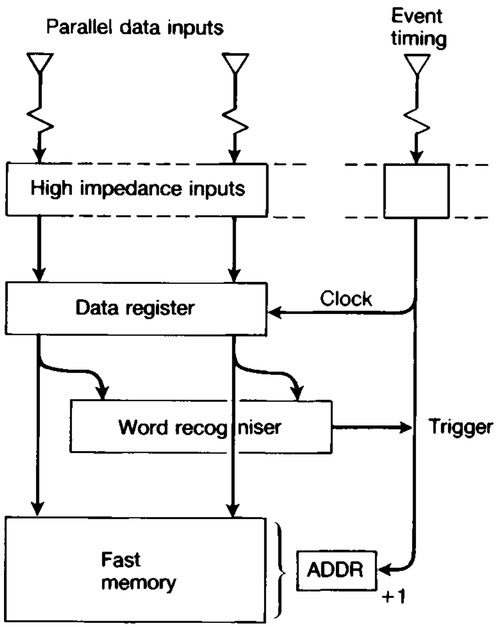


Fig. 1 The state analyser

Data is taken into the analyser in parallel via active inputs which present a high impedance to the SUT. These inputs are clocked into a data register using a clock edge also derived from the SUT; in a simple case the data inputs would be the bus data lines and the clock edge would be the bus strobe. The contents of the data register are then written away into a cyclic high speed memory, with each new clock edge incrementing the address. Microprocessor control and a video screen provide set-up and data-viewing facilities.

The key feature of the logic analyser is that if the user puts a data pattern into the WORD RECOGNISER register the instrument can detect the occurrence of this pattern across all identified input channels and use this recognition to trigger the start or end of the data capture.

The first commercially available instruments provided only simple triggering from a single state. Among the later developments, advanced instruments now provide triggering from a programmed sequence of events, data qualification and interval time measurement.

3.2 The 'timing' analyser

The state analyser is unsuitable for the monitoring of asynchronous events, so a technique was introduced to produce waveform-like displays from a 'timing' analyser. Figure 2 gives a block diagram. Data is collected via high

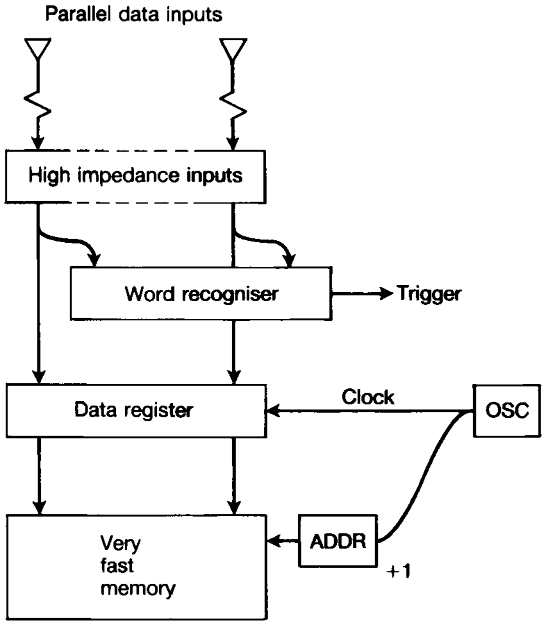


Fig. 2 The timing analyser

impedance inputs similar to the state analyser, but now the data register is clocked not from the system under test but from a free running crystal oscillator within the instrument; in most instruments this oscillator clocking can be set-up to give an interval range from 10 nanoseconds to several milliseconds.

For triggering, the word recogniser must now recognise asynchronous events and so looks at the data inputs prior to latching in the data register. The display, shown in Fig. 3, is a digital representation of the analogue original, with the level changes timed by the internal instrument clock. The timing accuracy of the display is therefore ± 1 clock time with respect to the original waveform. The facilities developed include 'glitch catching' (see para. 4.2.2 below) and specification of the duration of the trigger condition; this makes the timing analyser a very valuable tool, and it is widely used in hardware development.

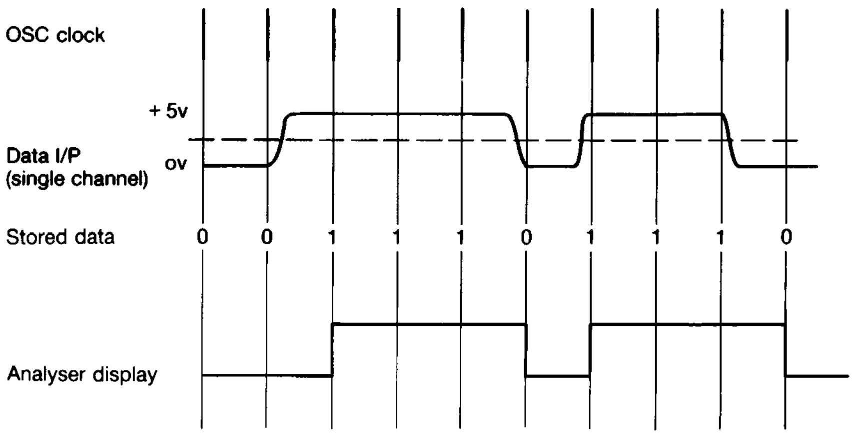


Fig. 3 Timing display

3.3 The analogue inputs

An additional feature available in recent instruments is the analogue input, in simple terms a digital storage oscilloscope input, usually with two channels provided. Since it can be linked to state and/or timing sections of the same instrument, with their powerful triggering facilities, it provides single shot analogue tracing for very complex events. For example the noise glitch which causes a fail once a day can now be examined in detail to determine the cause.

Maximum sample rates for the analogue-digital converter (ADC) can range from 100M to 400M samples/sec, so even very short duration pulses can be recorded.

3.4 Microprocessor code tracing

One of the original design objectives for the logic analyser was to provide a tracing tool to aid the development of microprocessor software. Continuing development has enhanced this capability and most current instruments offer facilities to cover a wide range of microprocessors.

As microprocessors have become more complex, with characteristics such as instruction pipelining, logic analyser developments have provided increasingly sophisticated facilities to support them. The instruments we now use include pre-processors, which sort out the obeyed instructions from those in the queue, and software to disassemble the hexadecimal trace information back to the manufacturer's mnemonics as used by the programmer.

The instruments still have their very powerful triggering facilities, which enable traces to be collected on unique paths through code.

4 Logic analysers in the Customer Service environment

Since our first use in 1978 of the early logic analysers we have, within our Customer Service operation, built up considerable experience in problem resolution using these instruments. A few examples of problems is perhaps the best way to demonstrate how we have been able to apply the basic facilities provided by the instrument manufacturers. By continued feedback to various manufacturers we have been provided with enhanced features to assist in our problem solving activities.

More recent instruments have both local storage and hard copy output, which allows some actual live traces to be used as examples.

4.1 The state analyser

One typical use for the state analyser is in the identification of store or data corruptions. This is a not uncommon type of problem, but is particularly difficult to identify without the logic analyser as the resulting failure usually occurs long after the corruption has taken place. Provided sufficient is known about the nature of the failure to enable a unique trigger condition to be specified for the logic analyser, then identification of the cause is virtually certain.

This first example shows a problem with 2900 series mainframes. The symptoms were single word corruptions in both operating system code and data areas. Dump analysis showed that the corruption data was always the same, and the address corrupted was always at the same offset within a page, although the page location was variable.

To help clarify the way in which the analyser was used, a few notes on the 2900 series architecture may be helpful. The main interconnection highway for the 2900 mainframe (the SCU) connects together processors (OCPs), peripheral controllers and mainstore modules. Since both OCPs and peripheral controllers have write access to the mainstore, and the source of the corruption was not known, this highway was chosen as a suitable monitoring point for the logic analyser. Data was 'clocked' into the analyser using the highway strobe, so that each line of the trace represents one transfer across the highway.

Label>	SMN	DMN	ADDRS	FUNC	BWR00	DATW0	Time	
Base >	[OCT]	[OCT]	[HEX]	[BIN]	[BIN]	[HEX]	[Relative	
[Mark]	03	06	XXXXXX	XXX	X	XXXXXXXX		
-0020	06	00	0C1114	100	1	32DEBA00	4.640 us	
-0019	00	06	000006	110	1	32DE3A00	320.0 ns	
-0018	06	00	0FCA40	000	0	32DEBA00	3.360 us	
-0017	00	06	000002	010	1	00000000	920.0 ns	
-0016	06	00	0FCD90	000	0	32DEBA00	2.080 us	
-0015	00	06	000002	010	1	00000590	920.0 ns	
-0014	06	00	0FD324	100	0	14400000	1.640 us	
-0013	00	06	000006	110	1	00000590	320.0 ns	
-0012	06	00	0FCAA8	000	0	14400000	760.0 ns	
-0011	00	06	000002	010	1	0000FB00	920.0 ns	
-0010	06	00	0FCAAC	100	1	0000FB00	1.760 us	
-0009	00	06	000006	110	1	0000FB00	320.0 ns	
-0008	06	00	0FC998	000	0	0000FB00	1.640 us	
-0007	00	06	000002	010	1	00000000	1.240 us	
-0006	06	00	0FCB70	000	0	00000000	2.240 us	
-0005	00	06	000002	010	1	00000040	920.0 ns	
-0004	06	00	0FCB74	100	0	00000001	440.0 ns	
-0003	00	06	000006	110	1	00000040	320.0 ns	
-0002	06	00	0FCB00	000	0	000FCB0F	400.0 ns	
-0001	00	06	000002	010	1	19640002	1.240 us	
+0000	06	00	1D9DCC	100	1	01940000	1.800 us	Corruption
+0001	00	06	000006	110	1	19640002	320.0 ns	
+0002	06	00	0FCB18	000	0	000FCB0F	440.0 ns	
+0003	00	06	000002	010	1	00640000	1.200 us	
+0004	06	00	0FCB38	000	0	01940000	3.680 us	
+0005	00	06	000002	010	1	00000400	1.000 us	
+0006	06	00	0FC9E0	000	0	01940000	2.240 us	
+0007	00	06	000002	010	1	00000000	920.0 ns	
+0008	06	00	0FCB3C	100	0	7FEFF4BC	2.040 us	
+0009	00	06	000006	110	1	00000000	280.0 ns	

Fig. 4 Identification of corruptions

Figure 4 is the analyser output from which the problem was resolved. SMN and DMN are the source and destination addresses respectively for the highway transfer. The FUNC column is the transfer function, where '100' = write, '000' = read. The trigger point at line +0000 shows the corruption taking place, i.e. the corrupting data #01940000 (column DATW0) being written (FUNC = 100) to a store address with a page offset of #1CC (ADDRS = 1D9DCC) with a source module address of 06, identifying an OCP as the source of the corrupting transfer.

Analysis of the sequence of store accesses made by the offending OCP prior to the corruption, combined with the timing of these transfers, enabled a fault to be identified in the OCP microcode. The microcode in error was handling a disc transfer which required alternate track addressing, so was used only infrequently.

4.2 The timing analyser

4.2.1 Detection of a spurious pulse. The first example of a 'timing' application is also from the 2900 series SCU highway. Although a comparatively simple fault, it gave symptoms which did not suggest the actual cause, resulting in the wrong parts being replaced. All the symptoms pointed to a 'hang-up' during a store access within a mainstore module. Again, a brief explanation of the transfer mechanism for the SCU highway may be helpful.

The SCU transfer timing is shown in Fig. 5a. When any module requires to transfer data across the highway it first raises a request (see QNNR4 in the waveform diagram). When the highway is free the requesting module will receive a 'select' (see QNNS4) and will then put address, data, function and destination module information on the bus.

After allowing time for data to stabilise, a strobe is provided to clock the bus data into the receiving module (see 'QNNT'). If the receiving module can accept the data (i.e. its input buffers are free), it will return an 'accept' to the source module (see QNCA 0) which will reset the request. If the source module does not receive an 'accept', it must maintain its request line until granted another select during which 'accept' is received.

The fault is shown in Fig. 5b to be a very short duration pulse (approx. 10 nS wide) occasionally appearing on the QNCA 0 line, coincident with the front edge of the select line. This extra pulse was resetting the original request whether or not the transfer was accepted by the destination module. Although the extra pulse appeared several times per day, the system failed only if the destination module could not accept the transfer, which was much less often.

The source of the extra pulse was traced to a hardware fault in another module (peripheral controller) which was not being addressed at that time.

4.2.2 'Glitch' detection. This second example of the timing analyser application shows the use of 'glitch' detection: a 'glitch' is a pulse which occurs between analyser sample times. Most instruments will detect glitches down to 5 nS wide.

The problem is from a 2900 series high speed printer fitted with dual interfaces, A and B say. While being used on the B interface, it would occasionally switch to A and become lost to the original system. This was very inconvenient as the fault seemed to occur most often during long print runs.

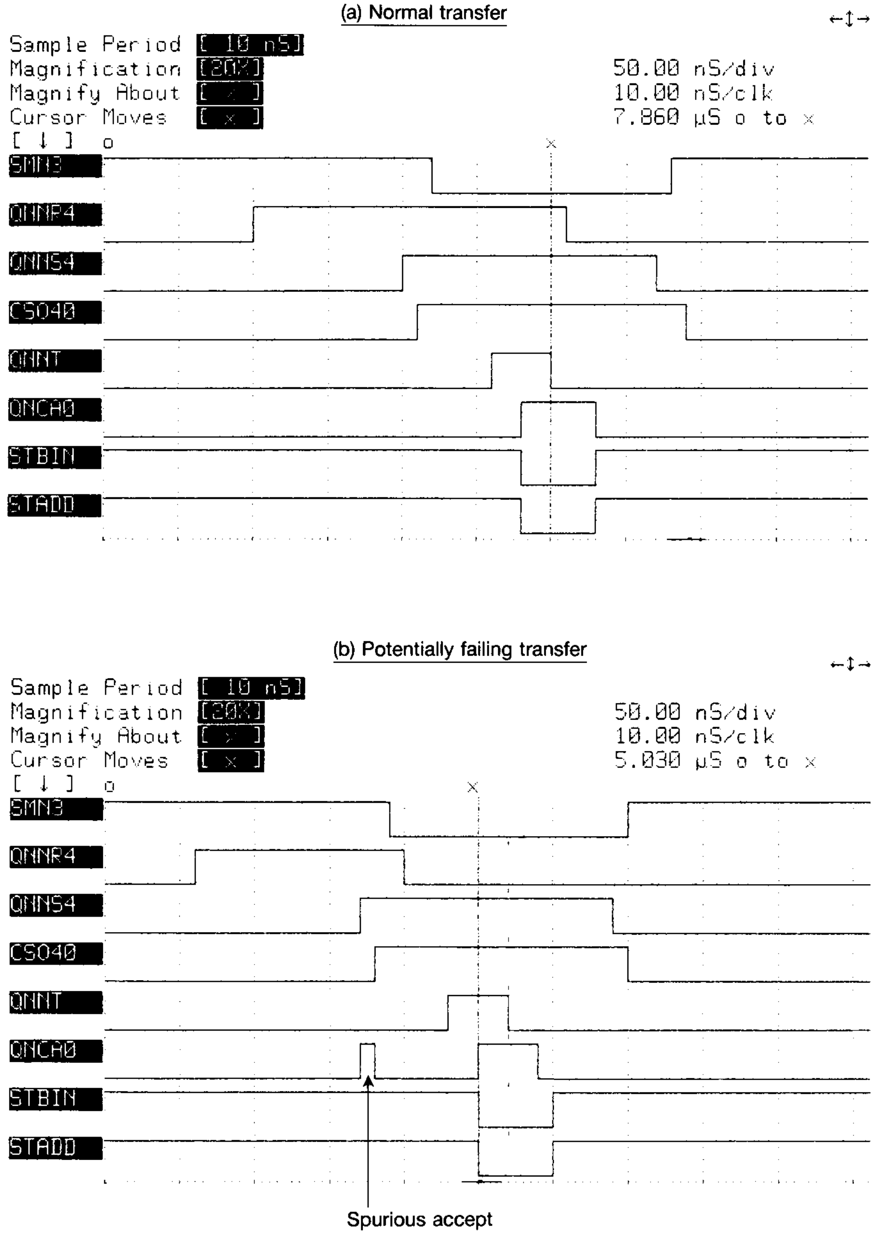


Fig. 5 (a) SCU transfer timing, (b) potentially failing transfer

The normal default condition when switched on was for the A interface to be enabled. However, several different conditions could result in selection of the A interface once powered up and B selected for operation.

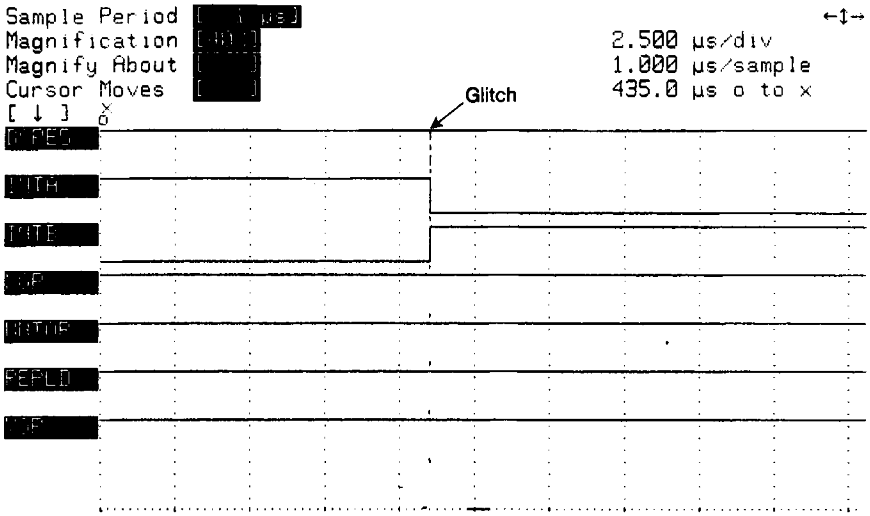


Fig. 6 Glitch detection

In Fig. 6 the trigger point is the switch between B and A interface enables (see INTA and INTB – both signals Low-True). All signals capable of causing the switch from both hardware and software were monitored (see XOP, etc.). A 'glitch' causing the switching was detected on the DCRESET line (see marker on DCRES – Low-True), which was caused by induced noise from a static discharge in one part of the paper movement mechanics.

4.3 Microprocessor code tracing

The majority of microprocessor driven applications have little or no software tracing and debugging facilities once outside the development environment.

Furthermore the more generally used microprocessors have virtually no architectural protection against code corruption, and added to this is the almost total lack of error detection at the hardware level. It is not surprising then to find that many faults with microprocessor systems are observed as inconsistent crashes resulting from some form of failure or corruption that occurred a long time before.

In many cases the logic analyser has been shown to be the only practical tool for resolving the more complex faults. One major problem resolved in this category involved the identification of a code error in a much used Cobol Compiler, which became evident only in multi-user environments. A data segment base register was not being restored correctly after use in a sub-routine, resulting in one user activity corrupting the code in another user area. Unfortunately the code path lengths involved make this type of problem unsuitable for illustration.

```
                                                                     <!>
Label>  ADDR     8088 Mnemonic                                       Time
Base >  [HEX]    [                 ASM                     ]    [Relative

[Mark]  X0085    [             Instructions                ]

-0033   1773D     MOV        CX,ES:[BX+05]                       760.0 ns
-0032   1773E       8B code fetch                                720.0 ns
-0031   1773F       4F code fetch                                760.0 ns
-0030   17740       05 code fetch                                760.0 ns
-0029   17741     SUB        CX,#03                              760.0 ns
-0028   01DB8       02    memory read                            1.000 us
-0027   01DB9       00    memory read                            720.0 ns
-0026   17742       E9 code fetch                                760.0 ns
-0025   17743       03 code fetch                                760.0 ns
-0024   17744     ADD        AX,#0009                            760.0 ns
-0023   17745       09 code fetch                                720.0 ns
-0022   17746       00 code fetch                                760.0 ns
-0021   17747     OUT        #8C,AL                              760.0 ns
-0020   17748       8C code fetch                                760.0 ns
-0019   17749     OUT        #84,AL                              1.120 us
-0018   0008C       BC    i/o write                              1.000 us
-0017   1774A       84 code fetch                                760.0 ns
-0016   1774B     MOV        AL,AH                               720.0 ns
-0015   00084       BC    i/o write                              1.000 us
-0014   1774C       E0 code fetch                                760.0 ns
-0013   1774D     OUT        #84,AL                              760.0 ns
-0012   1774E       84 code fetch                                760.0 ns
-0011   1774F     MOV        AX,CX                               720.0 ns
-0010   00084       1D    i/o write                              1.000 us
-0009   17750       C8 code fetch                                760.0 ns
-0008   17751     OUT        #85,AL                              760.0 ns
-0007   17752       85 code fetch                                760.0 ns
-0006   17753     MOV        AL,AH                               720.0 ns
-0005   00085       FF    i/o write                              1.000 us
-0004   17754       E0 code fetch                                760.0 ns
-0003   17755     OUT        #85,AL                              760.0 ns
-0002   17756       85 code fetch                                760.0 ns
-0001   17757     MOV        AL,#**                              720.0 ns
+0000   00085       FF    i/o write                              1.000 us
```

Fig. 7 Microprocessor code trace

The much simpler problem illustrated by the code trace (Fig. 7) is from a terminal controller handling transfers to a communications network. A serial I/O controller chip together with a DMA (Direct Memory Access) controller chip are used to provide automonous data transfer to the communications

lines. The symptoms of the problem were an apparent controller 'hang-up', with continuous activity on the comms line. The code trace shows the character count being written to the DMA controller (see lines +0000 and −0005) as hexadecimal FFFF – a somewhat large count (65535) for a 300 character per sec. comms line transfer!
Further tracing back from this point identified a code fault associated with polls and data buffer handling, the count of 2 (see lines −0027/28) being that normally used for a poll.

The trace also illustrates the value of the pre-processor and inverse assembler software, now available for all the commonly used microprocessors.

4.4 Analogue inputs

There are cases where it is required to see the precise nature of a particular signal in analogue form. If this signal subsequently results in some form of failure and will occur once only, then normal oscilloscope techniques are inadequate, since before the trigger, single shot data is required.

Typical cases are analogue servo signals, amplifier outputs from magnetic media and various types of noise problems. A number of instruments now include digital storage for analogue events linked to the normal analyser facilities.

The first example (see Fig. 8) shows a 'glitch' on a chip select signal (see CS) within a microprocessor controller, just at the time when the microprocessor was initiating an interrupt handling sequence (see IACK). The interrupt

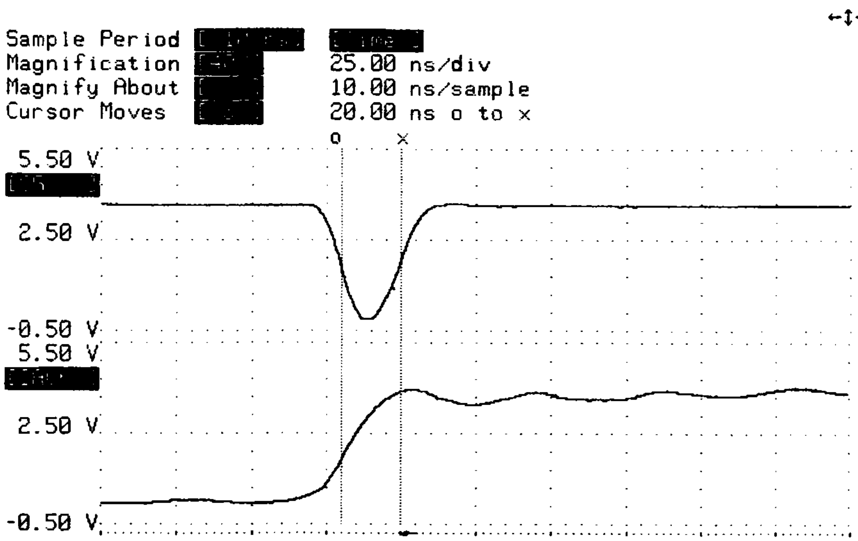


Fig. 8 Analogue glitch display

sequence subsequently failed, providing the trigger source, and the analogue channels provided the 'before trigger' details showing the nature of the fault. The second example (see Fig. 9) is an analogue picture of a static discharge of the type causing the problem illustrated earlier in Fig. 6. With some experience, the analogue presentation provides a way of identifying the source of certain types of electrical noise.

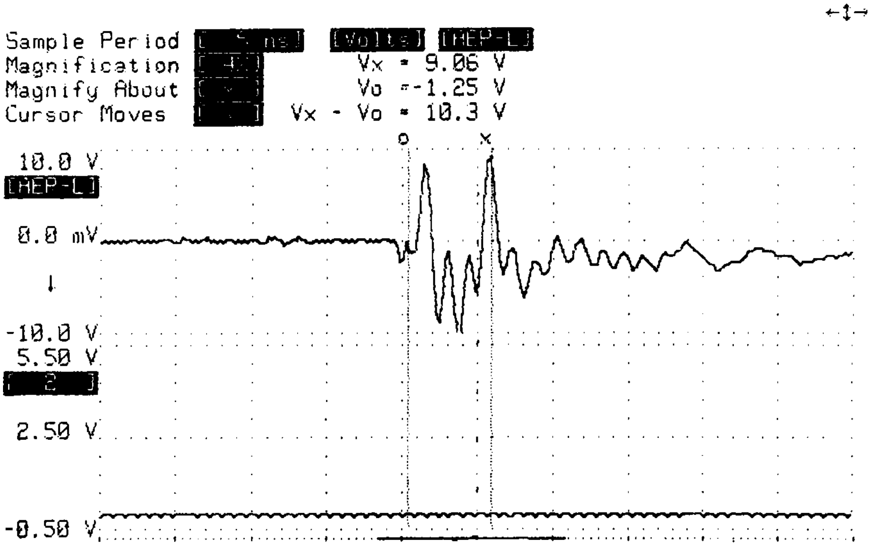


Fig. 9 Analogue-static discharge

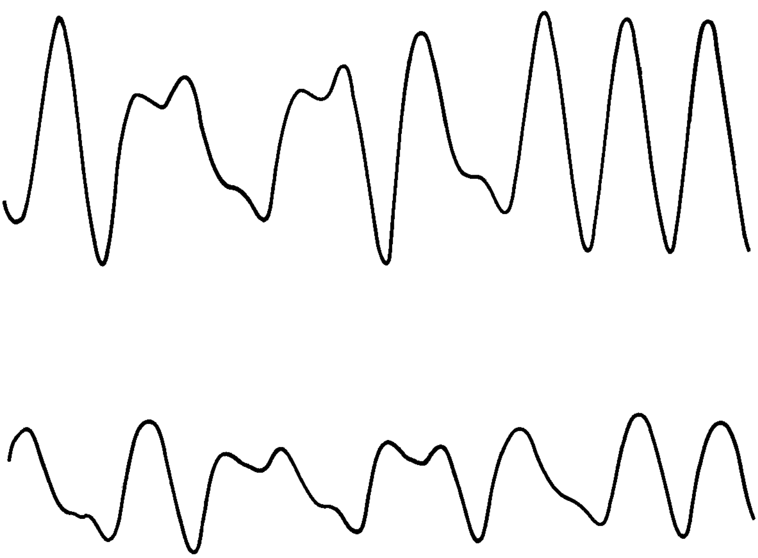


Fig. 10 Analogue-magnetic media output

The third example (see Fig. 10) shows analogue data from two tracks of magnetic tape suffering from temporary amplitude/band width problems.

4.5 Combining state and timing

When investigating a problem it is not unusual to have identified a trigger condition in state analyser terms, but to need to observe the precise timing of the logic in order to determine the cause. The reverse is also true for other problems. Current instruments can be configured as two, three or even four independent analysers linked by their trigger recognition. The example chosen here (see Figs. 11a and 11b) is from an 80186 microprocessor-controlled protocol converter. The symptoms seen from a dump of the software suggest an invalid interrupt. The 'state' part of the analyser was therefore set up to record the microprocessor code sequence, and to trigger from the entry to the invalid interrupt-handling code (see Fig. 11a – #FBC2B

```
                                                                  <!>
Label> ADDR   80186/8087 Mnemonic  (enhanced)                     Time
Base > [HEX]  [                 ASM                      ]   [Relative

[Mark] XXXXX  [               All Cycles                 ]

-0026* 1C1C8   MOV        AL,[SI+2E]                           280.0 ns
-0025* 1C1C9     44xx executed code                            120.0 ns
-0024* 1C1CA     xx2E executed code                            1.160 us
-0023* 34242     xx05 memory read                              520.0 ns
-0022* 1C1CB   AND        AL,#FE                               280.0 ns
-0021* 1C1CC     xxFE executed code                            240.0 ns
-0020* 1C1CD   MOV        DX,[SI+23]                           160.0 ns
-0019* 1C1CE     xx54 executed code                            400.0 ns
-0018* 1C1CF     23xx executed code                            520.0 ns
-0017* 34237     80xx memory read                              1.280 us
-0016* 34238     xx06 memory read                              640.0 ns
-0015* 1C1D0   ADD        DX,#02                               280.0 ns
-0014* 1C1D1     C2xx executed code                            120.0 ns
-0013* 1C1D2     xx02 executed code                            640.0 ns
-0012* 1C1D3   OUT        DX,AL                                280.0 ns
-0011* 00682     xx04 i/o write                                1.040 us
-0010* 1C1D4   JMP        np 1C281                             120.0 ns
-0009* 1C1D5     AAxx executed code                            280.0 ns
-0008* 1C1D6     xx00 executed code                            120.0 ns
-0007* 0C281     C3C1 interrupt ack                            1.960 us
-0006* 0FFFF     FFFF interrupt ack                            1.440 us
-0005* 003FC     04EB memory read                              1.160 us
-0004* 003FE     FB74 memory read                              920.0 ns
-0003* 3A47E     F206 memory write                             880.0 ns
-0002* 3A47C     1C1A memory write                             920.0 ns
-0001* 3A47A     00E1 memory write                             520.0 ns
+0000* FBC2B   PUSH       DX                                   1.320 us
```

Fig. 11a Microprocessor code sequence

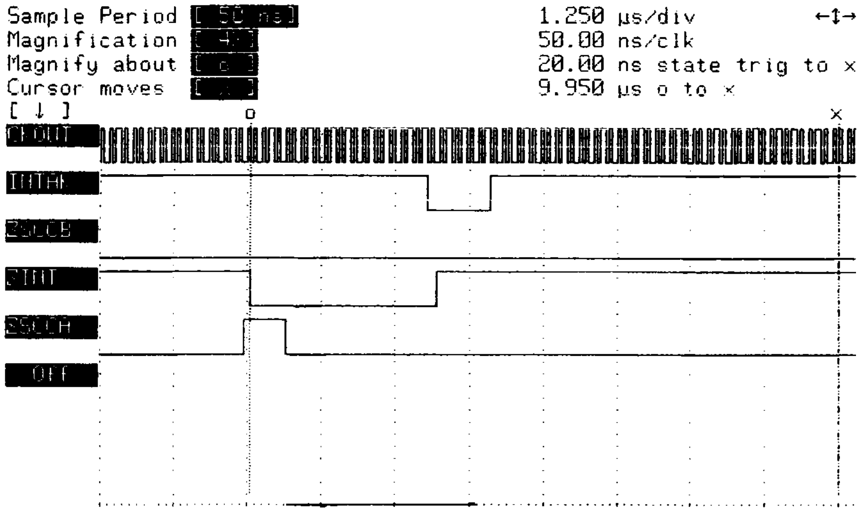


Fig. 11b Timing of invalid interrupt

at line +0000). The 'timing' analyser was set to end recording when the 'state' analyser triggered, so providing waveform information covering the previous interrupt sequence.

Just prior to the interrupt sequence the 80186 is doing an I/O write to a serial I/O controller on port address #00682 (see Fig. 11a, line −0011). The data OUT (#04) is instructing the controller to disable its 'transmit interrupt', and can be seen in Fig. 11b as the chip select 'ZSCCA'. At precisely this time the I/O controller is raising its interrupt line – see ZINT (Low-True). The microprocessor then enters its interrupt sequence (see line −0007), with INTACK going true to call for the interrupt vector. Unfortunately the I/O controller drops its interrupt as soon as INTACK is raised, so that the bus will float high when the 80186 is expecting the vector, resulting in #FF (see line −0006) rather than a true address pointer.

4.7 Performance measurement

The normally accepted method of measuring the performance of specific parts of computer systems is by special software monitoring packages. The software method has two major disadvantages. First the additional code actually changes the execution times of the paths being monitored, and secondly it cannot be used to measure intimate hardware functions. The logic analyser, being essentially a passive device as far as the system is concerned, can provide low level performance monitoring without these disadvantages.

The example chosen here (see Fig. 12) is again from the 2900 series intermodule highway (the SCU) described in sections 4.1 and 4.2. It was required to measure the spread of timings from sending a specific type of command message, to receiving the responding acknowledge message under conditions of very high I/O load on an SD 2988 system.

The logic analyser takes random samples of the specified conditions and presents the results as a histogram as shown in Fig. 12. In this particular case, whilst the majority of commands were acknowledged within 3 uS, there was a peak between 4–10 uS, and a very small number (<1%) took as long as 410 uS. The data provided initiated various performance improvement changes.

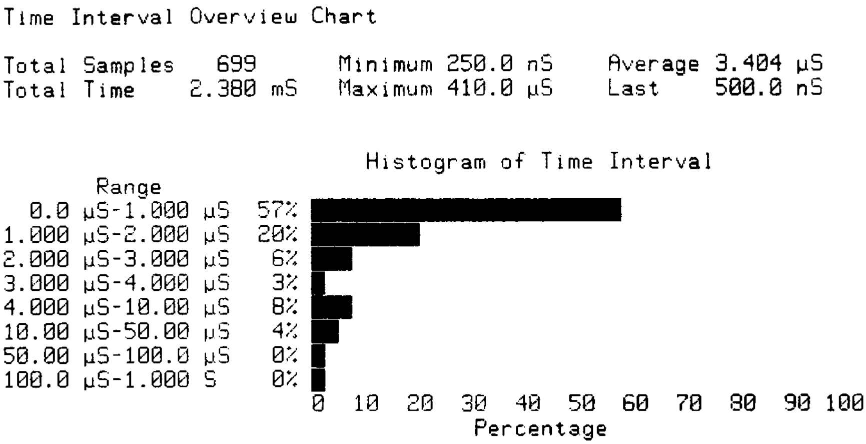


Fig. 12 Performance-acknowledge timing

A similar technique can also be used for low level software, with the histogram providing, for example, the relative time spent in different parts of the code. This technique is used to identify performance bottlenecks, and to highlight the most cost effective areas in which to devote 'tuning' activities.

4.7 Summary

It is hoped that the few simple examples described in this paper will convey an idea of the very wide variety of applications for which logic analysis techniques are a valuable aid to problem solving. New techniques have been developed, and with support from instrument manufacturers many of the requirements derived from our experience have been included in the design of new instruments, so enhancing this capability.

5 The future

By practical application a totally new technique has been added to our spectrum of problem solving methods. The application of the logic analyser has proved to be a powerful tool for resolving some of our most complex problems.

The areas of concern for this level of problem investigation in the future are many and varied. The impact of larger scale integration, new microprocessors with on-chip 'mainframe' architecture and customised gate arrays, together with the changes in manufacturing technology (e.g. surface mounting) and the impact of multivendor networks will all require new facilities and the development of new techniques. Already areas of enhanced performance measurement, and methods of validation using fault injection, are being addressed using logic analysers.

Altogether, it seems safe to say that the logic analyser and its derivatives will continue to play a key role in problem resolution in the foreseeable future.

Repair – past and future

G.M. Coiley
Worldwide Spares Repair, Cavendish Road, Stevenage, Hertfordshire

Abstract

All components of any manufactured product are subject to failure and therefore every manufacturer has to make decisions concerning repair and replacement. When the products are as complex as modern information processing systems and are installed in large numbers all over the world these decisions are not straightforward and are of great economic importance to the manufacturer. The paper discusses the general considerations on which such decisions are based.

1 Background

When one thinks about the 'repair business', be it in cars, buildings or electronic equipment, the amount carried out is perceived to be reducing: customers are no longer accepting breakdowns as a part of technological advance. Innovative ideas when hastily put into practice are seldom trouble free, but the progressive pushing forward of technological horizons should not be expected to affect reliability in anything but a positive way.

To analyse the overall effect on repairers, a number of contributing factors need to be understood.

1.1 Shipped mix

The first of these is the revenue earning despatches and its effect upon failures due to mix.

Upon the introduction of a more reliable product the installation and early life failure rates will improve. The rate of this improvement is very dependent upon how much of the weekly/monthly shipments are made up of the new product. Thus a motor manufacturer who markets only 3 models of vehicle will witness a more profound reduction of failures of new products in the field when exchanging one for a more reliable model than a computer manufacturer with 15 to 20 products who does the same.

A snapshot of shipment data is likely to reveal that only a minor part of this year's shipments contain this year's designs. It will be years perhaps before a

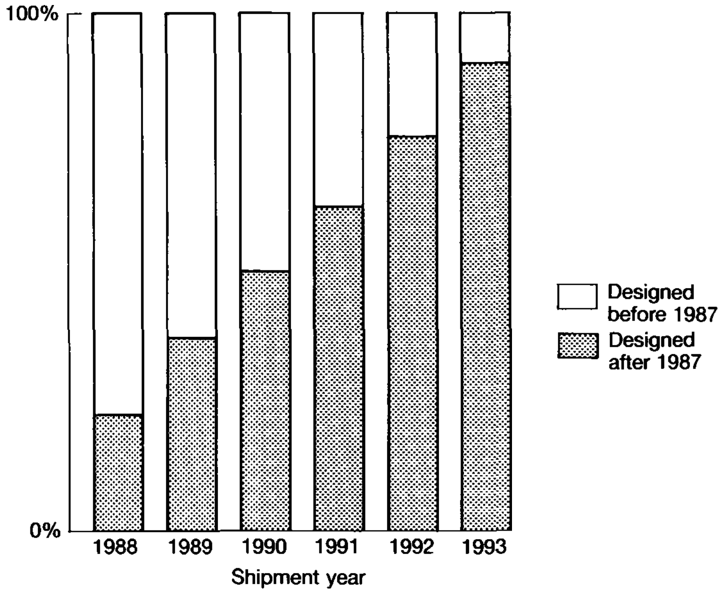


Fig. 1

shipment consists entirely of 'post-1987' designed products. This is illustrated in the 'Conceptual Chart' of Fig. 1. If this is true of shipments it is certainly true for the installed base, and the time taken to turn over the base completely is even longer than for the shipments.

1.2 Replacement

The option to repair can easily be discounted through manufacturing cost improvements. This is to say that when a spare part becomes as cheap to replace as to repair, or cheaper, the faulty part may be discarded. But again this has proved a slower process than at first thought, for three main reasons:

Firstly, technological turnover. Reducing costs in manufacturing tends to take place after a technology has been proved. For example, it has taken several years for monochrome monitors to become so cheap to produce that repair seems uneconomical, but during this period most consumers have upgraded their requirements and prefer colour monitors, for which repair is usually cost effective. The migration from floppy disks to hard disks tells the same story. Throughout history, old products have become throwaway whilst the latest technology becomes repairable.

Secondly, repair itself has become cheaper: the low profile keyboard for under £60.00 was followed by the £10.00 keyboard repair. Part of the reason for this is volume. The same effect which volume has on manufacturing is evident in repair.

Finally, consumables have a tendency to become repairables simply because they cease to be manufactured at all so that repair is the only option open, save for an expensive 'all time buy' with inherent inventory costs.

All of the above point to improved reliability on newer products and cheaper maintenance costs in the long term; but will repair volumes fall?

2 Growth of repair volumes

This is not a paper on market growth potential, but this has an important effect on overall repair volumes. It is reckoned that annual growth in mainframes will continue at 10% worldwide, minis at between 14% and 16% and office products at between 25% and 30%. Growth of installed product at these rates, without improvement in reliability, would lead to a massive increase in the absolute number of repairs required.

To analyse this more closely it is necessary to look at where improved reliability is coming from. An indication is given in Fig. 2, where general product types are grouped together showing their Basic Annual Maintenance Cost (BAMC) as it improves from year to year. This data is of a generalised nature showing trends only but the point is that mainframes yield

PRODUCT CATEGORY	YEAR SHIPPED					
	1986	1987	1988	1989	1990	1991
Mainframe	0.88	0.83	0.78	0.73	0.68	0.63
Minis, Peripherals, PBX, Keysystem, Modem, Mux, Front end, Lans	0.95	0.93	0.91	0.89	0.87	0.85
WP, PC, terminals	1.00	1.00	1.00	1.00	1.00	1.00

Fig. 2

	Growth	Reliability improvements	Repair volume
Mainframes	Least	Greatest	Reduces
Minis	Ave	Ave	Unchanged
PC/Office	Highest	Least	Growing

the biggest improvement, followed by minis, with office products showing no improvement at all.
Of course without actual studies in great depth this is hard to prove, but 3rd party maintainers and now even 4th party maintainers seem to be clearly focused upon a growing office systems opportunity. For my money repair volume is set to rise for at least the next 5 years.

3 Where to repair?

However much repair work must be carried out, if all other things were to remain equal the best place to effect a repair would be at the point of failure. Naturally all other things are not equal and some of the factors to take into account are as follows:

3.1 Capital

It may cost a great deal of money to buy the necessary test equipment to support a repair type. It may not therefore be economical to put these machines in many places, in fact it is often true that adequate loading may be realised only by collecting all such repairs together centrally. The key to dispersal of capital is therefore volume.

3.2 Transport costs

As well as capital, transportation costs have an impact. If a spare part is cheap to replace but even cheaper to repair, then if it exists in sufficient volume, on-site repair or local repair will be viable, but the cost to be borne by sending the part to a central repair depot may render repair uneconomical.

3.3 Spare parts prices

On the other hand, a part which cost a great deal of money but is found only in low volumes will be economical only if brought to a central depot where economies of scale can be brought to bear. If the volume is very low, even an expensive part such as this may be uneconomic to repair.

It will be seen that a number of factors can affect whether a spare part should be repairable or consumable, and if repairable whether this should be done locally or centrally: this is expressed in Patton's book, *Spares Management*[1]. The general principle is illustrated in Fig. 3.

3.4 Spares life cycle

Most engineers are familiar with the 'bathtub' curve, which shows how all things have a tendency to fail in early life, how a stable (usually, low) failure rate establishes itself and how unreliability through 'old age' eventually sets in.

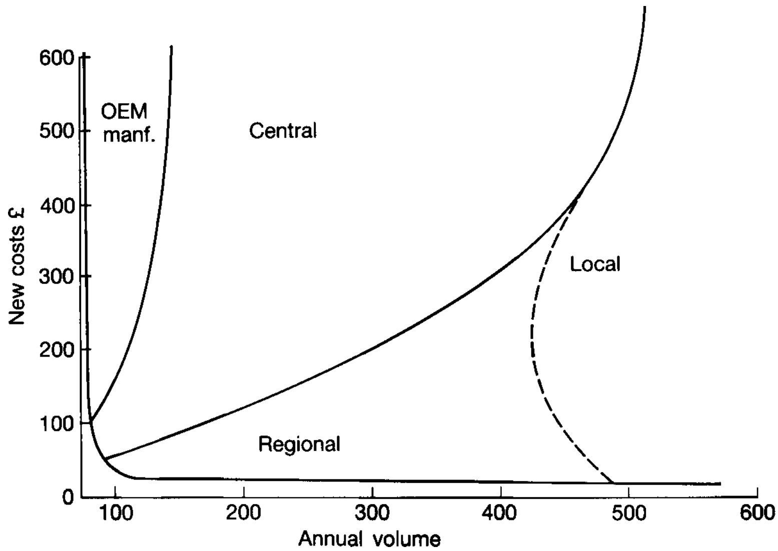


Fig. 3

The amount of machinery in the installed base also goes through a cycle of growth when products are being shipped, a peak when shipments stop and finally a decline. How these two effects interplay one with the other varies from product to product, but it is possible to predict how many failures will occur over the entire product life and when, at least on a coarse scale.

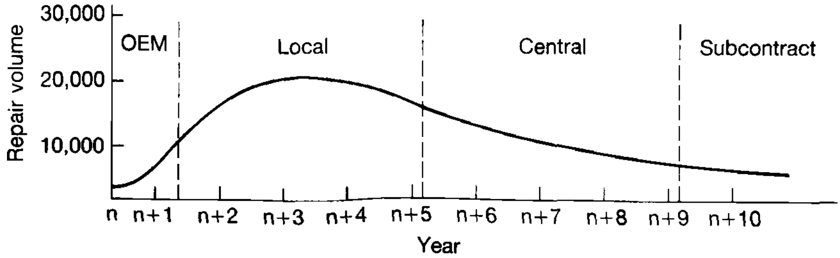
It is quite clear that the most economical place to carry out repair is likely to change at least once in the product's life. The two curves below show how different product life cycles can be addressed with different repair strategies over different parts of their lives.

In the first example it would be wise to take into account the equipment available locally and centrally to see if some common equipment could be used to base the repair solution on. In the second a single repair solution would be appropriate, and since portability is not required ATE (Automatic Test Equipment) would be ideal in this case.

4 Repair methods

So far I have tried to cover, in brief, why repair and where to repair. How to repair is of great significance.

At one time repair mirrored manufacturing to some extent, with similar people using similar techniques for machine test and repair; this is less and less the case today. The methods employed to manufacture have taken enormous strides forward. Flexible production lines can turn the same



Life cycle for product A

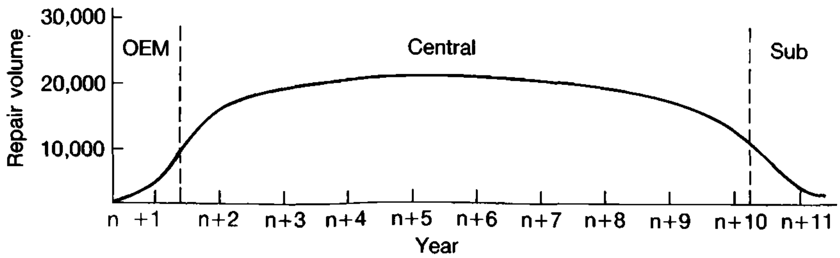


Fig. 4

production machines to new products, so repairers don't inherit the repair equipment so much any more: even if they did it would be too large and volume-dependent. Bear in mind that repair volume is likely to come from a larger installed base at greater reliability. Volume may not be a feature so much as variety; even the same product may come in several build levels.

Many repair solutions have to be portable (i.e. local and central), cost effective, and flexible in the variety of parts they can repair. Attention is now being focused on a new generation of test equipment which fulfils all these criteria.

4.1 Generic repair

Repair solutions which may be applied to more than one spare part have been developing for some time; power supplies and monitors, etc. are often tackled using such equipment.

This principle is now being used to repair more complex spares, including microprocessor based parts, where boards and assemblies which use similar microprocessors and design architecture can be repaired by the same equipment, usually a microprocessor emulator. The control of this equipment is written in software and interacts with the software in the product under test. This development requires an understanding of product design, by the repair development engineer, approaching that of the designer himself.

Perhaps as a final comment I should say that the skills of the manufacturing engineer and those of the field service engineer, and to a great extent of the software design engineer also, are met in the repair engineer of the future.

The process by which the field service engineer is reverting to whole-unit swop-out, and the manufacturing engineer is replaced with more and more mass testing and diagnostic machinery, is causing the component-level repair environment to become one of the most highly skilled areas of today's electronics industry.

Reference

1 PATTON, J.D. Jnr. Service Parts Management Instrument Society of America NC USA.

OSI migration

J. Houldsworth
ICL Network Systems, Stevenage, Hertfordshire

Abstract

The first issue of the ICL Technical Journal in 1978 contained a paper by the same author on the importance of an architectural approach to network systems design (Ref. 2).

This work was adopted by the standards organisations as the basis for the work on Open Systems Interconnection and is the strategic direction which is now being followed by the entire IT industry. The universal adoption of OSI standards is vital to the future of networking.

OSI has now matured into a practical procurement tool through the process of selecting defined combinations of standards for key functions and specifying conformance tests which can be used for independent certification. Users can demand Open Systems as the only way of escaping from the locked in world of proprietary protocols and architectures to the freedom of multi-vendor procurement.

This paper explains why OSI is so important and how to get started along the OSI migration route. It describes how to classify systems in terms of both geographic characteristics and system configurations and discuss the related migration targets. It explains how to avoid the 'big bang' approach by planning the migration in logical steps and shows how to relate these steps to the natural forces which drive system evolution.

It stresses the importance of installing an OSI 'bearer network' as the foundation for the OSI interworking and application functions.

1 Introduction

We are through the academic phase of Open System Standardisation. Enough standards are now in place for governments and major users to specify combinations of the base standards, known as profiles, against which they will procure. Failure to be able to supply to these profiles will exclude vendors from even bidding for the business. Many smaller users have recognised this turn of events and they too are actively starting to demand Open networks which the Information Technology suppliers are in a strong position to supply.

Those who do not adopt OSI will become locked into proprietary architectures which are controlled by individual manufacturers and they will become isolated from the benefits of Open Systems.

OSI Standards are in the public domain.

'OSI Migration' describes the movement away from proprietary architectures and protocols to OSI networking strategies and it is essential that the process should be accelerated.

2 OSI Migration – the driving forces

OSI Migration is not a 'big bang' process. Very few users will wish to make the transition to OSI in one stage; the disruption and cost would be too great. Progress will only be made when there are perceived benefits.

The migration process will follow a number of stages, which will depend on the initial configuration and the driving forces. It is essential to establish clearly the basic starting scenario, the target scenario after Migration and the valid intermediate steps which can be used to achieve the target.

The driving forces which motivate individual users will change from time to time as the system evolves to meet the needs of the organisation and these driving forces will always condition the choice of the next migration step.

Typical driving forces which influence the user in his choice of a specific Migration route are:

Corporate Processing Drive – The need for a more powerful or a higher facility mainframe.

Personal Processing Drive – The need to expand the terminal population or to introduce faster/higher facility/intelligent terminals.

Distributed Processing Drive – The desire for the flexible distribution of intelligence with local processing facilities.

Networking Drive – The desire for an integrated bearer network and a reduction in networking costs.

OSI Drive – The recognition that OSI represents the future and the need to be a part of the multi-vendor Open System environment.

It is important that these driving forces should be recognised and linked to the Migration strategy so that each stage of the evolution brings the user one step nearer to the OSI goal.

3 Planning for migration

Migration will require changes not only in the network itself, but also in the systems connected to it. Some systems will not support OSI and will need to be upgraded. Any new end system which is introduced must either be native OSI or be capable of being upgraded to an OSI end system (SWING systems).

The Migration process needs to be carefully planned if it is to be accomplished successfully. In addition to the evolutionary steps in upgrading the system, it must also address the use of migration aids and pilot systems and the contingency for regression.

The most important element of the plan is the end goal. Start by looking at the current situation and then clearly identify what the system will look like after migration. Ignore the solutions for today's problems during the process of charting the start point and the end goal, they will only cloud the issue and may lead into a blind alley in migration terms. They will fall into place when a proper and logical migration path is charted during which it will be easy to check that the solution to these individual problems is a valid step along the migration route. If you take your eye off the ball and proceed by always solving the next visible problem you will, almost certainly, fail!

4 OSI – The architecture

OSI standards make it possible to break down systems into a manageable number of simple alternatives, which can be fitted together like building blocks to construct any networked solution, no matter how large or dispersed. Familiarity with these building blocks helps in the evaluation of any environment and the assessment of the migration opportunities.

The classical Open System Interconnection architecture (Fig. 1) is a seven 'layer' model in which each layer represents a key function in the system operation. Each layer is labelled with its basic function which is reasonably self explanatory. Seven is not a magic number: it was chosen because the organisations involved in its creation agreed that seven is appropriate for achieving a manageable analysis of the functions involved in data communication.

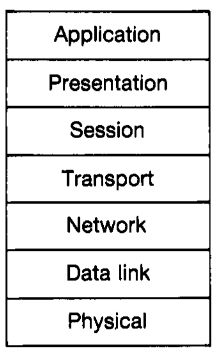


Fig. 1 The seven layer model

Alone, this simple model has done more for Open Systems than anyone could have imagined. Having created the template for the generation of all other standards for interconnection and interworking, the prospect of Open Systems is now a reality. Indeed, most users now insist that any system that they buy is designed according to the principles of the OSI Model.

Layers 1 through 4 (commonly referred to as the 'Transport Function') contain the interconnection or 'bearer' elements and layers 5 through 7 (the users of the Transport Function) contain the interworking elements.

The lower 4 layers (the Transport Function) aim to create a transparent interconnection environment over which the interworking functions in the upper 3 layers can run independently of the transport media (Fig. 2).

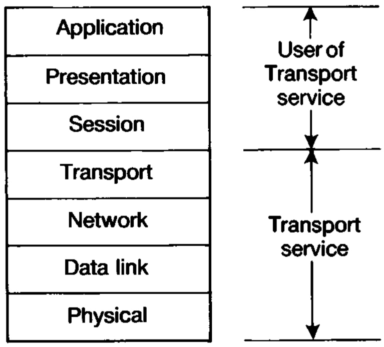


Fig. 2 The transport service

The Transport Layer is the key layer in the model because it forms a neat dividing line between interconnection and interworking. It quarantines the upper 3 layers from the network, allowing standard applications to run over the various types of Local and Wide Area Networks which are needed to suit the specific practical interconnection requirements. We will see more of the 'mix and match' concept when discussing the selection of the elements of the bearer network to suit particular geographic requirements.

The division at Transport Layer allows a simple concept of the model to be evolved (Fig. 3) around the two basic layers, 'interconnect' (which joins things together) and 'interworking' (which makes sure that they all understand each other).

Figure 3 indicates the need for gateways in the interworking area. They are required to communicate with systems which use proprietary interworking protocols but the need will diminish as more manufacturers introduce standard OSI interworking protocols.

Figure 3 is completed by a third grouping which represents the overlay

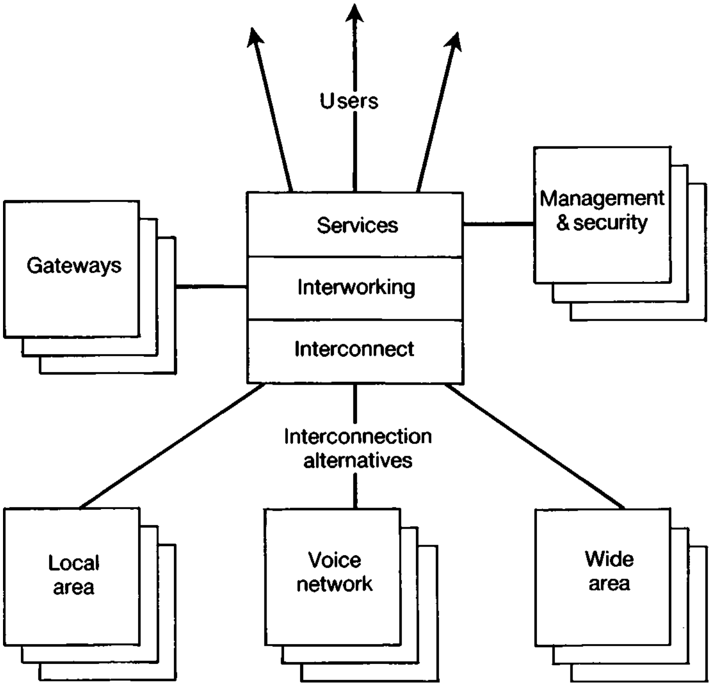


Fig. 3 Simplified model

services which span several layers of the model such as: system management; security and directory control.

5 Functional standards – the Migration catalyst

The logical grouping of the interconnection and interworking functions is being exploited by several user groups, including government procurement agencies, and the Standards Organizations. They are creating 'Functional Standards' for procurement by selecting preferred combinations of the base standards in the interconnect layers 1 through 4 and the interworking layers 5 through 7. These are sometimes referred to as 'Functional Profiles'. Functional Standards are option free and rigidly defined and they can be used by independent test houses for formal conformance testing and certification.

The Functional Standards activity began in Europe in the SPAG Manufacturers' Group, of which ICL is a founder member. The European work has now been adopted by the CEC standardisation committee consortium, CEN/CENELEC/CEPT, who are preparing European Standard Profiles. General Motors and Boeing have spearheaded an activity to select Manufacturing Automation Protocols and Technical Office Protocols (MAP/TOP).

The British and USA governments have set up the UK GOSIP and US GOSIP committees to establish preferred Government OSI Procurement profiles. The USA manufacturers have set up the organisation known as COS to establish the preferred US profiles. ICL is the first European company to be allowed to join COS and has been given a seat on the COS board. ICL is using its joint membership of SPAG and COS to pull the European and USA initiatives together. Japan has also set up a similar organisation, called POSI, which has links to COS and SPAG.

The ISO has recently established an activity to put the International seal on the profile work. They are using the other profiling groups as feeder organisations and will produce International Standard Profiles (ISPs).

The Functional Standards activity has moved OSI from being a collection of standards to a new role as a procurement tool.

6 Transport layer – the key to Migration

The quarantining effect of the Transport Layer can be used to simplify OSI Migration.

A good example is the strategy which ICL has adopted in its Information Processing Architecture (IPA). IPA (ref. 3) covers the entire ICL Networked Product Line (NPL) and includes both existing proprietary standards and OSI standards with tools for the graceful migration to full OSI. A policy decision was made several years ago that all IPA systems must include the Transport Layer to link the interworking and interconnection functions, regardless of whether proprietary or OSI protocols are being used.

Transport is now implemented across the ICL Networked Product Line and allows the well established IPA applications and services to run over either proprietary (CO3) interconnect or OSI bearer services (Fig. 4). In Migration terms this means that any user can instal an OSI bearer service and continue to run existing IPA applications and services until it is convenient to introduce the equivalent OSI applications. During the Migration phase, the OSI applications can run alongside the traditional applications to avoid any disruption of service.

This leads to a simple phased OSI Migration Strategy:

Get the Transport Layer in place.
Introduce the OSI bearer networks.
Continue to run existing applications and services to avoid disruption.
Introduce OSI applications and services as pilots alongside the existing applications.
Switch over to full OSI applications.

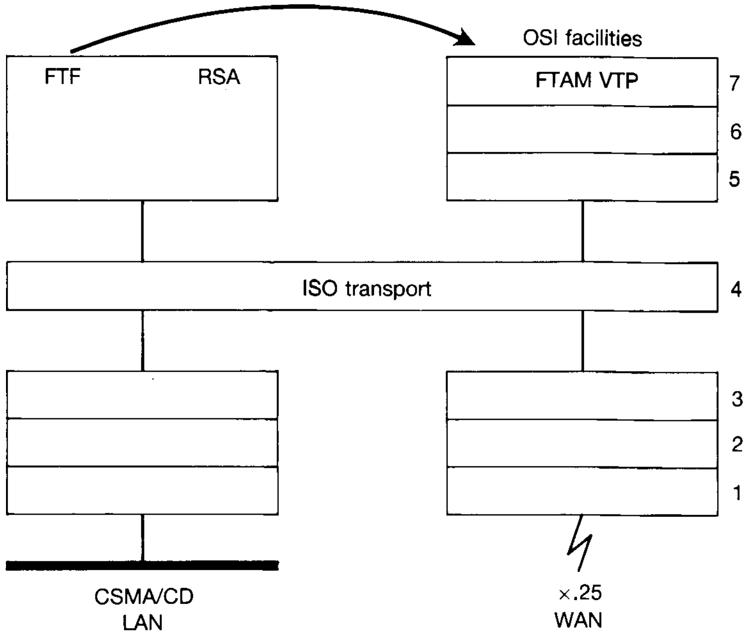


Fig. 4 Transport – the key to migration

7 OSI bearer networks – the sound foundation

Figure 3 shows that the key parts of the OSI bearer network can be created by mixing and matching selections from the LAN and WAN and PABX interconnection alternatives. These alternatives are discussed below.

7.1 Local Area Networks

The ISO has developed four LAN standards but only three of these are serious contenders for local area network implementations. The four standards are: CSMA/CD baseband bus (often referred to as ETHERNET), which is the most popular system for multi-vendor environments; Token-bus, which was developed to run over broadband systems as a part of MAP; Token-ring, which is a competitor to CSMA/CD but is not often applied in multi-vendor environments and Slotted-ring, which has found almost no application and can be discounted.

The first OSI LAN functional profiles to emerge from Europe were based on the CSMA/CD standard. TOP has also adopted CSMA/CD and MAP has adopted Token-bus as mentioned above. Hence, the choices of LAN have been narrowed down by the profiling exercise and this simplifies the choices and the design of the local area interconnections. All the current LAN profiles terminate in the richest class of the Transport standard (class 4), which includes full recovery from errors.

7.2 Wide Area Networks

The OSI choices in the WAN area have been narrowed down even more by the profiling exercise. Discounting Scandinavia, which favours switched digital services, the only serious contender is the X.25 packet switching standard which was profiled by the CCITT and has now been adopted by CEN/CENELEC/CEPT.

The current standard profile uses the 1980 version of the X.25 standard, terminating in either class 0 or 2 Transport, but this profile is being upgraded to embrace the 1984 version of X.25 which has some additional facilities.

7.3 The Private Automatic Branch Exchange

The PABX is included as an interconnection element because it can be used to provide switched access from enquiry terminals to the OSI LAN through a gateway. The relationship between the LAN and the PABX is discussed later. The PABX will assume an increasingly important role as a true part of the OSI bearer network when 64 Kbits/sec ISDN extension standards are implemented. Some PABX suppliers are providing high speed digital connections by using special data over voice adapters but these are proprietary and not a part of the OSI standards set. If such connections are used they should be carefully planned into the Migration strategy as a transition route to ISDN later.

7.4 Integrated Services Digital Networks

ISDN standards are being implemented to provide 30 PCM digital voice channels for inter-PABX connection. Each of the channels runs at 64 Kbits/sec and the channels on the inter-site ISDN trunks can be shared between voice connections, high speed interconnections between LANs on different sites, access to X.25 Wide Area Networks and the interconnection of ‘nodes’ in private packet networks.

8 OSI bearers – making the choice

8.1 OSI bearers – geographic considerations

Organisations fit broadly into two geographical classes: those whose operation is on one or more closely grouped sites (a local area), and those who have several such sites which are widely dispersed (requiring a wide area network component in the solution).

Figure 5 shows a typical span of environments that will be encountered in a major corporate organisation. These are: a corporate office; a factory; a

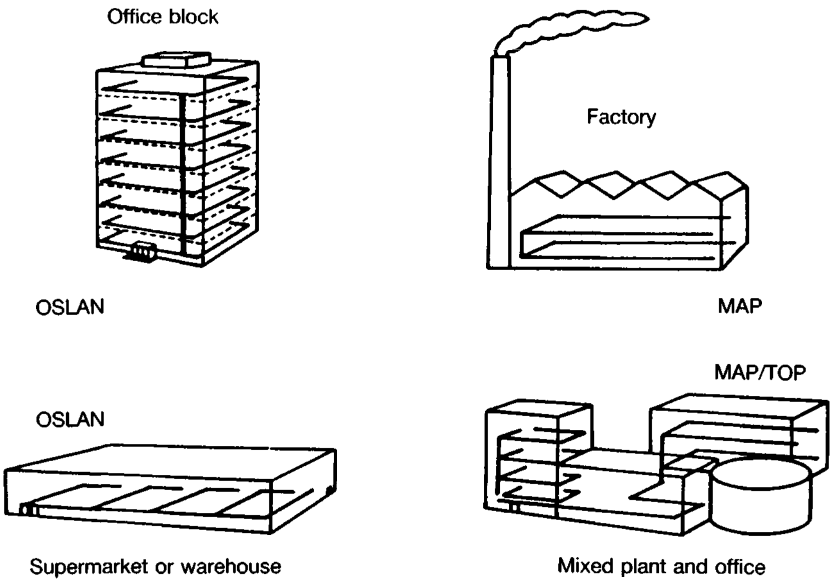


Fig. 5 Typical span of environments

mixed office and processing plant and a warehouse. The individual implementation choices will be mapped on to this diagram as the overall network picture is constructed.

The design can start from either direction.

If the network is widely distributed, with only a few densely populated sites, OSPAC or a public X.25 network (or a mixture of both) will dominate. In such cases, it is best to start by considering the wide area network needs for the distributed terminals, then group the terminals and other resources in the more densely populated sites around LANs and link the LANs to the X.25 network.

If the network is dominated by a few densely populated sites with a modest population of off-site terminals, it is better to consider the local environments first and then the inter-site and off-site links. This approach is taken first.

8.2 OSI bearers – the local area network

Most local area environments have grown haphazardly. Systems have been purchased to serve individual needs, often without reference to other departments. The PABX will have been purchased by a separate department. There will be a wide variety of single purpose terminals, lots of leased lines to other sites, and a complex arrangement of building wiring with 'star' connections between each of the mainframes and its own terminal population.

There are immense benefits to be gained by integrating these systems into a multi-vendor, resource sharing network with simplified cabling.

Figure 6 highlights such a situation where groups of data processing resources have been traditionally sited in a headquarters building and the corporation now plans to expand into an adjoining building which is isolated by a public highway.

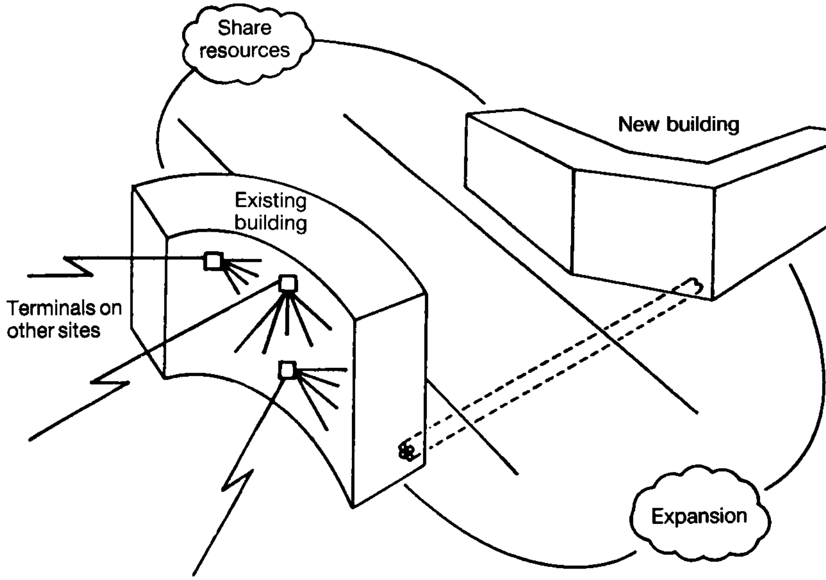


Fig. 6 Local area – haphazard growth

Clearly, it would be inappropriate to extend the current haphazard cabling through the restricted linking ducts and the single integrated local area network shown in Fig. 7 is the ideal solution which allows resources and users to be moved freely between either of the two locations.

Simplified cabling and multi-vendor operation are major benefits to be gained from migration to OSI.

8.3 OSI bearers – LAN and PABX relationships

There is a traditional role for the PABX as a data switch but the relationship between the PABX and the LAN needs to be established. The PABX is not a direct alternative to the LAN which is far superior for high-speed, high volume data, such as file transfer and professional data terminal traffic. As a simple rule, the PABX should only be used for connecting low utilisation enquiry terminals through the extension wiring which reaches most staff

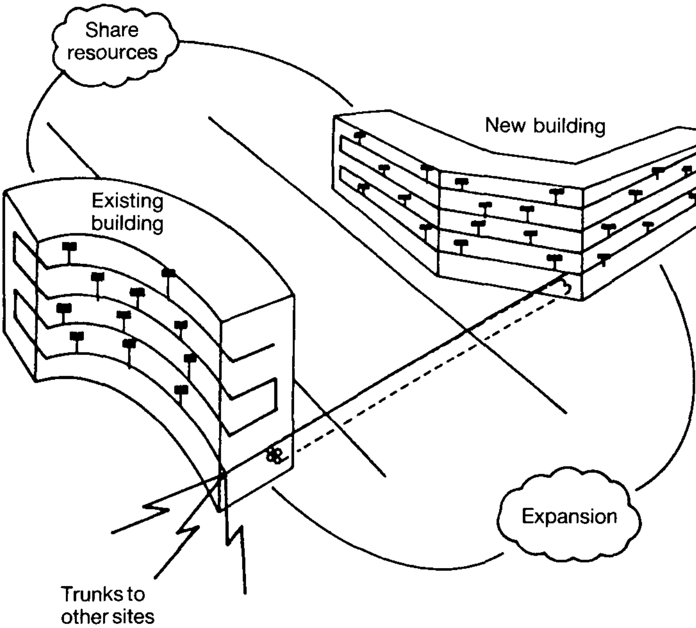


Fig. 7 Local area – simplified network

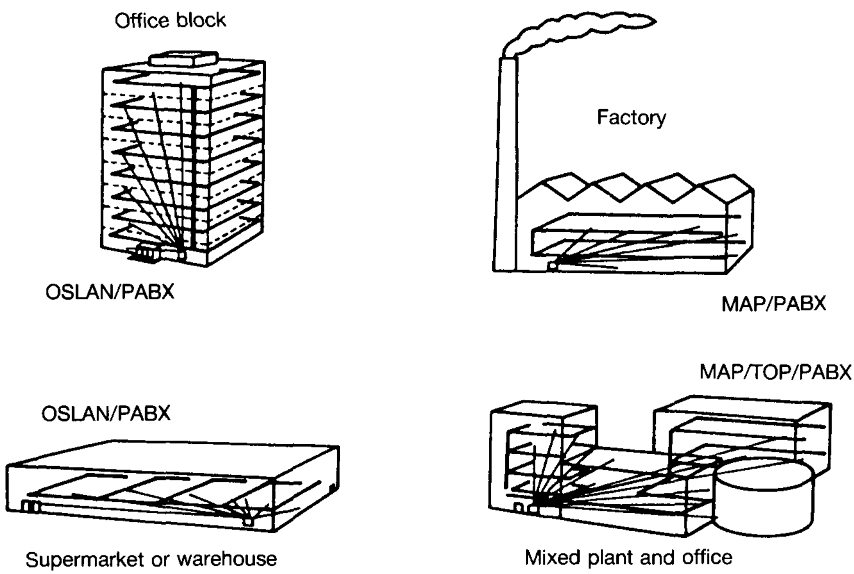


Fig. 8 LAN and PABX mappings

positions. The mapping of the PABX wiring on to the schematic environmental diagram is shown in Fig. 8. An integrating bridge should be provided between the LAN and the PABX environments to ensure that the terminals which are connected to the PABX can access the services on the LAN.

The relationship between the LAN and the PABX will change as more sophisticated PABXs are installed with high speed ISDN digital extensions but the above principles will continue to apply. As the evolution takes place, it will be possible to attach higher speed professional work stations through the PABX but the backbone LAN will still be needed to interconnect Office, Departmental and Corporate servers.

The need for an OSI compatible LAN will not disappear.

8.4 OSI bearers – the Wide Area Network

There are two elements to consider; linking together the LANs on the various local sites and connecting remote terminals to the appropriate resources.

The easiest way to link LANs is through 'Bridges' which interconnect at layer 2 of the OSI model and join several LANs into a single virtual LAN community. Local Bridges (LOBs) are used if the individual LANs are in the same building or on the same campus and Remote Bridges (ROBs) are used if they are on different sites. ROBs communicate via one or more 64 Kbits/sec channels.

An X.25 network can be installed specifically to link the LANs if the network is complex. There will always be arguments about whether bridging or X.25 is the most appropriate for linking LANs but it isn't too difficult to decide. As a general rule, small numbers of LANs should be linked via Remote Bridges. As the number of buildings increases to around 5 or 6, the cross linking becomes harder to manage and an X.25 network may be better because of the enhanced features for managing the flow of information around the linking network.

An X.25 network is an obvious choice for flexible routing between a high population of dispersed terminals and the Corporate, Departmental and Office resources and also for providing common links to external public services.

Even if an X.25 network is installed to handle a dispersed population, some of the larger sites will still justify the installation of LANs and it makes sense to link these LAN communities via the X.25 network.

Even if an X.25 network is installed, the LAN communities can be linked together via LOBs and ROBs if there is heavy traffic on specific routes. However, the X.25 network permits line sharing and could reduce leased line costs.

8.5 OSI bearers – the inter-site trunks

ISDN trunks can be used to link sites together and these can be multiplexed to share the channels between voice connections from the PABX and the data connections from the building LAN (via ROBs and X.25 links). They can also be used to interconnect the nodes in private X.25 networks to reduce leased line costs.

The inter-site links are mapped on to the typical organisation diagram in Fig. 9, which summarises the overall bearer network choices.

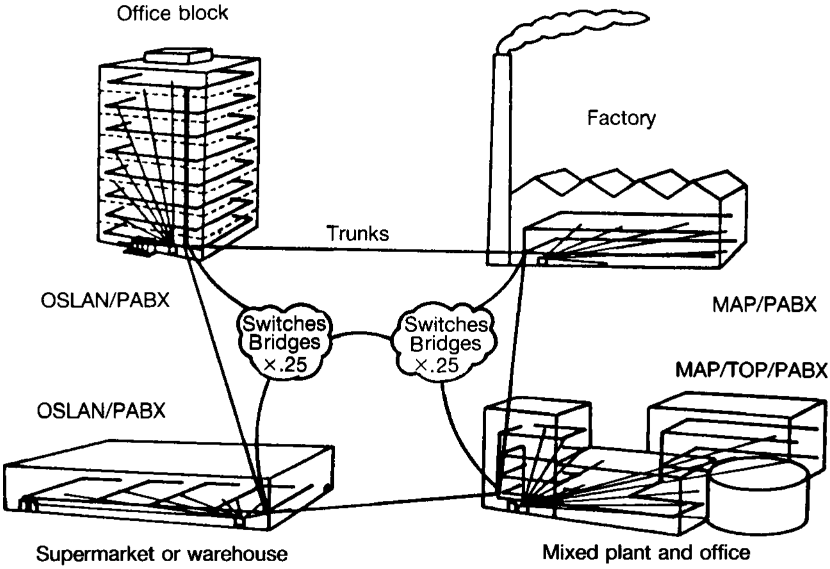


Fig. 9 Network choices

8.6 OSI bearers – summarising the choices

An OSI bearer network contains these elements:

Linked LANS. Local Area Networks within buildings serving local Corporate, Departmental and Office resources and interconnected by remote bridges. This provides a network for an organisation with several sites and many terminals and services on each site.

PABX network. Connection between PABXs on several sites to provide the traditional telephone service but with a gateway for routing enquiry terminals to the LAN network.

Packet network. An X.25 packet network providing the access between remote terminals and services and the main sites.

ISDN trunks. Integrated transmission equipment for multiplexing inter-site voice and data traffic.

9 OSI applications

9.1 OSI applications – the basic elements

It has already been explained that the Transport Layer quarantines the upper layers from the different kinds of OSI Bearer network. This means that it separates the application services from the actual interconnection media.

The three Session, Presentation and Application layers complete the model and they provide the user with a collection of services which can be combined as needed to allow the user to carry out any kind of distributed processing function.

The Session Layer supports the dialogue between Application processes. Each time a processing task needs to communicate, a 'session' is established and it remains active until the communication need is over. Different kinds of session are needed for specific functions and these can be negotiated at session establishment.

The Presentation Layer applies a common shape, structure and format to the data, having regard to the kind of interchange which is taking place, such as file transfer. A 'Transfer Syntax' has been created to define the common rules for interchanging data.

The Application Layer is used to construct the user applications by calling in standard 'Application Service Elements' as they are needed for the job in hand. The elements are FTAM (File Transfer Access and Management), JTM (Job Transfer and Manipulation) and VT (Virtual Terminal). These are 'bound' together into the required 'suite' for the job by a set of Common Application Service Elements (CASE).

9.2 OSI applications – applying the elements

The Application services are bound together to match the required processing service, just like assembling proprietary application elements, but an Application which is constructed from standard OSI Application services will run in a multi-vendor environment.

The first migration steps towards this kind of multi-vendor operation will be the application of individual application services, such as FTAM, to interchange files between co-operating systems. This piloting exercise will lead to the combination of several Application services into more sophisticated suites. The end goal is the flexible distribution of Applications around the network.

Once again, the Migration process must be planned step by step as the need for co-operation between systems arise and where user benefits are clear. The best way to approach the Migration planning is to link the process to new system requirements. A good example is the evolution of a corporate Message Handling Strategy embracing existing multi-vendor mail systems.

9.3 OSI applications – message handling

A Functional Profile for 'mail' services has been assembled by the CCITT, using a selection of the elements from the upper 3 layers. These run independently over the standard OSI Bearer network profiles which have already discussed. The standard family is known as the X.400 series and the profile has now been adopted by CEN/CENELEC/CEPT as a European standard for message handling.

The X.400 standard is usually implemented in 'postroom' servers, known as 'User Agents' which are distributed around an OSI Bearer network. Messages are entered into the network via a local User Agent and transferred on a store and foreward basis via 'Message Transfer Agents' to other User Agents where the addressed recipients can access them. Figure 10 is a schematic diagram of the arrangement.

The X.400 family provides an excellent basis for a multi-vendor corporate message handling strategy which is totally aligned with the goal of Migrating to OSI Application services.

A typical starting position in a large corporate organisation is shown in Fig. 11, where the user has several freestanding proprietary mail systems. The goal is that all the systems should be able to interchange mail freely and that they should all have access to the external mail services which are used in the organisation's business.

X.400 is the key. Most manufacturers have already declared support for the X.400 Message Transfer Agent (MTA) access profile and many of them have the product available. The Migration strategy should be to introduce a local MTA as shown in Fig. 12 with common access to the public X.400 service. Access to other external non-OSI mail services, such as Telex, can be provided by the local MTA through open system gateways. These external services will increasingly adopt X.400 protocols and this will ultimately standardize the MTA access procedures.

The Migration strategy can be evolved to include distributed MTAs with an underlying X.25 bearer network which provides the routing between the MTAs and access to the public MTAs when relevant.

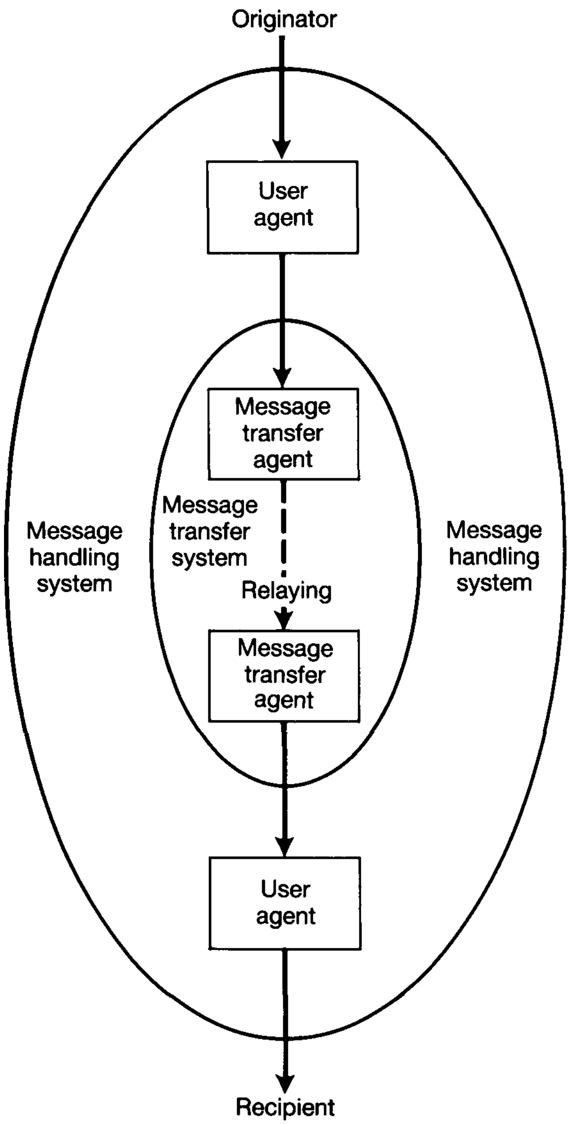


Fig. 10 Message handling schematic

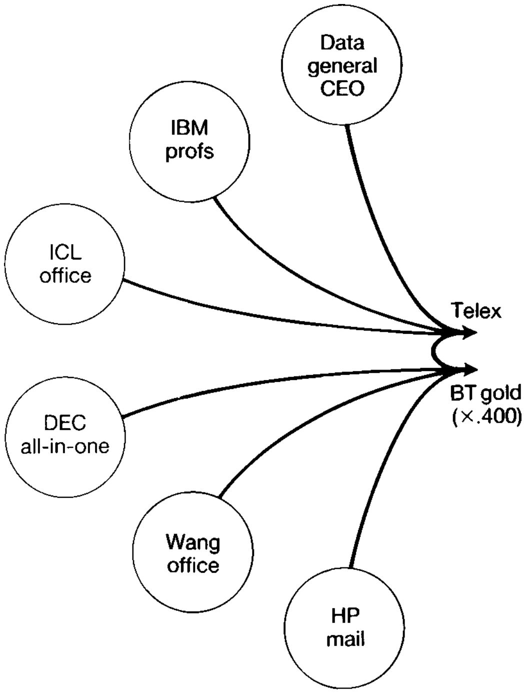


Fig. 11 Message handling – typical corporate problem

These migration steps lead directly to multi-vendor message handling.

10 OSI systems – management and control

Management and Directory services run across all the layers of the model.

Management services provide 'hooks' into all the levels to gather the information which is needed to control both the distributed network and the suites of Application processes. The management standards will handle the selection of the protocols which ensure full interworking within the final connection and also maintain a statistical log of the performance and status of the bearer network elements.

OSI system directories are being standardised to simplify the access to the user by naming the required service. Directory services relate real names and addresses to the logical names and addresses which are independent of the actual location of the services within the network.

Some of these standards are in place and others are being formulated.

The multi-vendor environment will be very difficult to handle without common OSI directory and management structures and the Migration strategy must be planned to intercept these standards as they emerge.

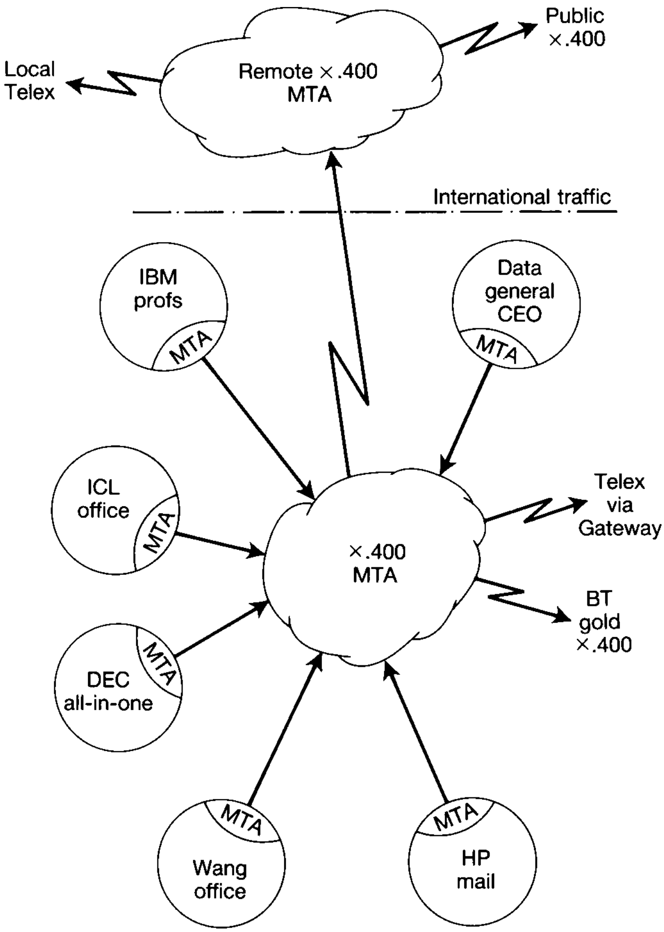


Fig. 12 Message handling – X.400 solution

11 OSI Migration – five steps to success

- Commit to Open Systems.
- Survey and know the OSI Standards.
    - Existing and Planned.
- Survey Suppliers' Products.
    - and Plans.
- Define your needs and Driving Forces.
    - and the Constraints.
- Define a Clear Migration Strategy.
    - Define the End Goal first.
    - Then chart the Valid Steps along the way.

GLOSSARY

Standards Organisations

ISO	– International Organisation for Standardisation.
CCITT	– Comite Consultative International de Telephon et Telegraphie.
CEN	– Comite de Normalisation Europeen.
CENELEC	– Comite de Normalisation Europeen Electrotechnique.
CEPT	– Conference des Administrations de Postes et Telecommunications Europeenne.
SPAG	– Standards Promotion and Action Group (Europe).
COS	– Corporation for Open Systems (USA).
POSI	– Promotion for Open System Interconnection (Jap.).
UK GOSIP	– Government OSI Profiles (UK).
US GOSIP	– Government OSI Procurement (USA).

Other terms

OSI	– Open Systems Interconnection.
MAP	– Manufacturing Automation Protocols.
TOP	– Technical and Office Protocols.
LAN	– Local Area Network.
WAN	– Wide Area Network.
PABX	– Private Automatic Branch Exchange.
ISDN	– Integrated Services Digital Network.
PCM	– Pulse Code Modulation.
CSMA/CD	– Carrier Sense Multiple Access/Collision Detect (ISO LAN Standard 8802/3).
ETHERNET	– Xerox Trade Mark for CSMA/CD LANs.
LOB	– Local LAN Bridge.
ROB	– Remote LAN Bridge.
X.25	– CCITT Standard for Packet Network Access.
X.400	– CCITT Family of Message Handling Standards.
OSPAC	– ICL's Open System Packet Network.

References

1 The OSI Handbook. Author J. Houldsworth. Published by ICL. Copies available free on request from: ICL INFOPOINT, Bridge House North, Putney Bridge, Fulham, London SW6 3JX. Tel: (01) 788 7272.

2 Networking Architectures – J. Houldsworth. ICL Technical Journal, Vol 1, Issue 1, (pp 50–65) (1978).

3 IPA Networking Architecture – J. Brenner. ICL Technical Journal, Vol 3, Issue 3, (pp 234–249) 1983.

A Network to Support Application Software Development

Vince Bodsworth
Cuatro Information Systems Limited, London

Abstract

The development of network technology is constantly promising a new way of working, one where people do not have to travel to work but where they communicate electronically and the work comes to them. F International are pioneers in remote working and have piloted various methods. One which has gained favour is the concept of work groups, located close to where people live, but linked by voice and data networks to other work centres, FI offices and clients' premises. In order to gain the maximum functionality from network services, and easy connectivity to development facilities, it has been a useful exercise to design a communications architecture to provide the framework for implementation and procurement of equipment and services.

Introduction

F International (FI) is a company perhaps better known for its people than for its products or services. This is due to the fact that the majority of the FI workforce are freelancers who work from a 'home base' rather than a corporate office; this was a pattern which was most unusual 25 years ago when FI was founded and it is not that common today. During the 1980s, however, teleworking has become a popular concept and there is no doubt that it will become more and more popular in the future.

Most of the FI workforce are women, leading to the popular misconception that the 'F' stands for female, rather than freelance as is actually the case. There is no doubt that women have had a more pressing need to innovate in ways of working (to allow the simultaneous care of a family and the holding down of a job) but today the male of the species is also finding the flexible work pattern an attractive proposition. Pressures on office accommodation in cities, increasing fuel costs, environmental pressure groups and the simple desire to live not where your work dictates, but where you want to, are just some of the reasons why all the pundits forecast that 'teleworking' is the thing of the future.

Another popular misconception is that FI only supplies contract programmers. While it is true that individual assignment of programmers on contract always has been part of the business, a moment's thought will reveal that this is not the ideal when the workforce is scattered and includes a lot of part-timers. Most people taking on contractors want them in their own office, and rather assume that they have 100% of their time. The ideal sort of work for the type of resource FI has at its disposal comes in large chunks which can be split up into parcels, and distributed to wherever it is going to be done.

A desire to take on projects, or provide services, of ever increasing size and complexity has many consequences, which include:

Specialised, industry oriented marketing.
Project management.
Emphasis on quality.
Structured methodologies.

Moreover the type of 'service product' offered must increasingly diversify to include not only bespoke systems development, the main FI service, but also maintenance, semi-customised systems (kernels), rapid development methods, etc. if the maximum added value is to be achieved.

Above all, to achieve anything like the cohesion of an office based team requires excellence in communications so that the members of a widely scattered workforce can remain at all time in touch with each other.

Network requirements in FI

Program development, originally, revolved around the use of coding sheets, punched cards, and computer lineprinter output. Remote working, when all compilations and test shots were a batch process, was almost the same as local working, with just the delay of the postal services making any difference. Today all phases of the development, from business systems analysis, through design and programming, to testing and handover, involve the use of computers. Increasingly Personal Computers can provide many of the facilities for development and testing of programs, which previously had to be done on the target machine. In some areas, notably editing of programs and text, and in systems analyst's tools, Personal Computers are demonstrably superior to the mainframe or minicomputer.

As telecommunications use has expanded in DP establishments facilities have also been available to FI personnel working on projects. FI have only been able to exploit telecommunications by either using the PSTN (public switched telephone network) or by actually travelling to client sites, to use their facilities. As an increasing amount of the development work requires interaction with either the 'host computer' or a suitable development facility, it has become increasingly evident that FI must gain access to suitable network services and development facilities, if it is to maximise the advantages offered by a freelance workforce.

At the same time as the needs of the software development function have expanded, so have the needs of the administrative and management staff in FI. Unlike other companies many of the administrative and clerical staff are also home based, and those who are not work from small offices located close to major cities.

The main requirements of a network service fall therefore into two major categories:

1 Those to support the development process from office, home or client sites.
2 Those to support the administration and management of the company.

Both of these split down into a number of facilities which in some cases are in conflict with each other. A major potential conflict is always that of cost and this is particularly important when many of the sites to be connected are individuals' homes, and when error-free transmission is required. The easy solution of leased facilities is not normally cost effective.

Facility requirements

The major facilities required are:

Electronic mail for office and home based staff.
File transfer for development staff.
Shared databases for a variety of business and development functions.
Data capture for business and management accounting functions.
Terminal access to in-house development facilities.
Terminal access to customers' development facilities.

These facilities must exhibit the following attributes:

High levels of security.
Low cost of access from home or office.
High data integrity on all connections, but in particular on dial up links.

A number of these services have been run as pilots on an individual basis, the results of some of these pilots are summarised in this paper. As a result FI embarked on a strategy which will enable them to be largely independent as far as telecommunications go. This will require a network architecture which is tailored to meeting their own specific facility needs and requirements.

Overview of the Network Architecture

In common with other network and system architectures FI Net, to give it a name, uses a layered structure to separate clearly distinct levels of functionality one from another, and to define common interfaces between the layers. This enables replacement of components of the network at lower levels, without the necessity of simultaneous replacement of those at a higher level. In most cases higher level components can also be replaced without disturbing lower levels, but there could be limitations if the higher levels demand increases in capacity from those underlying.

The basic scheme follows that of the OSI model which offers an effective guideline for the implementation of distributed systems, and also which will allow the use of OSI conformant components as they become available commercially.

The OSI model divides a network structure into 7 layers, from the top:

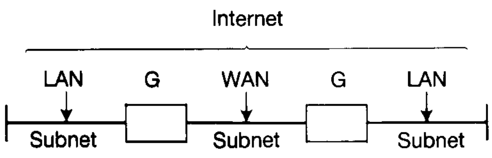
Application	Network Service, e.g. Mail service
Presentation	Data Format Conversion
Session	Programmatic Interface
Transport	Reliable End-to-End Communications
Network	Subnet- to Subnet- Communications
Link	Communications line protocols
Physical	Communications line hardware.

The Application layer concerns the provision of the service required, for example, an electronic mail service.

At the presentation level the problem of differing data formats from system to system is dealt with. Conversion of data format will be undertaken by the system as necessary. Both application developers, wishing to develop distributed applications, and system programmers implementing the functions of the FI network need access to a programmatic interface, provided at the session level.

The Transport, Network, Link and Physical levels are all part of the system which has traditionally been regarded as the communications system, and it is there to get the data reliably from one point to another.

The FI network will span a fairly wide geographic area and incorporate several network technologies at different points. Typically even a point to point connection will involve local area networks and wide area network technologies as shown below.



6 designates a gateway. Where the protocols across the internet are identical these gateways could simply be bridges (for example if both LANs were ICL

Oslan) but in the more general case protocol conversion will also have to take place.

The network layer is responsible for determining which path a particular message should take through the internet.

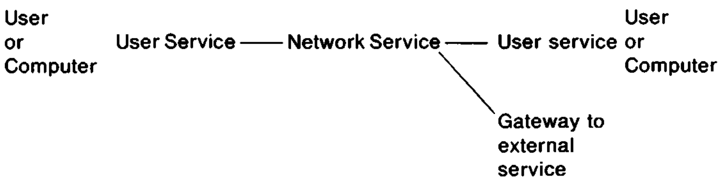
The link layer is responsible for transmitting the data between computers that are physically connected, and the physical layer is concerned with the actual type and characteristics of the wires (copper, optical fibres, or whatever) connecting the systems.

At one end of the network is the user (sometimes perhaps a computer program, rather than a person) and at the other the service provider. In fact three classes of service can be distinguished in FI net.

External services.
User services.
Network services.

As far as is possible, the users will perceive that they are connected to a service, rather than to the network itself. For example on first startup (or connection over a dial up) the user would be connected to a menu service. This would define the other services available to that user. The services should be easy to use for the class of user. Those available to all users will normally present a menu interface, although if accessed via a PC, the interface could be more WIMP oriented (Windows Icons Mouse Pointers). Programmers, on the other hand, often prefer a terse command driven interface, at least when they have become accustomed to the service. As far as is practicable the interface for the user will be presented by the users workstation rather than some remote computer; this is certainly true for network services. External services and user services, on the other hand, are typically development computer services such as VME MAC, or MVS TSO, or transaction processing services, and once connected to these the user is at the mercy of their command structure. In some cases they may be services such as external mail services. In these cases it is highly desirable that the network service should be able to 'gateway' into the external service to make it transparent to the end user. (In the specific case of Email, the universal provision of X400 message handling should make this transparency quite automatic).

Service structure



At either end of the network there is either a computer or a user. Some network services connect users together, e.g. electronic mail, and others connect users to computers, e.g. terminal emulation on to a mainframe. Yet others connect computers to computers, e.g. file transfer.

The network service will of course itself, in fact, be supplied by either special purpose computers or general purpose computers with special software. For the sake of economy the same computers can perhaps undertake both user and network services when the network is small. Later on the services can be segregated.

The network service

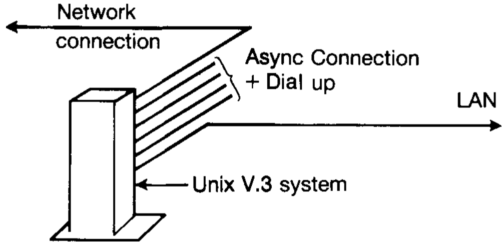
The network service will be provided by a set of systems operating under software conforming to the Unix System V.3 interface definition. In due course, as the standard develops, Posix could take the place of System V. Interconnection between nodes utilises ISO standard protocols on public or leased services. These will utilise HDLC or Async with error correcting modems, for leased connections, and X.25 where connection is via Public Data Networks.

At each of the nodes several workstations will be locally connected, using either a LAN, or Async connections, and remote workstations will use either the PDN, or direct dial up to gain access to the node.

It is imperative that a reliable transport service be utilised over all links, and the local dial up link will use error correcting modems operating with the MNP or similar link level protocol. An added advantage of MNP, at the higher classes, is that it supports negotiation of connection speed (up to 9600 bps) and data compression techniques improve data throughput by a greater factor than the protocol overhead. It is possible therefore to get performance similar to direct connection for those users who need it, while maintaining low cost (300 bps and 1200 bps) access for casual (e.g. Electronic mail only) users. (Note: MNP autoanswers at the lower speeds, thus allowing older or less capable modems to connect, and negotiates the higher level protocols and higher speeds with those modems that have the capability.)

Network node

A network node will be installed in each of FI's regional offices, and at new



work centres to be set up close to groups of FI people. The choice of Unix V.3 systems will enable a choice and span of systems from 4 to over 100 users per node.

Workstations

For reasons of economy, and because of the wide range of hardware and software available, the standard workstation is an IBM PC or PC/AT compatible system.

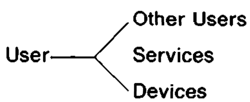
Naming of users and services

A single naming system will be used for both users and services, in a three level hierarchy. The three levels may be used for any purpose but, for example, could correspond to:

Division.
Physical location.
Name.

Any particular user has a scope, associated with their name, which defines the subset of the total which they may access. For example a particular clerical or administrative user could have a scope which corresponds to all the other users of the system, the electronic mail service, and a local printer. This user could send and receive mail from any other user, and access the local printer for printouts.

Another, development user, may have access to all other users, the mail service, the file transfer service and a client-supplied development system.



Security considerations

Security aspects of the network are clearly of paramount importance for a company servicing external clients. One of the primary objectives is to link up F International workers with services on client computers. Some of these workers will be using client premises, some F International work centres and others will be using the PSTN to dial up from home or other locations. FI have adopted a three level security structure so that only authorised persons can gain access to any facility.

The three levels are:

Physical access.
Network.
Service access.

Physical access

In office locations physical access is controlled by a combination of Receptionist/visitors badges, and card controlled access to areas where terminals are located. Only authorised persons, that is someone bearing a current card, can enter areas where terminals with access to the network are located.

Workers who are working from home, or other uncontrolled locations, will use dialled access to the network where a dial back facility is used to control physical access. The unique combination of user name, telephone number and password is used in the following process:

User dials in.
User logs on and supplies password.
Access control computer checks authenticity and looks up phone number.
Access control computer disconnects.
Access control computer redials the caller.
Access control computer optionally asks for a second password.

Network transport

Network transport security controls the ability of the user to access any particular node or port on a node, in the network. This is effected by showing any particular user a virtual network, on a menu. The virtual network appears to the user as a list of named facilities or services. An additional password is required by the network to gain access to these facilities. In fact the network can also be controlled in command mode, rather than menu, to speed up access for those users familiar with the system. This does not extend the access capabilities of a user, it merely allows access via a short command (e.g. call VME) if the users know where they want to go.

Service access

Having gained 'access' to a terminal, and by suitable identification and password a transmission service across the network (for some user this may be simply access to their local node) the user is required to again provide name and password to the service desired. This third level applies only, and then only potentially, to External and User Services. Network services are secured only up to the network level.

In effect the network will be providing a virtual circuit into a computer, which may be able to offer various services. It is up to the operating system and the services to police these accesses. The FI network can of course assist, by ensuring unauthorised users are not even allowed in, and by limiting other users to particular physical ports. This helps the mainframe or minicomputer operating system to further limit the facilities available.

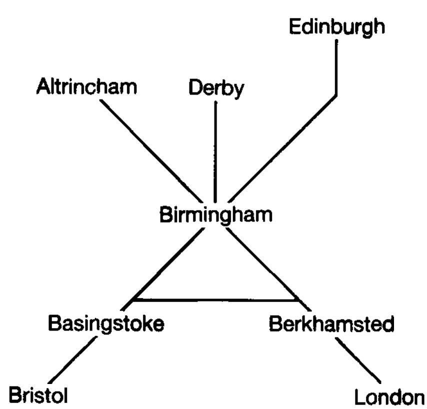
Geographical coverage

FI is represented in almost every corner of the country, at least from the Mull of Kintyre to Dartmoor if not from John O'Groats to Lands End. This makes for a considerable challenge in terms of network implementation and coverage. These outlying areas are clearly only going to be economically covered by use of dial up access and experience has shown that some of the more rural telephone exchanges impose limitations on the line speeds that can be supported. It is in this area that the higher level error checking protocols will be very important.

The main FI offices are located as follows:

Berkhamsted, Hertfordshire	Head Office
Ditto	Regional/Sales Office
Basingstoke	Regional/Sales Office
Birmingham	Regional/Sales Office
Bristol	Work Centre
Derby	Work Centre
Altrincham	Work Centre
Edinburgh	Regional/Sales Office

The main network nodes will be established in the Regional Offices, and will also serve the sales centres set up in the same locations. The work centres are equipped with a complete set of development workstations, and also support dial-up access from elsewhere. The overall physical connections planned are illustrated in the diagram below:



Other Work Centres can be added, as required. It is envisaged that each major node will be able to support up to 16 work centres.

Implementation

Initial implementation has consisted of a number of pilots, and the provision of a comprehensive electronic mail service from a Value Added Network service supplier. The pilots have included access to central Unix V and DEC Vax development facilities, as well as gateways into ICL and IBM mainframes for specific projects undertaken in 1987. Full scale implementation is planned for 1988, but, having established the overall architecture, it will be possible to respond tactically to business demands, while still maximising the long term benefits.

Universal Communications Cabling – A Building Utility

A.V. Flatman
Manager LAN Technology, Systems Engineering, ICL Network Systems

Abstract

This paper examines a structured approach to building cabling, and reviews the technological options available. A particular cabling architecture is considered which takes account of user needs, both current and future. The support of industry standard communications systems and interfaces is presented, and the evolution of functionally integrated services is explored.

The paper concludes with a review of related standards initiatives and some comments on the feasibility of 'open systems cabling'.

1 Introduction

Cabling for information systems – data, voice, text and image – is now as important for the efficient functioning of a building as mains wiring and plumbing. As with the other services, interruptions can be extremely expensive. Poor quality of service, due to lack of design foresight, can threaten the very existence of an organisation in today's highly competitive business environment.

The IT user community will naturally wish to embrace Open Systems Interconnection as a means of gaining greater independence through multi-vendor procurement. This will be made possible in practice by the publication of OSI Standards for information networks, communications protocols, and functional profiles. In terms of approved International Standards however, the number of possible options increases dramatically within the lower levels of the OSI model. Local Area Network Standards exist for CSMA/CD, Token Bus and Token Ring access methods; each targeted at general purpose commercial and industrial data applications. If the emerging variants of these standards are taken into account, the marketplace is in grave danger of confusion arising from the plethora of physical networks and end systems operating at different bit-rates, over all topographies and media types imaginable. One may ask if standards organisations reflect the wishes of users, or manufacturers; and is it in fact possible to identify a unified approach to physical networking?

It is indeed possible to define a common cabling infrastructure which is capable of supporting industry standard LANs and point–point communications links (i.e. V24). The architecture adopted for this solution actually corresponds to the traditional approach used for circuit-switched voice, and therefore represents the foundation of a unified building cabling scheme. The adopted scheme must also take account of future requirements, which are expected to include functionally integrated services (e.g. ISDN), and higher bandwidth applications such as high-definition image, graphics or video. In order to make the cabling scheme 'futureproof', careful choice of transmission technologies will be necessary.

A universal cabling scheme will provide the tenant(s) of a building with the technical and commercial independence he needs. Correctly applied, it will also facilitate organisational freedom; and thoughtfully designed, will accommodate communications requirements through the 1990s. A detailed study of this subject has been made recently by Camrass and Smith[1].

2 Architecture

The building communication system infrastructure must be capable of supporting long term networking solutions which are expected to be highly distributed, and at the same time be compatible with existing and intermediate requirements. The adopted cabling architecture must therefore exhibit sufficient flexibility to support the graceful evolution of network solutions. It is also important to accommodate different rates of change within the same cabling system.

A conventional 'tree and branch' topography provides a powerful solution to the problem. The architecture shown in Fig. 1 comprises a number of 'Local Cabling Zones', each interconnected by a single 'Trunk'. The points of interconnection are known as 'Nodes', where each Node provides the key to network flexibility. Nodes can contain active (or intelligent) functions, or in their simplest form be a point of passive interconnection between the 2 physical cabling domains. The approach shown in Fig. 1 is essentially a

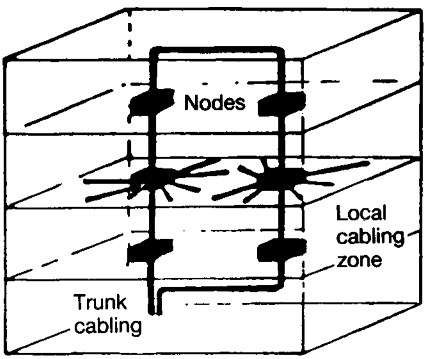


Fig. 1 Cabling architecture

'ducting' architecture. Whilst this defines the geographical routing of each cable section, the overall logical configuration may be ring, bus, star, or point–point. The latter will be determined at a Node, or a defined set of Nodes.

3 The Local Cabling Zone

The Local Cabling Zone covers a convenient and manageable section of a building, which in practice may be an entire floor of a high-rise block. If we assume the floorspace of a typical high-rise building to be 800 sq.m per floor (say 40 m × 20 m), then we may have up to 100 employees present based on current legislation relating to office workspace. Each of these 100 employees would have a telephone, and most are expected to own a data terminal in the foreseeable future.

The voice and data service requirements of the majority of users can be comfortably accommodated by relatively low bandwidth channels (for sizing purposes I will assume that an average user requires 64 Kbit/s total bandwidth). In deciding how many cables will be required to connect each user to the information network, account must be taken of the fact that voice and data services will remain separate for many users for some time to come. A small proportion of users are expected to require more sophisticated terminals/workstations on their desks in the immediate future, however this may increase dramatically in certain market sectors due to a demand for high definition image/graphics and even video services, plus an increase in disk-less workstations. The latter could have a dramatic effect on the choice of medium for Local Cabling.

The selection of an appropriate medium for Local Cabling is somewhat dominated by the massive installed base of unscreened twisted-pair cable[2]. Although the technical specification of this cable type varies from country to country for telephone systems, it is a comfortable technical solution for operating 64 Kbit/s over a maximum path length of 50 m (corresponding to a Node positioned centrally within the high-rise building example quoted above). It is hardly necessary to explain why it also represents the most economically viable option. Telephone-grade cables have additional advantages in terms of bulk (cross-section and weight), and flexibility.

Unscreened twisted-pair cables are not particularly efficient for high bit-rate transmission, and substantial research is being done to establish the fundamental limitations of this medium (more about this later).

It is essential that Local Cabling has a long installed life; that is a minimum of 15 years, in order to provide a stable user environment (they do it for telephones!). It is therefore vital to take full account of the growth in bandwidth over the installed lifetime of the Local Cable.

4 Saturation vs Dedicated Cabling

The traditional approach to communications cabling within a building has been based on the use of 'dedicated' connections. The cabling system has in this case been built around a 'snapshot' in time of the locations of each IT product. The number and routing of dedicated cable sections will change according to the population of resident users.

An approach based on dedicated cabling is understandably tempting owing to its low entry cost, however the subsequent cost of ownership of this type of system is currently subject to extensive debate within the user community[3,4]. The cost of installing and re-routing dedicated cables within the working environment is now recognised as a major consideration. The 'disturbance' factor must also be taken into account in the cost equation.

Based on the growth of IT in the office, and the need for users to be mobile, a more flexible approach to Local Cabling is emerging – Saturation Cabling. Saturation Cabling simply provides network 'Access Points' which are always within convenient reach for a user, wherever he is situated within the building. The 'saturation' approach is somewhat akin to the local mains power distribution system, where sockets are provided at regular intervals throughout the workspace.

The initial cost of an IT network based on 'saturation' cabling will clearly be higher than that based on 'dedicated' cabling, however studies indicate that cost of ownership is equivalent after a period of 5 years or less for larger networks[5]. The saturation cabling approach then becomes more economical, as shown in Fig. 2.

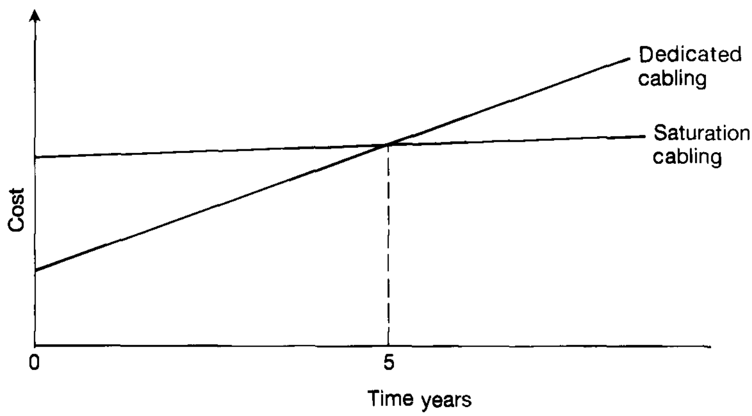


Fig. 2 Saturation vs dedicated cabling costs

In fairness to the Telecommunications Industry, the saturation cabling philosophy is being used increasingly for telephone connection. Perhaps the Data Communications fraternity should learn from this.

5 Local Access Connection

If an Information Network 'Access Point' is to be ubiquitous, it must be inexpensive. The Access Connection must also be simple and clearly labelled. Separate connectors will be required for voice and data services in the shorter term, with the national standard telephone jack providing the connection for voice. Ideally, a second co-located connector should provide a single 'Access Point' for the remaining communication services. It is therefore essential that this second point of connection becomes recognised as an industry standard too, without which we shall have lost the major benefits of a 'universal' cabling scheme.

The choice of connector for non-voice services must take account of the growing requirement for higher bit-rate transmission. A practical solution is to adopt a standard faceplate and two standard connectors, one for multiple twisted pair-cable, the other for optical fibre. The former would represent the short-medium term solution, whilst the latter could be phased in progressively, as required. Fibre optic interconnection is a reasonably mature technology today, albeit expensive.

6 Nodes

Nodes provide the communications cabling system with the flexibility its users require. In its simplest form, a Node will manage the cross-connection between Local and Trunk Cabling domains (Fig. 3). This represents a highly structured way of enabling dormant cable sections to be brought into service in the Local Zone, existing users to be re-located, or even disconnected.

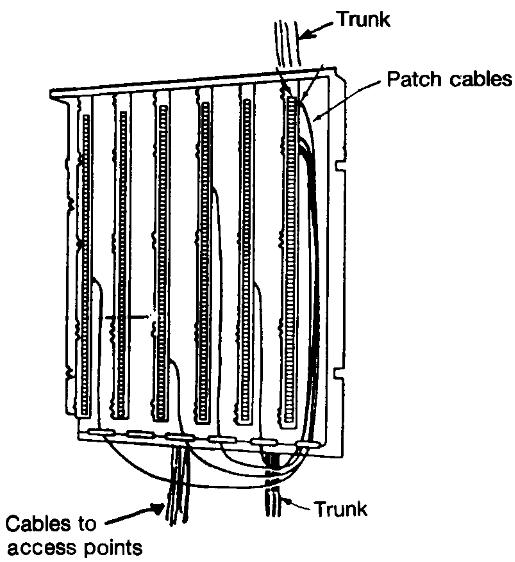


Fig. 3 Example of cross-connect frame at a Node

Nodes also provide a convenient junction between the local working environment and the information trunk, or backbone. This 'junction' is important as it decouples the evolution of technologies applied in each of the cabling domains, thus allowing more frequent (and radical) changes in the Trunk whilst maintaining stability in the local user environment.

A Node may also possess greater functionality. Signal amplifiers, repeaters and multiplexers may be located at a network Node – also bridges, gateways and shared IT resources (servers). The Node will then become a logical 'junction' linking different protocols and transmission speeds.

Owing to the strategic importance of the Node, it is recommended that a small secure room (say 2 m × 3 m) be provided for cross-connect frames and equipment. Returning to the previous example of the high-rise office block, such a room would be required somewhere on each floor.

7 The Information Trunk

The Information Trunk must generally support the total communications requirement of a building. An estimate of Trunk capacity may be obtained from the earlier example for a high-rise office block, where each floor (Local Zone) would correspond to 100 users × 64 Kbit/s, or 6·4 Mbit/s peak. Multiplying by, say, 15 floors would require an aggregate of 96 Mbit/s peak. This may appear high, but remember that this does not take account of the increasing application of high-definition image/graphics workstations, or the transmission of real-time video!

Fortunately, it will be easier in practice to evolve technology applied to the Trunk system rather more rapidly than that used in the Local Zone. The life expectancy of Trunk transmission systems can be 5 years or less, compared with 15 years or more for the Local Zone.

In the short term, Trunk systems will comprise separate cables for voice and data services. A traditional PBX Trunk would be made up of individual

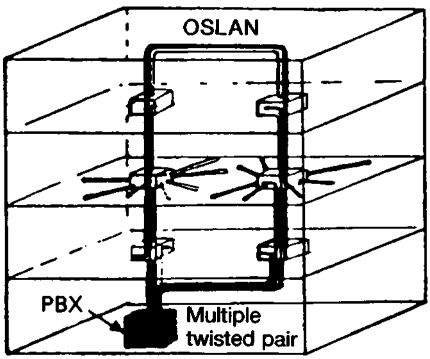


Fig. 4 Trunk system with PBX and LAN cabling

telephone cables which would be cross-connected through to corresponding lines in the Local Zone. Today, the majority of data services would be provided by an industry standard LAN, the most successful being 'Ethernet' (ISO 8802-3 known in ICL as OSLAN). Figure 4 shows how these 2 cabling systems are overlaid, side-by-side, to form a Trunk.

In the longer term, Trunk systems must not only cater for greater bandwidths, but need also to accommodate the functional integration of communications services (more on this topic later). Both requirements will inevitably lead to the widespread application of fibre optics technology.

8 Transmission technologies

There are 5 basic cabling technologies:

- Twisted-Pair, unscreened
- Twisted-Pair, screened
- Coaxial, baseband
- Coaxial, broadband
- Optical Fibre.

It is worth reviewing the relative merits of each of these candidates before formulating a view as to which technologies are most appropriate for Local and Trunk cabling.

Twisted-Pair, unscreened: Traditionally used for telephone cabling. Small cross-section, lightweight and extremely flexible and inexpensive (pence per metre). Low cost connectors and termination (insulation displacement). Inexpensive, balanced transmission links with a reasonable level of common mode noise immunity, particularly at lower frequencies. Capable of supporting 1 Mbit/s over approximately 1 Km.

Substantial research is currently under way to determine the capabilities of unscreened twisted-pair cable in the 1–10 Mbit/s range[6,7]. Various independent measurements indicate that 10 Mbit/s looks technically feasible over cable lengths of 50–100 metres. Radio frequency emission energy is reported by many to be within the FCC classification A (commercial) limit. Susceptibility to electromagnetic noise may be difficult to achieve in some environments (a data service will require a bit error rate of no worse than 1 bit in 10^8), however careful design of transmission circuits may overcome any difficulties encountered at higher bit-rates.

One of the limiting factors of unscreened twisted-pair cables relates to longtitudinal variations in impedance due to dimensional differences between the individual wires and the variation in twist-rate along the cable. This imbalance is directly related to both emitted and injected electromagnetic energy. This limitation becomes a particular problem with the multi-pair cables and shared cable ducts (due to signal crosstalk).

Twisted-Pair, screened: A metallic braid is added to the twisted-pair cable to act as an electromagnetic screen. This screen will attenuate emission and injection energies, and therefore greatly reduce crosstalk, however the screen component must be correctly terminated at each end of the transmission line. The addition of this electromagnetic screen will add noticeably to the weight and cross-section of the cable. A screened twisted-pair cable is less flexible and costs significantly more than its unscreened counterpart. An additional cost will be incurred in the preparation of the screen connections prior to final termination. The connector must provide a reliable contact for the screen component; this will add noticeably to the size and cost of that connector.

There are no technical difficulties in operating screened twisted-pair cables at data rates of tens of Mbit/s over 100s of metres. Emission and injected energies are effectively contained.

Coaxial, baseband: A coaxial cable will outperform a screened twisted-pair cable of the same outside diameter by approximately 2:1 when transmitting baseband information. This means that it can either operate twice as fast, or twice as far. Size for size, its cost, weight and flexibility are generally equivalent. The two cable types are also very similar with regard to electromagnetic compatibility. Cost of termination of coaxial cable is marginally less than screened twisted-pairs due to the physical compatibility between coaxial connector and cable.

Coaxial cables are capable of operating at very high frequencies (hundreds of MHz), however baseband transmission will generally confine signal energy within the spectrum DC – 50 MHz (see Fig. 5). The cable screen required to contain RF emission and attenuate electromagnetic noise injection is usually a single braid of stranded copper wire. Thin metallic screens may be added to further improve the noise performance of a cable, however this will render the cable structure somewhat less flexible. An example of a braid + solid

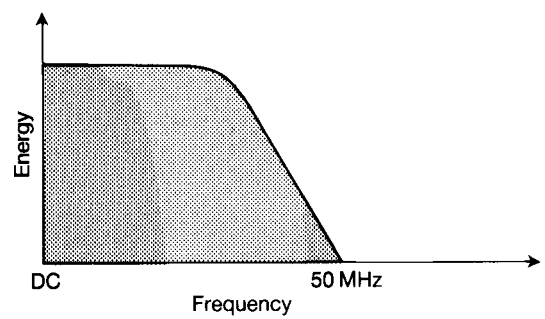


Fig. 5 Energy spectrum of baseband transmission

screened coaxial cable may be seen in Ethernet (which actually employs 2 braids plus 2 aluminium tapes)[8].

Coaxial cables are generally less complex than screened twisted-pair cables, and usually cost less. Signal transmission via coaxial cable is relatively simple and inexpensive.

Coaxial, broadband: Owing to their ability to operate at very high frequencies, coaxial cables can be used to transmit broadband signals. As the title implies, this technique uses a broad slice of the frequency spectrum, typically 300 MHz, and possibly up to 450 MHz in practice. Broadband signalling conveys a multiplicity of separate information channels, as shown in Fig. 6. This approach is the basis of CATV systems, which are commonplace in the USA and many other countries.

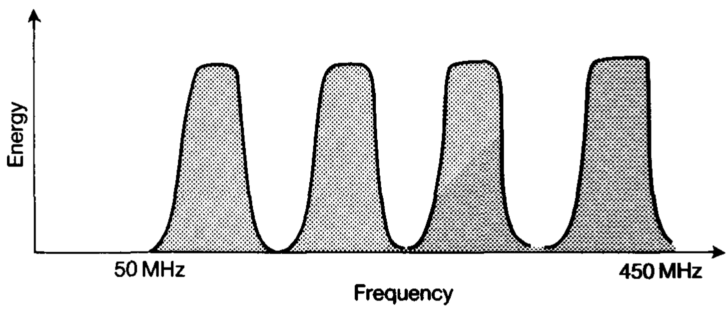


Fig. 6 Energy spectrum of broadband transmission

In addition to the efficient utilisation of cable bandwidth, coaxial broadband transmission can achieve generally longer links than coaxial baseband systems. The latter is achieved by the application of conventional radio techniques (e.g. superheterodyning, automatic frequency control). Broadband cables generally require solid screens to contain the high frequency energy, however this requirement is relaxed to some extent by the use of very low level transmission signals. Once again, solid-screened cables can be quite inflexible.

The support of multiple, concurrent services, integrated within a single cabling system, is particularly attractive. Broadband transmission could, in practice, carry several real-time video channels, data, voice, etc. The application of broadband signalling to the interactive user then creates a need for either a separate transmission system for the return-path (dual cable), or alternatively the allocation of part of the single cable frequency spectrum for return-path transmission. Depending upon the ratio of 'outbound' and 'return-path' bandwidths, the single cable broadband system will be organised as 'mid-split' in the case of an evenly-balanced system, or 'high-split' in the case where more outbound bandwidth is required. The necessary frequency translation is performed at a 'head end'.

Whilst the unit length cost of broadband cable is very similar to baseband cable, the economics associated with a broadband network can be quite startling. A frequency translating 'head-end' may cost in the region of £1000–£2000, and modems (the means of accessing the cabling system) cost approximately £300–£1000 each.

A more detailed review of this technology, and a useful survey of products is referenced[9].

Optical Fibre: A great deal has been published in recent years on the merits of optical fibre as a transmission medium for long-haul communications networks. Theoretical estimates of the capabilities of silicon-based fibres are 300 Km at a data rate of 1 Gbit/s[10]. Some of the longest unrepeatered distances yet achieved with today's commercial optical fibre systems run only about 60 Km (transatlantic TAT 8). As a taste of things to come, fluoride-based optical fibres are expected to be capable of supporting bit rates upward of 1 Gbit/s over distances in the neighbourhood of 3600 Km.

Optical fibres are also finding a niche on a more local scale; an interesting example being the high-speed computer mainframe network known as Macrolan[11]. A frequently posed question is: 'When will optical fibre be commonplace within normal commercial or industrial premises?' Before we attempt to answer this question, it is worth reviewing the cost/performance figures relating to optical fibre-based communications within the confines of a single building (or campus).

An optical fibre is formed by 'drawing' 2 concentric components of optically-pure material into a single strand measuring approximately one-tenth of a millimetre in diameter for silica fibres, and one millimetre for polymer fibres. Optical energy is then simply propagated within the low-loss central 'core', and contained by the outer 'cladding'. The 3 types of optical fibre in common use today are shown in Fig. 7.

Step-index fibre is available in polymer, silica, or a combination of both materials (polymer-clad-silica). The most fundamental limitation of the step-index technique is bandwidth, being typically 30 MHz Km (bandwidth distance product). This limitation is due to the dispersion of multiple light 'modes' propagating along the optical transmission line. A novel way of overcoming this limitation is to grade the refractive index of the core component as shown in Fig. 7. This has the effect of increasing the propagation velocity of the higher order light modes as they traverse the peripheral regions of the fibre core (velocity is inversely proportional to refractive index). Graded-index, multimode silica fibres possess bandwidths in the region of 500 MHz Km. The 3rd type of optical fibre is known as 'single-mode', where the diameter of the core component is reduced to a mere 8 micrometres in order to selectively propagate axial light modes. The bandwidth of a single-mode silica fibre is in excess of 1 GHz Km.

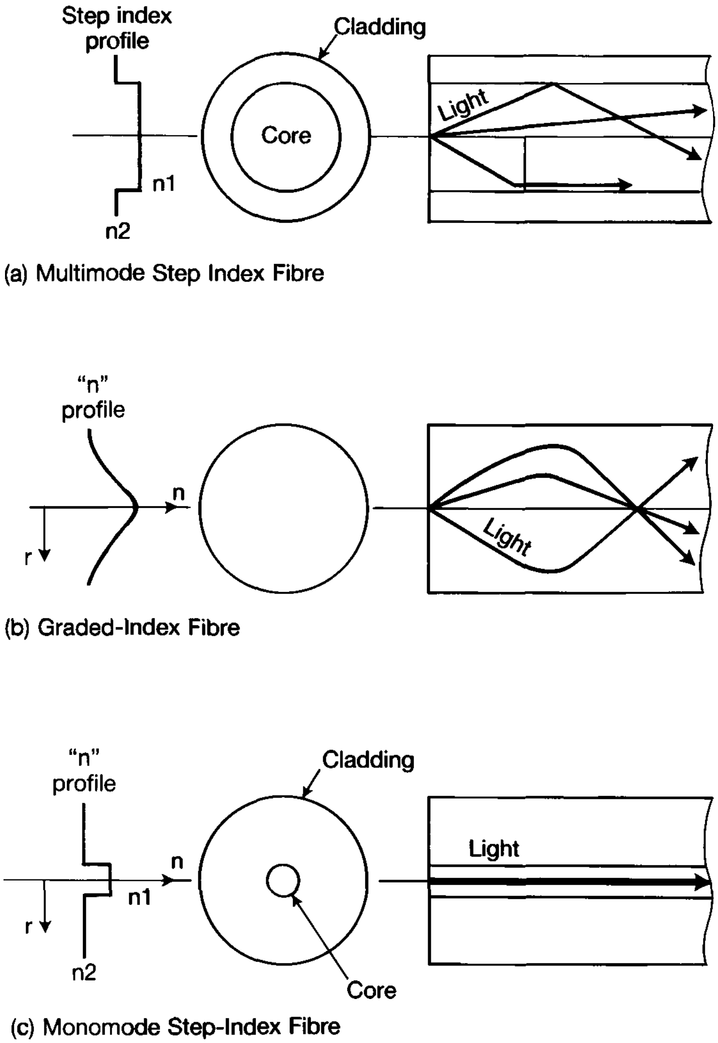


Fig. 7 Optical fibre types, showing (a) Step-index multimode, (b) Graded-index multimode, and (c) Single-mode structures

An additional limitation of optical fibre transmission relates to 'chromatic dispersion' due to the spectral bandwidth of the light source (this is analogous to 'group delay distortion' in electrical transmission). Semiconductor lasers are superior in this respect due to their inherently fine 'linewidths' (1 nanometre or less), compared with Light-Emitting-Diodes which can possess linewidths up to 150 times greater.

With costs of optical fibre falling to less than 10 pence per metre, optical cables have already become economically competitive with other media types. Optical fibres not only provide greater bandwidths over longer distances, they are also intrinsically immune to electromagnetic noise and do not radiate. Owing to the non-conductive nature of the materials used to make an optical cable, they are also electrically safe. Based on the foregoing, it may be difficult to understand why optical fibres have not yet found extensive application in Local Networks; the reason for this is related to the current high cost of optical transceivers and connectors, particularly for single-mode fibre systems.

Industry standards are emerging for optical fibres and connectors. Several graded-index fibres are subject to international standardisation, these being 85/125, 62·5/125 and 50/125 (core/cladding diameters in micrometres). There is little difference in overall cost and performance of these 3 candidates, so the user community will decide for itself which will dominate. Having said this, there are a number of LAN Standards projects which are currently promoting the use of 62·5/125 as a preferred medium. 2 types of single-way optical connector are also being standardised – FSMA (based on the existing BNC format) and S/T. In addition to this a low-profile duplex connector standard is also emerging from the ANSI FDDI project. Other emerging standards for optical LANs are expected to adopt the latter.

Finally, it is worth taking account of the evolutionary potential of optical fibre technology. Today's optical fibre transmission systems are generally based on a single wideband channel and 'direct detection'. This particular approach has been likened to the 'spark gap' transmitter used in the early days of radio. The application of 'coherent' techniques[12] will enhance the total bandwidth capacity of a single-mode fibre system from 1 Gbit/s to upwards of 100 Gbit/s in the not too distant future.

The main attributes of each of the media considered above are presented in Table 1 by way of a summary and comparison.

Table 1 General comparison of cabling technologies

	UTP	STP	Coax (base)	Coax (broad)	Fibre
Bend radius	0·5 cm (3 pair)	7 cm (2 pair)	7 cm	10 cm	7 cm
Cross section	0·15 cm^2	0·8 cm^2	0·8 cm^2	0·8 cm^2	0·3 cm^2
Weight	30 Kg/Km	90 Kg/Km	90 Kg/Km	100 Kg/Km	20 Kg/Km
Bandwidth/100 m	10 MHz	25 MHz	50 MHz	300 MHz	10 GHz+
Bandwidth/Km	1 MHz	2·5 MHz	5 MHz	300 MHz	1 GHz+
Cost/m (1988)	5p	50p (2 pair)	30p	40p	<10p
System Cost '88	low	low	low	high	high
System Cost '91	low	low	low	high	med
RFI	limited	good	good	OK	none!
EMI	limited	good	good	good	none!
Evolution capacity	nil	little	possibly broadband	900 MHz	1 Gb/s–100 Gb/s+

9 Supporting standard LANs

The support of industry standard LANs via the cabling architecture presented in earlier sections of this paper is discussed below. CSMA/CD, Token Bus and Token Ring LAN Standards are considered.

CSMA/CD: 10 Mbit/s CSMA/CD, frequently referred to as Ethernet, is currently specified to operate over a multi-drop coaxial cable bus with baseband signalling[8]. Several techniques have since been established to operate this access method over alternative media. The important aspect to note with all alternative approaches is that operation is 'transparent'; specifically, the end system Attachment Unit Interface (AUI) is observed, and there is no modification to the transmission protocol.

Perhaps the most interesting of these alternatives is the operation of 10 Mbit/s CSMA/CD via star-configured, twisted-pair cable. Here, each end system is attached to twisted-pair cables through a local transceiver unit, as shown in Fig. 8. The twisted-pair cables are then routed to a common point, where they are terminated by an active hub. Dual twisted-pairs are generally

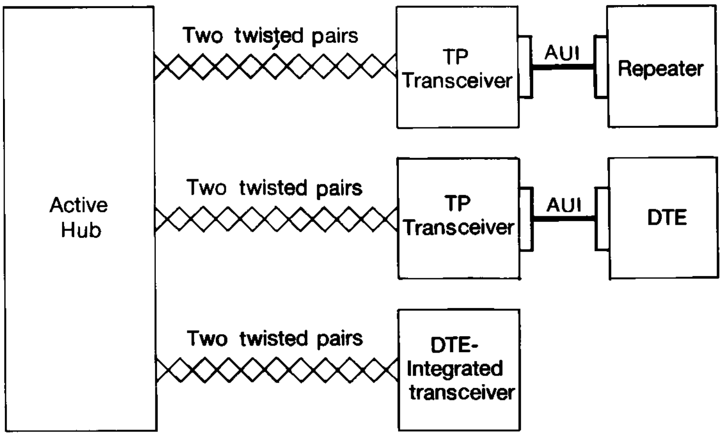


Fig. 8 10 Mbit/s CSMA/CD LAN via twisted-pair cable

used; with one pair for transmission, the other for reception. Collisions are detected and indicate centrally by the hub unit itself. The application of screened twisted-pair cables is fairly well understood, and generally accepted as a comfortable technical solution[13]. The applicability of unscreened twisted-pair cables is now being investigated, and indeed is already subject to many claims and much enthusiasm. Several US companies feel sufficiently confident about the technical feasibility of the latter to have announced supporting products during 1987. The radial distances anticipated using unscreened twisted-pair cables are in the range 50–100 metres.

Much attention has been devoted to the use of fibre optics to support 10 Mbit/s CSMA/CD LAN. Once again, a dual cable, star-configuration is used based on either active or passive hubs, as shown in Fig. 9. Active hubs provide electronic signal regeneration and central collision detection and indication[14,15]; whereas a passive hub will simply act as a logical 'mirror' to provide an optical broadcast function[16]. Collision detection remains distributed in the case of the passive hub. Link lengths will generally be constrained by the system propagation delay allowed by the CSMA/CD transmission protocol, rather than that imposed by cable attenuation or bandwidth.

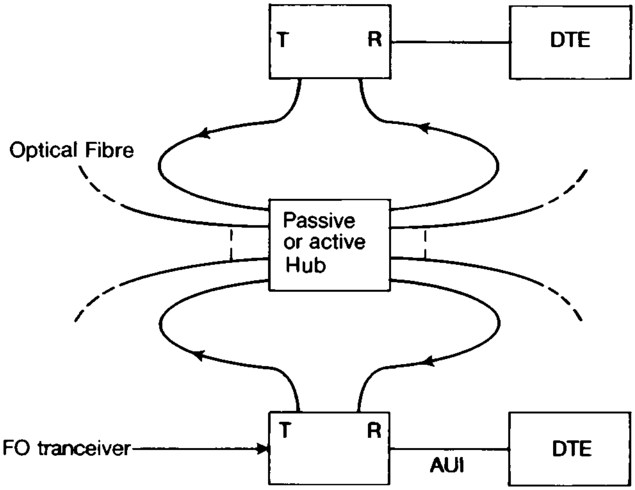


Fig. 9 10 Mbit/s CSMA/CD LAN via optical fibre

A 1 Mbit/s version of CSMA/CD LAN, known as 'Starlan', is operated via dual, unscreened twisted-pair telephone cable and the ISDN connector[17]. It is interesting that Starlan was created specifically to suit the existing installed base of telephone cables in North America. Radial distances of up to 500 metres are possible with this lower-speed version, and active hubs may be hierarchical, as shown in Fig. 10.

Token Bus: Token Bus LAN is currently specified to operate over a multidrop, coaxial cable bus with broadband signalling[18]. 10 Mbit/s Token Bus LAN has been adopted by the General Motors MAP initiative for application in a manufacturing environment. Within this initiative, there has been much discussion on the use of fibre optics for Token Bus LAN[19]. A passive optical star network is used to interconnect in the region of 16–32 Token Bus end systems via dual fibres, as shown in Fig. 11. Link lengths will be determined by the number of attachments and the optical loss of the cables used, however these will be reasonably long in practice (in excess of 500 metres radius).

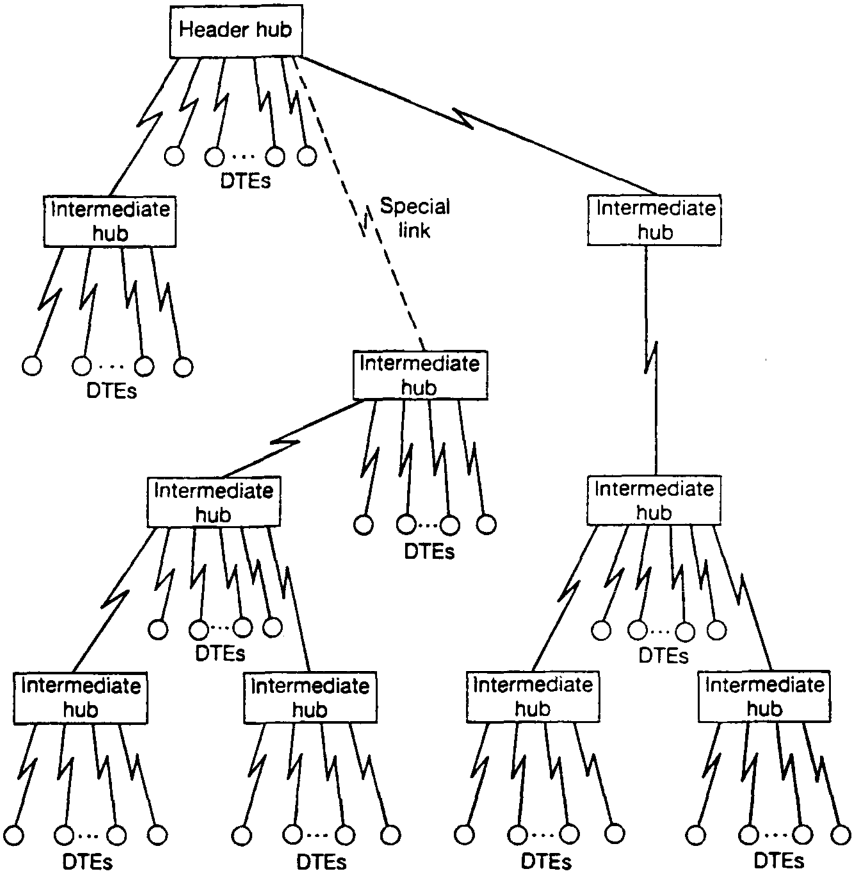


Fig. 10 1 Mbit/s CSMA/CD LAN (Starlan) via telephone cable

Token Ring: Token Ring LAN is currently specified to operate at a data rate of 4 Mbit/s over screened twisted-pair cable[20]. This logical ring is configured as a star in order to overcome the reliability limitations arising from the series of concatenated end systems connected to the sub-network[21]. The ring structure is physically distorted to create a series of 'lobes', as shown in Fig. 12. A Trunk Coupling Unit (TCU) will 'bypass' any radial connections that are not attached to a working station (end system); this is achieved in practice by the use of 'phantom' signalling across the attachment cable (a working station will impose a voltage between the Transmit + Receive wires to actuate a relay device located at the TCU). The radial cable length between station and TCU may be up to 300 metres.

One rather interesting (and unique) aspect of Token Ring LAN relates to the connector used for station attachment. This is a hermaphrodytic, 4 pole connector with appropriate screen contacts. The uniqueness arises from the

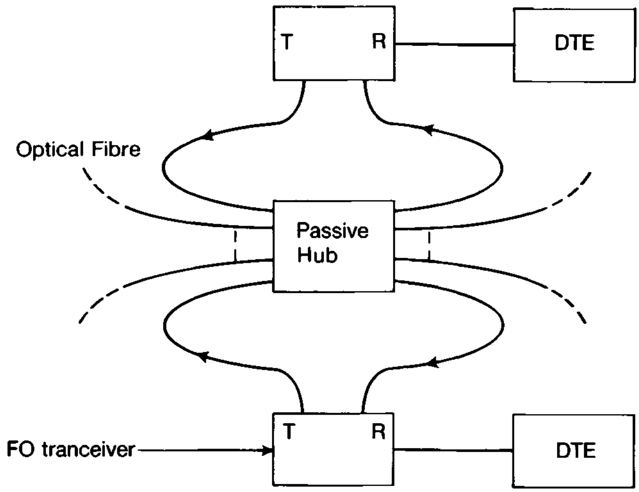


Fig. 11 10 Mbit/s Token Bus LAN via optical fibre

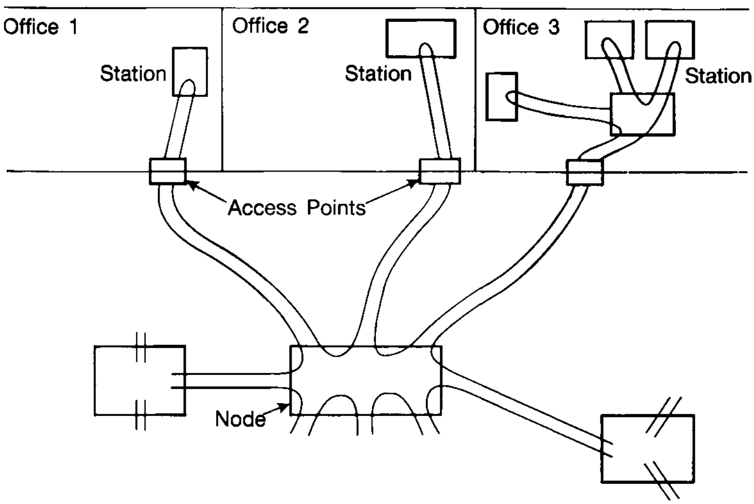


Fig. 12 4 Mbit/s Token Ring LAN via screened twisted-pair cable

'self-healing' switching function which is actuated whenever the connector in un-mated (inbound contacts are automatically switched through to outbound contacts). This functionality overcomes problems of resilience of a ring, and also adds to the size and cost of the connector. Perhaps a more significant consequence of this uniqueness is the lack of applicability to logical bus networks.

An extensive study has been made of the technical feasibility of operating 4 Mbit/s Token Ring LAN via installed telephone cables[22]. The results of this investigation are quite positive, and IBM has since announced its support of this approach. The tradeoffs associated with operating 4 Mbit/s over telephone cables are firstly, a reduction in sub-network connectivity from 260 stations to 72; and secondly, a maximum cable distance of 100 metres between station and TCU. Station attachment in this case will be via standard telephone jack, which implies that the network will be less resilient than the screened twisted-pair system which employs the special connector described earlier.

Work is now under way to extend Token Ring LAN to operate at 16 Mbit/s over screened twisted-pair medium. The previously specified cable and connector should cater for the higher bit-rate, however there may be a reduction in radial cable length due to increased attenuation at 16 Mbit/s.

10 Integrated Services Local Networks

A number of exciting developments are taking place under the generic title 'Integrated Services Local Networks' (ISLN). These developments generally fall into two categories: 'Local Access Networks' and 'Trunk Networks'. A brief review of the more significant projects is presented below.

The major benefits of functional integration of communication services are simplicity and economy. A common, coherent, fully-managed network capable of supporting concurrent data, voice, text and image services represents a more natural, efficient and effective mode of communication. A trend towards functional integration is also generally accepted to be taking place in the evolution of IT workstations. The physical integration of all modes of communication into a single cable is therefore clearly the path to follow.

Local Access Networks: The most notable Integrated Services Network is ISDN (Integrated Services Digital Network). Basic Rate ISDN is capable of providing each user with a 144 Kbit/s duplex channel, organised as 2B + D in each direction (where B is 64 Kbit/s and D is 16 Kbit/s). This circuit-switched facility contains two 64bit/s channels (2B), plus a signalling channel (D) for associated protocols. Multiple B channels may be combined by Time-Division-Multiplexing (TDM) to form Primary Rate ISDN, which in Europe is organised as 30B + D.

Several organisations are separately addressing the need for a higher performance hybrid circuit-switched/packet-switched Local Integrated Services Access Network. For example, IEEE Committee 802.9 is developing a standard for IVD (Integrated Voice + Data) Local Access[23]. This initiative is intended to support both ISDN and IEEE802 Packet Protocols, and is currently based on a 2B + "bigD" structure (where the "bigD" represents a packet channel of about 1 Mbit/s capacity). The IVD Local Access standard is being designed to operate at a data rate of approximately 2 Mbit/s over existing, unscreened twisted-pair telephone cable. Other organisations actively pursuing this development are ECMA, BSI and ESPRIT (Project 43).

Trunk Networks: A substantial amount of development has taken place over the last 4 years in the area of Integrated Services Trunk Networks, and many projects have adopted a common approach based on hybrid circuit-switching/packet-switching. Perhaps the most publicised project is FDDI-II (Fibre Distributed Data Interface), which is a 100 Mbit/s dual optical fibre ring incorporating a dynamic boundary between TDM-based circuit-switched channels and a wideband packet-switched channel[24]. FDDI-II is being developed as an ANSI Standard, which has a planned approval date of 1989. A further example of a High-Speed ISLN Trunk is the ESPRIT Project LION (Local Integrated Optical Network), which adopts a similar approach to FDDI-II but operates a 565 Mbit/s using laser transmission via single-mode optical fibre[25]. An interesting review of recent ISLN projects is referenced[26].

11 Standards initiatives

The development of Industry Standards relating to 'Universal Building Cabling' is under way within two US organisations, namely IEEE and EIA. The primary objectives of these initiatives are as follows:

(a) To define a common Fibre Optic (FO) infrastructure which is capable of supporting the various emerging standards for FO-LANs; specifically FDDI, FO-CSMA/CD, FO-Token Bus, and FO-Token Ring.
(b) To define a twisted-pair cable plant which will support existing applications, and future local interface standards such as IVD, ISDN, and Starlan.
(c) To get some order from the potential chaos arising from the many proprietary cabling products that now exist.
(d) To respond to market requirements for an open, flexible, and future-proof scheme for Building Communications Cabling.

Current status of each of the relevant standards activities is presented below.

EIA TR 41.8: This committee is developing a 'Communications Wiring Standard for Commercial Enterprises'[27]. The level of investment and participation in this project is substantial, and the plan is to issue an approved document late 1988.

The scope of the initial draft document (October, 1987) is 62·5/125 micron graded-index fibre, twisted-pair cable (currently all candidates), and even coaxial cable (75 and 93 ohm versions). The proposed cabling architecture is based on radial configuration within the workspace (referred to as 'horizontal cabling') and the use of Wiring Closets (Nodes) to flexibly connect Local Cabling Zones onto a 'Backbone Network' (Trunk). The draft standard arguably contains too many options to provide users with a sound strategic basis for the future, however it should be interesting to see if the number of options reduces as the standard matures.

IEEE 802.8: As the Fibre Optic Technical Advisory Group (FOTAG) within LAN Standards Committee 802, this committee has been addressing the practical application of Fibre Optics for some time. The 802 FOTAG has 3 main activities:

(a) *Recommended Practices:* This document will recommend the physical and mechanical practices for FO Cable Plant. Guidance is also provided on the design, test, and maintenance of FO systems. An initial draft was issued November, 1987.
(b) *Universal Cable Plant:* An independent specification was being developed, however EIA TR 41.8 is now acknowledged as forerunner. This group will continue work in this area, and will liaise closely with EIA to support its development.
(c) *Technical Advisory on FO-LAN Standards:* General guidance is given to 802 Working Groups on FO-LAN Standards in an attempt to make them complete and consistent. Examples of the kind of advice given are: the measurement of optical bandwidth and waveshape, measurement of signal jitter, and multi-fibre operation. Much of the technical support comes from the ANSI FDDI project and other national/international FO Committees.

12 Open Systems cabling

Open Systems Interconnection must provide the user community with a broad level of independence in the procurement of IT equipment. Whilst OSI LAN Standards have aimed to fulfil this objective, the actual variations in data rates, media type and cabling topographies clearly do not help.

The cabling architecture presented in this paper represents a possible solution to the above problem. Based on twisted-pair and optical fibre media, the 'tree and branch' approach could support today's industry standard LANs, point–to–point communication links such as V24, and circuit-switched telephony within a neat, integrated cabling system. Furthermore, the proposed combination of architecture and media provide an appropriate infrastructure for tomorrow's functionally integrated services networks. This natural evolutionary path would protect initial investment, and establish the cabling system as a long-term building utility.

The proposed cabling architecture and choice of media is presently being processed by the US Standards community as the basis for a 'Building Communications Cabling System'. The applicability of this emerging standard must, in principle, be international.

In conclusion, it seems that 'Open System Cabling' is indeed a technically feasible and practical possibility.

Acknowledgement

I would like to express my thanks to Ian Barker of ICL Mainframe Systems for his helpful comments on the technology aspects of this paper. I am grateful also to Adrien Bolton, Alec Domm, Brian Rossiter and Edwin Turner, of ICL, and Keith Cowes of STL for reviewing this paper prior to publication.

References

1 CAMRASS, R. and SMITH, K.: Wiring up the Workplace – A Practical Guide for Management. ISBN 0-907822-66-5. IBC Technical Services (1986).
2 FORRESTER: The Network Strategy Report, Volume 1, Number 12; Local Wiring Futures, November 1987 (pp. 2–11).
3 SOUTHWOOD, W.: Designing Buildings with Wiring in Mind; Practical Wiring Issues Conference, 13th October 1987. IBC Technical Services.
4 MOBBS, R.: Data Cabling Infrastructure for Lloyd's of London. Practical Wiring Issues Conference, 13th October 1987. IBC Technical Services.
5 Gartner Group Network Market Forecast (1987).
6 CAMPBELL, R. and CONTE, R.A.: AT&T 25-Pair Cable Transmission Characteristics. Working Paper submitted to IEEE 802.3 (November 1987).
7 CONTE, R.A.: AT&T 25-pair Cable NEXT Contributions at 10 MHz. Working Paper submitted to IEEE 802.3 (November 1987).
8 ISO IS 8802-3; CSMA/CD Access Method and Physical Layer Specifications.
9 GIBSON, R.W.: Comparing Features Aids Selecting Broadband Local Nets. Data Communications, April 1982 (pp. 127–135).
10 STRAUSS, P.: Lightwave Future gets even brighter. Data Communications, February 1987 (pp. 53–56).
11 STEVENS, R.W.: Macrolan – A High Performance Network. FOC/LAN'83 (October 10–14, 1983). Information Gatekeepers Inc., Boston.
12 STANLEY, I.W.: A Tutorial Review of Technologies for Coherent Optical Fibre Transmission Systems. IEEE Communications Magazine, August 1985 – Vol. 25, No. 8 (pp. 37–53).
13 KELLY, J.C.: Can One Cable Type Handle all Network Needs? Data Communications, September 1987 (pp. 171–181).
14 RAWSON, E.G. et al.: Fibrenet II; An Active Star-Configured Fibre Optic Local Computer Network with Collision Sensing. OFC'82, USA (April 13–15, 1982).
15 ROWORTH, D.A.A. and HOWE, N.: Isolan – A Fibre Optic Network Conforming to IEEE 802.3 Standards. Fibre Optics'86, London. SPIE (1986).
16 OZAWA, K. et al.: Fibre Optic CSMA/CD LAN. Submission to IEEE 802.8 by NTT, Japan (July, 1987).
17 IEEE 802.3e; Physical Signalling, Medium Attachment, and Baseband Medium Specifications, Type 1Base5.
18 ISO IS 8802-4; Token Bus Access Method and Physical Layer Specifications.
19 KIELI, M. et al.: Fibre Optic LAN for MAP Applications. EFOC/LAN'87 (June 3–5, 1987). Information Gatekeepers Inc., Boston.
20 ISO IS 8802-5; Token Ring Access Method and Physical Layer Specifications.
21 LOVE, R.D. and TOHER, T.: How to Design and Build a Token Ring LAN. Data Communications, May 1987 (pp. 213–226).
22 HOBGOOD, B.: Field Evaluation of the IBM Token-Ring Network using Installed Telephone Pairs. IBM Technical Report TR 29.06434 (September, 1986).
23 BEDARD, C. et al.: A Proposal for an IEEE 802 Integrated Voice/Data LAN Standard. Submission to IEEE 802 Executive Committee (1986).
24 CAVES, K.: FDDI-II; A New Standard for Integrated Services High Speed LANs. Networks'87, London (1987).
25 REY, J.C. et al.: An ESPRIT Project; A Local Integrated Optical Network. FOC/LAN'85 (September 18–20, 1985). Information Gatekeepers Inc., Boston.
26 STRAUSS, P. and BORRUS, A.: New Fruit of Hybrid Circuit/Packet Switches Promises Network Efficiencies. Data Communications, October 1986 (pp. 91–96).

Collecting and generalising knowledge descriptions from task analysis data

Peter Johnson, Hilary Johnson
Department of Computer Science, Queen Mary College, University of London
Fred Russell
ICL Human Factors Laboratory, Bracknell, Berkshire

Abstract

The paper is concerned with work that forms part of a project funded by the ICL University Research Council, Task Analysis as a Design Tool. It outlines a methodology for carrying out task analyses, and extends our approach to task analysis, known as Knowledge Analysis of Tasks (KAT). The methodology concentrates on gathering information about the knowledge people possess about the tasks they are required to carry out. We propose methods for eliciting and analysing various aspects of task knowledge from planning and task decomposition knowledge, to knowledge about objects and their associated actions which comprise tasks. Additionally, we argue that task objects are highly structured in the sense that combinations of objects and actions that make up tasks are not selected arbitrarily rather that only certain objects can be combined in tasks, and only in certain ways. A further theoretical position relates to the representation of task knowledge by the task performer; our view is that tasks can be treated as conceptual structures stored in memory. Finally, we are concerned that not all objects and actions, that make up the procedures to be carried out in tasks, are equally representative or important in task execution. Object representativeness and importance have implications for interface design. We believe that the more representative task objects must also be those represented at the user interface: The consequences of not doing this can lead to an increase in user errors and other decreases in usability.

Task analysis is the investigation of what people do when they carry out tasks. However, task analysis involves more than collecting information about how people perform tasks. An approach to task analysis involves a number of aspects; how data are collected, a description of that data, a method of analysing tasks and a representational framework for modelling those tasks. In this paper we discuss data collection, a method of analysing and generalising from those data, and a framework for task modelling. The data collection, generification method, and task modelling put forward are part of our approach to task analysis, known as Knowledge Analysis of Tasks (KAT). This approach has been developed from earlier work on Task

Analysis for Knowledge Descriptions (TAKD), Johnson, Diaper and Long (1984). KAT has been fully described in Johnson and Johnson (1987a, 1987b, 1987c) and is concerned with the knowledge people possess about a task. It is to be contrasted with other task analysis techniques which are not concerned with knowledge, such as ability profiling (Fleishman and Quaintance, 1984) and Hierarchical Task Analysis (Annett and Duncan, 1967), and other techniques which have an evaluative role in assessing the complexity of task performance but have no explicit method of task analysis. The work of Kieras and Poulson (1985), Payne and Green (1986) and Card, Moran and Newell (1983), are good examples of current approaches to predicting the difficulty of using an interactive computer system. Each of these approaches has (in varying degrees) suggested how proposed designs can be evaluated in terms of the ease with which users could perform given tasks. There are two important features to these approaches; first, they are designed purely as evaluative tools and assume that decisions about what tasks the system should support have been made elsewhere, and that one or more design solutions have been proposed. Second, they focus on the evaluation and prediction of user performance and do not detail any method of task analysis. In contrast, TAKD and more so, KAT are the only approaches to task analysis that identify the knowledge requirements of tasks and are aimed at assisting in the generation of design solutions. KAT can also form an important part of an evaluation methodology. This paper describes the method of KAT and the underlying rationale, and makes recommendations as to how KAT can be used in the design process.

Theoretical basis for KAT

Before discussing the identification of knowledge in tasks, we discuss the theoretical notions of: representing tasks as concepts, task structure, and object representativeness.

(i) Tasks as concepts. We believe that tasks are represented as concepts or general knowledge structures in long term memory. All of the knowledge a person possesses about a task is contained within the general knowledge structure and this knowledge is used in task execution. More details about concepts and general knowledge structures are given later in the section on task modelling.

(ii) Structure in tasks. Tasks would be unstructured if the task world formed a set of tasks in which all possible elements and attributes of tasks could co-occur with equal probability combined with all other possible elements or attributes of tasks. This is obviously not the case; task elements or behaviours do not occur independently of one another. Some pairs or even *n*tuples of task elements are quite probable whereas others are highly improbable; some groups of attributes while being logically possible may never occur in reality. Some elements of tasks are naturally carried out together, precede, follow on from, or prime one another. Elements of tasks are generally carried out according to some feasible temporal ordering, designated by a plan.

For example, a builder who is building a house cannot begin to build until the bricks have arrived. An architect designing the layout of a house cannot design the upstairs layout until he knows how many bedrooms are required. The same architect might have to treat certain related task elements at the same time. For instance, in designing bathroom layout, the respective position of the bath is considered at the same time as the positions of the wash-hand basin and toilet. A similar example occurs when overall layout is concerned, and bathroom and kitchen positions have to be considered simultaneously because of constraints imposed by plumbing requirements.

(iii) Object representativeness. Objects and their associated actions differ in how representative of the task they are. For instance, in the same way that one might argue that a 'robin' is a typical instance of the essence of the 'bird' category so might the procedure 'drawing house sketches' be more typical of an architect task than 'answering the telephone', although both procedures may go some way to achieving the goal of designing a house, and therefore both procedures are parts of the task. The above example can also be used to illustrate the difference between typicality or representativeness, and importance of an object within a task. Answering the telephone and finding that finances are available to build the house might be more important in attaining the overall goal of building a house than producing a sketch of the house layout. However, producing the sketch may be a more typical or representative aspect of the task. Importance is not a static quality however, and therefore will be determined by the particular circumstances and constraints imposed on a given task in a particular context. In this paper we do not distinguish explicitly between importance and representativeness since the consequences of overlooking both representative and important task elements in interface design are the same, namely, an increase in errors and a decrease in usability.

Identifying Knowledge

Different tasks require different knowledge, and there are different knowledge aspects within the same task. Therefore, we assume that there are subsets of knowledge which make up people's total task knowledge. First, the analyst needs to identify the person's goals, subgoals and subtasks. In other words how the person divides up the task into its constituent parts and how these are achieved for the particular task in question. Second, related to the first criterion, it is important to consider the plans and structure (e.g. sequencing) of the subtasks. Third, it is necessary to identify the procedures and operations of a plan, and fourth, the objects involved in the task and the actions which are associated with them, these are the action/object pairings. Finally, we believe that the objects themselves might be structured in some way and that this structuring is itself an important aspect of knowledge. In summary, the knowledge that is required by a task involves knowledge of the task decomposition into subgoals, etc., the plans and their structuring; the procedures within a plan and the actions and objects within a procedure; and how those objects are structured.

A method of carrying out task analyses

1 Applying knowledge gathering techniques to task analysis

This section is divided into three parts: the first part is concerned with general guidelines for task analysis. The second part outlines guidelines for using the various techniques and finally, the last part of the section is concerned with guidelines for identifying knowledge components in KAT.

1.1 General guidelines for task analysis

Johnson and Johnson (1987a) are concerned with *how* to identify knowledge for a task analysis. Here we provide some general guidelines for carrying out knowledge identification in task analysis.

Important guidelines to remember are:

(i) identify the purpose of the analysis;
(ii) check the analysis with the task performer(s);
(iii) analyse more than one person and one task, in alternative settings;
(iv) do not rely on only one technique of gathering knowledge.

1.2 Guidelines for when to use the various knowledge gathering techniques

1.2.1 Structured interviews and questionnaires. Interviews and questionnaires are suitable for extracting rules and background information, covering low probability events and identifying general principles and reasons underlying behaviour. Interviews are faster than other techniques to carry out but they do not provide in-depth knowledge, and should be supplemented with direct observation of the task performance of a number of individuals.

1.2.2 Observational techniques. These are particularly appropriate for providing corroborating evidence, gathering in-depth knowledge, for when knowledge is context-bound, and when the task involves many individual steps. However, the technique is time consuming, cannot be used in isolation and requires inference on the part of the analyst to identify the structure of the task and certain types of objects and actions.

1.2.3 Concurrent and retrospective protocols. Protocols provide detailed information on all aspects of a task, e.g. planning, how the task is decomposed, and of the actions and objects. However, protocols require some inference on the part of the analyst, the responses must be carefully coded within the context of some broader framework, and the enterprise is time consuming. In concurrent protocols (CP) subjects report what they are doing while they are doing it. CP are appropriate when there is insufficient time to carry out retrospective protocols (RP), when the analyst is interested in what a subject is doing at that time, and when physical plans and other physical aspects of tasks are of interest. In retrospective protocols (RP) the subject is required to generate a durable memory trace while completing the task, and then the contents of the trace are verbally reported. A retrospective protocol could be given whilst the task performer observes his/her own task performance, for example, using a video. RP reports are appropriate when the analyst requires more information than is available through CP. Additionally, RPs are appropriate when the analyst is concerned with the reasons for and explanations of actions, with cognitive aspects of tasks, and with the feelings and emotions the person entertains about the task.

1.2.4 Experimental techniques. In this section we consider several techniques which may be employed in identifying the similarity of concepts, sharing of attributes and establishing the structure of concepts. All the techniques described in this section require the analyst to have already obtained background information, conducted interviews or analysed protocols.

1.2.4.1 Kelly's repertory grid (Kelly, 1955). To carry out this technique the task analyst must take note of all objects or entities manipulated, or mentioned by the person performing a task. These objects or entities form the concepts to be represented. The technique involves first, selecting a set of objects and then presenting these to the subjects in groups of three. The subject is then asked in what way(s) two of them are alike and different from the third. This grouping and questioning process is repeated until all the objects have been presented to the subject. The result is a structuring of similar objects which share common attributes. One problem with this technique is that the analyst has to be very careful in choosing which three concepts are put together, since the contrasting set is all important. There is also a possibility of forcing a classification outcome which is arbitrary, an artifact of the selection procedure, and which is not at all representative of the task.

1.2.4.2 Card sorting (Rosch, 1978). In this technique the analyst is concerned with the similarity of concepts involved in the task. The concepts here are construed as the class of objects which with their associated actions, constitute the action/object pairings. Higher level concepts might represent the tasks themselves. The procedure of this technique is somewhat similar to Kelly's repertory grid. Object names are entered on cards, one card for each object, and the subject is instructed to group 'similar objects', or 'objects which are the same kind of thing'. In the psychological literature, the instructions are generally of the following type: 'put together the things that go together'. The result of this technique, as with Kelly's repertory grid, is a structuring of similar objects which share common attributes.

1.2.4.3 Rating scales. Another technique which could be useful in identifying object structure are rating scales. The name of each object is entered on a separate card and subjects are instructed to judge the given object for representativeness or importance in the task on an appropriate scale, with the highest number of the scale representing greater importance, and vice versa. An alternative to this procedure is to instruct the subjects to sort the cards into an order of relative importance or representativeness of the task.

1.2.4.4 Frequency counts. With frequency counts the analyst must note on how many occasions an object is either manipulated or referred to. The assumption is that an object which is more important will have a higher frequency score than an object of lesser importance. Frequency counts are important because they provide one way to identify which objects should be represented at the interface, and the relative detrimental effects of leaving out an object of a given degree of importance (i.e. frequency). Frequency counts also provide an index which can be used to compare individual differences across different people performing the same task. Such a comparison might provide some indication of differences in task organisation and planning across individuals. One problem with this technique is that it is likely to be very time consuming and exacting for the analyst and will almost certainly involve recording the prior interview or observation. Furthermore, frequency is only one criterion of importance, and some objects may be infrequently used or mentioned but crucial to task performance in certain contexts. The technique, like the others in this section, because of its very nature cannot be used in isolation.

1.2.5 Other useful techniques. Other techniques which might be used in addition to, or incorporated into, the above techniques are:

(i) knowledge competitions;
(ii) group discussions;
(iii) multi-choice questions;
(iv) analyst to carry out task with instruction;
(v) observation with a knowledgeable person providing commentary;
(vi) asking for sample outputs.

1.3 Guidelines for identifying knowledge components in KAT

The knowledge required by KAT is concerned with actions and objects, and the structure of those objects; knowledge about structuring and planning; and knowledge about how a task is decomposed into goals and subgoals.

1.3.1 Identifying objects and actions. Identification of objects and their associated actions used in carrying out the task can be obtained from:

(i) selecting objects and the actions associated with them from textbooks, a tutorial session, pilot study or with the analyst herself carrying out the task;
(ii) questioning the task performer in a structured interview about the

actions and objects, and then listing all the relevant nouns and verbs produced by the person in answering the questions;

(iii) asking the task performer for a list of either just the representative, or possibly all the objects they can think of, which are involved in the task, and the actions carried out on them;

(iv) observing the person carrying out the task, carefully noting what objects they manipulate and in what ways;

(v) noting all the objects and actions mentioned by the person in either concurrent or retrospective protocols.

1.3.2 Identifying object structure. Objects can be structured in terms of their representativeness and importance in the task. Identifying object structure can be achieved using one or more of the following methods:

(i) Count the frequency of how many times an object is referred to, in either interviews or protocols. The assumption here is that the more representative or important objects will be the most referred to.

(ii) The person who carries out the task is required to check the validity of a list of objects, produced by the task analyst, involved in a task and systematically leave one or more objects out and check the person's reaction to the list; it is assumed that the more important omissions will be commented on. This is fairly laborious, and a different list order must be presented to each subject to reduce any ordering bias there may be. It is generally not to be recommended since it is a weak methodology, relies on trial and error, and is only feasible given no strict time restraints.

(iii) The analyst may use rating scales where the name of each object is entered on a separate card and the person asked to judge the importance of a given object on a scale of, for example, 1 to 5, with 5 being the most important.

(iv) Using object names on cards as in (iii) the subject is required to sort the cards into an order of increasing importance in the task. Again, this technique is an alternative way of identifying the relative importance of objects.

(v) The analyst instructs the person to list all the objects in tasks, and the actions associated with them. The order in which they are listed is assumed to reflect the order of representativeness of the objects within tasks. The resulting lists (one from each person) can then be compared with one another to check the possibility of consensus of object representativeness.

It is important to identify the importance and structure of objects in tasks since it might not be feasible to represent all objects at the user interface. However, the user interface should at least represent objects which users deem to be important and representative across tasks. The analyst would therefore be able to identify which objects are likely to detract from task performance if not represented to the user.

1.3.3 Techniques used to identify planning knowledge. This section summarises techniques for identifying the knowledge a person has in terms of the task-plan, the structure or sequence of carrying out routine procedures, and knowledge about strategies used in the task, such as shortcuts and rules of thumb.

(i) Specific questions in the structured interview. This involves asking a person how s/he plans the task, if s/he uses the same plan for any other task, and identifying any modifications required to the plan. It is useful to ask specific questions of the sort 'what do you do if ...', for example, 'X goes wrong or fails'. The analyst should also ask whether any particular strategies or procedures exist for carrying out some part of the task, and if so how they are used, and why they are there; whether actually carrying out the task differs in some way from the ideal, or textbook approach, for example by taking shortcuts, and why. Ask what signals the end of one part of the task, and what triggers the start of another procedure;

(ii) via protocols, and observation. This involves initially having some knowledge of the task so that the ending of one phase and the starting of another can be easily identified.

1.3.4 Knowledge about how a task is decomposed. This can be obtained by:

(i) Asking specific questions about what the goal, subgoals and subtasks of carrying out the task are.

(ii) From any available checklist which comprises ideal components of the task, or a textbook, or instruction manual which decomposes the task into its constituent parts.

(iii) Asking or aiding the person to construct a hierarchical tree or flow diagram of connected parts of the task, making a specific requirement for different parts of the task to be labelled.

(iv) Identifying different phases of the task either from observations, concurrent protocols or retrospective protocols. When using observations a phase of the task may be identified by pauses. In concurrent or retrospective protocols, it is important to make a note of such statements as, 'now, I intend/want to ...', etc. A note of caution here about anaphoric reference. The analyst should be sure which referents belong to 'this', 'that', 'it', etc. This task decomposition should then be verified in some way, for instance, by checking against the decomposition provided by the person.

Generification – A process of identifying generic properties of tasks within a given domain

2 A method of generification

A method of generification must be capable of satisfying the following requirements:

1. Identifying generic actions and generic objects.
2. Identifying generic plans, generic procedures and task sequencing.
3. Provide a method of verifying the generic properties identified in **1** and **2**.
4. Generalise across users, tasks contexts and organisations.
5. Provide input to a task modelling activity appropriate for use in system design.

The method put forward in this section meets each of these requirements. Throughout, the method of generification is exemplified by an analysis of an architectural task.

2.1 Generic actions and objects

The following steps are recommended for achieving identification of generic actions and objects:

1. The first step in identifying generic actions and objects is for the analyst to construct two separate lists, one for the actions and one for the objects that have been manipulated, mentioned or referred to in some way by the task performer(s). These lists will contain disparate, and often repetitive information from each task performer over one or a range of tasks.

From the task analysis carried out by the authors of the architect task of designing the room layout of the house, we produced two lists containing all the actions and objects from each architect. Several actions and objects were manipulated or referred to on several occasions, therefore the list was repetitive, examples of the objects were: *plans, symbols, windows, pipes, appliances, doors, pens, rulers,* etc. Examples of actions were *draw, rehang, check, reposition,* etc.

2. The second step (as in Johnson, 1985) is to reduce the above list to a 'comprehensive list' with each action and object appearing once only. In the architect example, the two lists of actions and objects were reduced so that each action and object appeared only once. The original lists were kept however as these provided some measure of the respective importance of the objects and actions in the task.

3. The third step is to identify generic actions and objects; this can be achieved in a number of ways.

(a) By assuming a critical value or threshold – the analyst must now decide at what level the threshold is to be set in order to judge if something is or is not generic. The analyst must be cautious in setting this level as the ways in which objects and actions are identified may be biased in favour of only identifying generic actions and objects. At the outset then, we would argue that the best policy is to treat an item as generic if it is referred to by two or more task performers. If this yields an unmanageable list of generic actions and objects then the threshold may be made more stringent. The threshold relies to some extent on the analyst's intuition, and underlies the fact that there is no absolutely correct specification; or

(b) By grouping like terms – as in Johnson (1985) the analyst must associate all like terms. The assumption here is that the 'comprehensive list' contains all actions and objects involved in the task and that these are then grouped. Associating all like terms can be achieved by:

  (i) the analyst(s) relying on his/her/their intuition and using an iterative procedure where a particular term is associated with any other similar term. Similarity is determined by attempting to re-express the original task description in terms of the alternative action or object. If the alternative term was 'adequate' then the two were said to be similar; or
  (ii) grouping by independent judges. The analyst asks one or more judges to sort cards into groups with the instruction to 'put together the things that go together' or 'put together the things which are the same kind of things'. The results of each judge's sorting can then be correlated to identify the agreed, generic task elements;
  (iii) asking 1 or more *task performers* to group the objects and actions, in the same way as above;
  (iv) after the groups have been chosen, the next step is to identify a generic label or term which might cover all the individual elements in a particular set. These terms then represent the generic task elements.

In the example from the architectural task, we used the threshold level [(a) above] to identify generic actions and objects. This procedure was used since there was a time constraint and generally it is quicker to use a critical value than to group like terms. However, it is not as appropriate a procedure as grouping like terms because there is no opportunity for the task performer to judge whether the generic actions and objects identified are actually common properties; this can be improved by involving the task performers in the verification process. By this method we identified many generic actions and objects, examples were 'plans' and 'draw'.

4. The fourth step is the validation of the generic elements. This section only applies to the grouping of like terms, as the definition of what is generic is decided by the analyst where a critical value or threshold is set.

To validate the generic elements, list all the actions and objects with the generic titles above. The *task performers* are then instructed to place an identifying letter from the generic group titles, at the side of each action and object. If the action or object is not adequately covered by a generic title then the task performer is free to supply an alternative group title.

Finally, in considering object structure, the generic elements which are referred to by the most task performers are the more representative and typical generic objects of the task.

2.2 Generic plans and task decomposition

Plans and task decomposition (goals, subgoals, subtasks) are considered together. They differ from generic actions and generic objects in terms of number available. For example, there may be a large number of objects and actions which have to be manipulated in performing a task. However, we expect there to be a smaller number of different plans involved in carrying out a task and do not expect there to be as many different task decompositions. Plans have generic features which are always present in carrying out a particular task, but there will also be specific features which make the plan flexible and which depend on differing circumstances and contexts. Therefore the procedure for identifying generic plans and task decomposition is different to the identification of generic actions and objects. Identifying generic plans and task decompositions involves listing all the answers to the questions and all the other means of identifying plans and task decomposition outlined above in sections 1.3.3 and 1.3.4. The generic plans, goals and subgoals are those which are mentioned by two or more task performers.

To identify generic plans, procedures and task decomposition:

1. List all the components and sequence-related details of plans, procedures and task decompositions which result from carrying out the identification procedure in section 1.3.3 and 1.3.4 above.

2. Verify these details with a number of task performers by asking if the procedures are appropriate and if they are in the correct sequential order, or by having activities written individually on cards that task performers must sort into an appropriate order for carrying out the task.

3. Include in the generic description all generic details provided by two or more task performers.

4. Verify this generic description with task performers by asking if this is how the task is usually carried out and by noting under what circumstances exceptions are appropriate.

If there is at any time any conflict in terms of what elements are manipulated at what stage, i.e. if the plans, procedures and task decompositions differ in any way, then the one most frequently referred to should be the one adopted as generic. In terms of user interface design there must be a facility at the interface which allows for flexible organisation of carrying out the task. This depends to a large extent on the type of task, the circumstances, and the task performer. Therefore the analyst must be able to identify different strategies and the circumstances under which they are employed.

In our example task of designing the layout of a house the architect can take one of at least two starting points, both of which were mentioned during the task analyses. The architect may either begin with the overall shell drawn and s/he then proceeds by mapping out space within this shell, or alternatively s/he can begin by drawing one room first and then adding other rooms until the overall plan (as in sketch or diagram) is drawn. We found that the first approach was the most prevalent, and therefore assumed this as the generic plan to be presented at the user interface. However, the latter approach also had to be accommodated since this was used by other architects, and therefore it would be recommended to the software designers that the architects should be given an optional starting point.

2.3 Generification methodology summary

2.3.1 Generic actions and objects

1. Construct two separate lists, one for the actions and one for the objects from all the tasks and task performers.

2. Reduce the above list to a 'comprehensive list' with each action and object appearing once only.

3. Identify generic actions and objects by:

(a) adopting a threshold of treating an item as generic if it is referred to by two or more task performers. If this yields an unmanageable list of generic actions and objects then make the threshold more stringent; or

(b) associating all like terms which can be achieved by:
- (i) relying on one's intuition, and by using an iterative procedure to judge entity similarity, or
- (ii) grouping by independent judges, or
- (iii) grouping by using task performers as judges;
- (iv) identify a generic label or term for all the individual elements in a particular set.

4. Validation of the generic elements by task performers.

2.3.2 Generic plans and task decomposition

1. List all the components and sequence-related details of plans, procedures and task decompositions which result from carrying out the identification procedure in section 1.3 and 1.4 above.

2. Verify these details with a number of task performers.

3. Include in the generic description all generic details provided by two or more task performers.

4. Verify this generic description with task performers.

5. If there is at any time any conflict in terms of what elements are manipulated at what stage, then the one most frequently referred to will be the one adopted as generic and this will be represented as the default.

Task modelling – Identifying the structure and rules for modelling tasks

3 Constructing task models

We demonstrate the representational framework here by constructing a task model of the architectural task. Various stages have to be undertaken in constructing task models.

Construction of a task knowledge structure (TKS).

Construction of a goal-oriented substructure.

Construction of a taxonomic substructure from the generic task actions and objects.

Embedded task rules and the construction of production rules.

3.1 Construction of a task knowledge structure (TKS)

The first stage in our task modelling scenario is to construct a task knowledge structure. This is achieved by first, identifying the task to be analysed and then collecting the appropriate data using the procedures outlined earlier in section 1. These data are then subjected to generification processes by which common task elements are identified. These common task elements make up a subpart of the TKS.

The task to be analysed is assumed to have a corresponding conceptual structure or TKS associated with it, and therefore the task becomes the label attached to the TKS. In the architectural task, this is 'design the room layout of a house'. All the knowledge collected by the task analyst must be represented in the TKS either (i) in the goal-oriented substructure which contains the goal structure, plans, strategies and procedures which define the sequence of carrying out the task; or (ii) in the taxonomic substructure, which is closely related to the goal-oriented substructure through sequences of procedures, and contains the categories of generic actions and objects that are required in the task and their degree of representativeness. Additional specific knowledge is also linked to the TKS and there are also links to similar tasks. It is an empirical question whether all the knowledge identified can be captured by the substructures outlined above, but TKS's do have the capacity to represent all aspects of knowledge. A TKS for the architectural task is given as an example in Fig. 1.

A TKS is a collective term for all the knowledge a person possesses about a task and gives the task analyst the opportunity to label the knowledge s/he has identified. One significant advantage of modelling knowledge in this way is that links to similar tasks show where commonalities may lie, such as

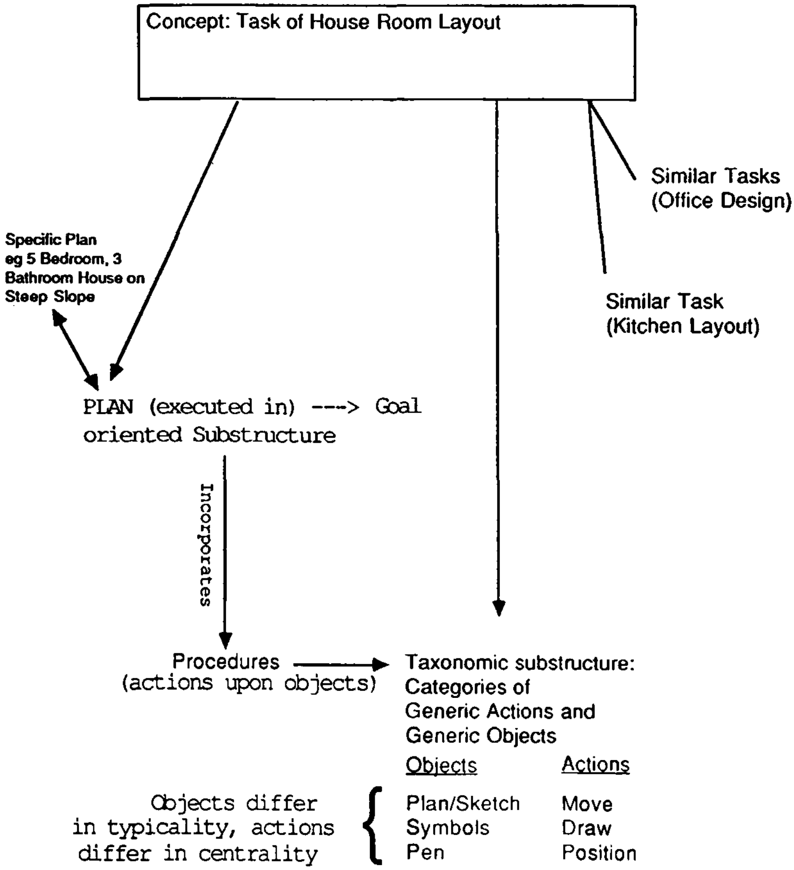


Fig. 1 Task knowledge structure

common objects, actions and plans. Thus, a computer system or its user interface might support several types of task and the model identifies what the common properties across a variety of tasks are.

Within a TKS there are goal-oriented and taxonomic substructures and these substructures are a part of a task model. The next two sections are concerned with the construction of a goal-oriented substructure and a taxonomic substructure, part of which is called up by the goal-oriented substructure

3.2 Construction of a goal-oriented substructure

Planning activity involves satisfying a set of goals and subgoals in a prespecified sequence of procedures of actions upon objects. Therefore, plans are inherent in goal-oriented substructures. Goal-oriented substructures contain many goals. Structured goal nodes direct sequences of events, which unfold over time and eventually satisfy the subgoal nodes. Goal nodes can vary in hierarchical level, for instance the sub-subgoal 'find a small piece of paper' is subordinate to the more superordinate subgoal, 'draw a rough sketch'. An assumption we are making here is that goals, subgoals and subtasks can be represented by nodes with links between them. Thus nodes can be treated as either conditions or states in production rules.

The goal-oriented substructure 'calls up' appropriate knowledge from the taxonomic substructure by the use of procedures. Associated with subtasks are sets of procedures which have to be executed in order to achieve subgoals. Procedures are sets of actions carried out upon objects, and call up the task rules outlined in the taxonomic substructure. The task rules determine the sequence in which actions are carried out on which objects, and which other objects are associated with a given action. Therefore, an acceptable procedure is contingent on the task rules represented in the taxonomic substructure. A set of procedures are carried out to achieve a subgoal. A single procedure is modelled by a production system containing production rules.

Not only may the task be decomposed in different ways, there may also be a choice between a number of competing sets of procedures, one set of which may be more appropriate than the others. This appropriateness will be affected by contextual information and the circumstances under which the task is to be executed. We call the selected set of procedures a strategy. Additionally, single procedures in a given strategy will differ in how important or central they are to the task as a whole. Some procedures will be so central to the task that a failure to execute will result in the task being unsuccessful. Figure 2 is a subpart of a goal-oriented substructure for the architectural task.

Nodes and links between nodes in the network are identified by observing and questioning task performers as to what they are doing now and what they will do next, see Johnson and Johnson (1987a).

3.3 Construction of a taxonomic substructure

The taxonomic substructure contains knowledge about generic actions and objects and the relationships between them. The taxonomic substructure has three levels.

The top level of the taxonomic substructure is the superordinate task category, 'design the room layout of a house'. The second or basic level of the taxonomic substructure contains the objects and their associated actions involved in designing the room layout of the house which comprise a 'basic level' category (see Rosch et al, 1976), for example 'plan' and 'draw'. The basic level category contains a large amount of knowledge about the objects that make up the superordinate task category. The basic level task category

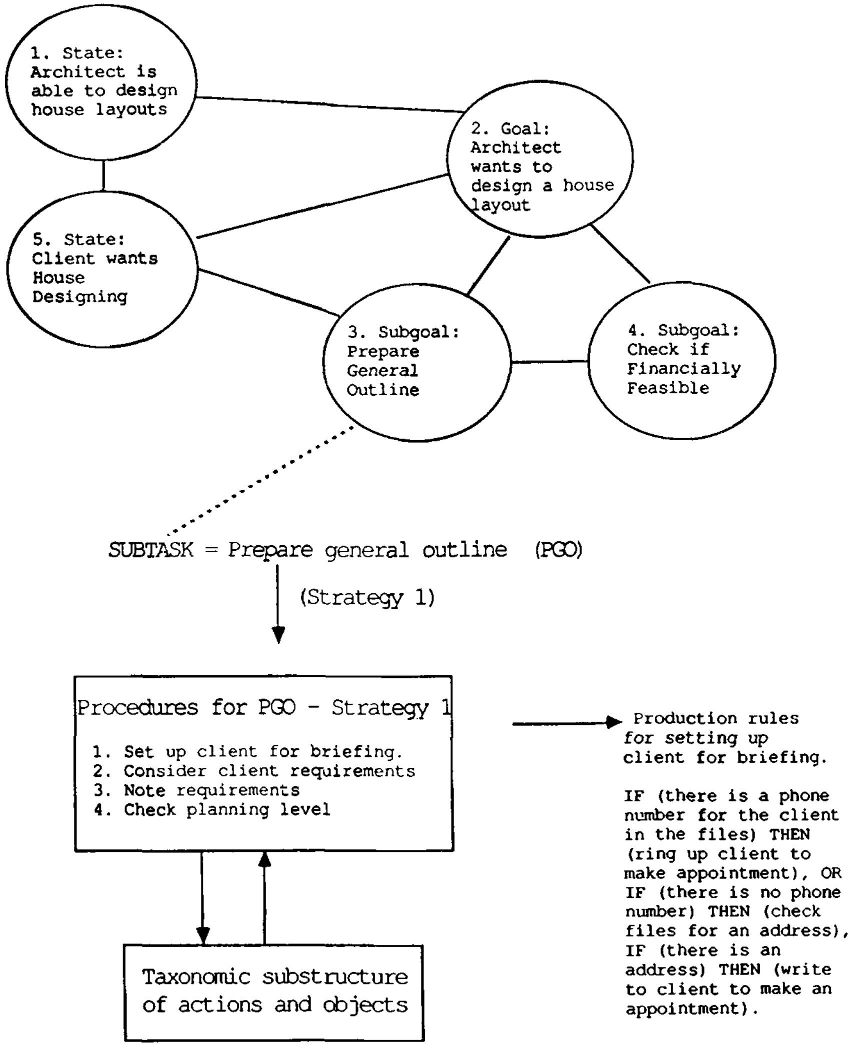


Fig. 2 Subpart of a goal-oriented structure of the architectural task

represents knowledge about (i) in which task procedures a category member is used; (ii) which other task objects a category member is related to, and what that relationship is, i.e. whether the category member primes, precedes, follows or is carried out in conjunction with other task objects; (iii) which actions are associated with a category member; (iv) what features or properties a category member possesses; (v) the usual circumstances under which a particular category member occurs, for example, whereabouts in the task the category member is manipulated, and finally; (vi) a set of category members.

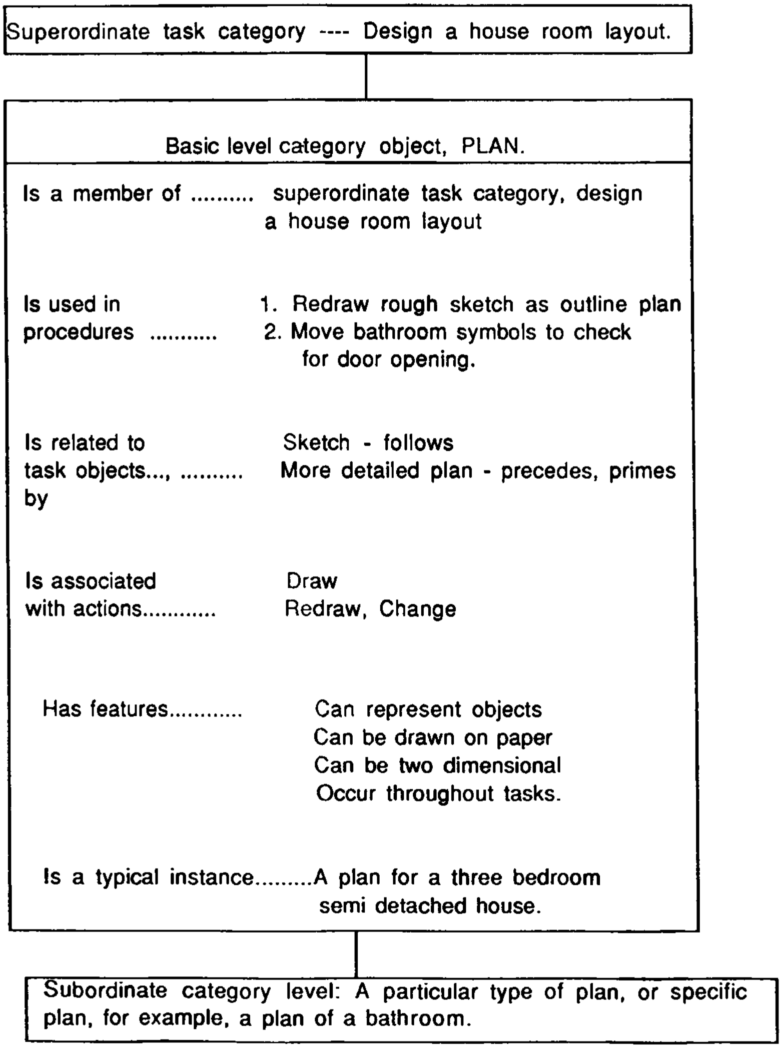


Fig. 3 Taxonomic substructure for the architectural task illustrating the basic level object, PLAN, and its relations to the superordinate and levels

Finally, the bottom level of the taxonomic substructure is the subordinate task category which contains a particular type of the object represented at the basic level, in this instance a type of plan. The hierarchy is shown in Fig. 3 using the example of a plan (i.e. sketch-plan) as the category member.

3.4 Embedded task rules and the construction of production rules

The taxonomic substructure in conjunction with the goal-oriented substructure provide the structure for the representational framework of the task model. The task rules are embedded in the taxonomic substructure and describe which objects go together or occur in close temporal sequence, and which objects are more important or representative than other objects. The nature of the task rules provides constraints on the combination of task elements. The task rules are called up by the procedures defined in a strategy. These procedures are then modelled by production rules.

At this time it is not intended that the production rules are syntactical in the sense of providing a grammar relating to where in an action sentence the individual elements can appear. An example of production rules that might model the procedures carried out in the architectural task are given in Fig. 4.

```
IF (architect has ability to design house layouts) AND
   (client wants house designing) THEN
   (architect sets goal 'design house layout')

IF (architect has goal 'design house layout') THEN
   (architect sets subgoal 'prepare general outline') AND
   (check financial feasibility)...................

IF (general outline to be prepared) THEN
   (set up client for briefing) AND
   (contact client)

IF (there is a phone number for client) THEN
   (ring to make appointment) ELSE
   (check for address in files) AND
   (write to client to make an appointment)
```

Fig. 4 An example of production rules for the architectural task

An alternative to production rules might be to use a grammar as in TAKD (Johnson, Diaper and Long, 1984), or frame-based representation such as those outlined in Preliminary Analysis for Design (Keane and Johnson, 1987). At present it is an empirical question as to which of these techniques prove to be more useful at this level of representation and the approaches will be considered in more detail in future research.

4 Conclusion

The framework presented here relies on category theory, general knowledge structures and other cognitive psychology phenomena to provide the rationale for making design recommendations and improving design usability. We believe that existing user knowledge will be maximised, leading to potentially less errors and easier task execution if the design of the system represents the task elements that have been regarded in this paper as being part of the knowledge of tasks. If representation of all task elements is not possible then at least the more representative or important actions and objects that have been identified must be represented. A prediction here is that the usability of the system will decrease proportionally to the number of representative or important objects or actions not represented at the interface.

Moreover, we also believe that the user interface design should not violate the usual sequence for carrying out the task(s). If, however, in extreme circumstances the sequence is violated, it should be done in a consistent and systematic manner, in order that the system responds in the expected manner and is easy for the user to learn. This objective can be achieved by supporting the different, previously identified strategies, which are employed by task performers in unusual or novel circumstances. These alternative strategies should ideally also be supported by the system.

Acknowledgements

The work reported here is part of an ICL, URC funded research project on 'Task analysis as a design tool'. The authors are grateful for the support and cooperation provided by ICL and colleagues at QMC. Requests for reprints should be addressed to the first author.

References

ANNETT, J. and DUNCAN, K.D. (1967): Task Analysis and training design. *Journal of Occupational Psychology*, 41, 211–221.

CARD, S.K., MORAN, T.P. and NEWELL, A. (1983): *Psychology of Human Computer Interaction.* Hillsdale, New Jersey, Lawrence Erlbaum Associates.

FLEISHMAN and QUAINTANCE (1984): *Taxonomies of Human Performance.* New York, Academic Press.

JOHNSON, H. and JOHNSON, P. (1987a): *The development of task analysis as a design tool: A method for carrying out task analysis.* QMC report No. 1 to ICL March.

JOHNSON, H. and JOHNSON, P. (1987b): *The development of task analysis as a design tool: Generification – a process of identifying generic properties of tasks within a given domain.* QMC report No. 2 to ICL June.

JOHNSON, H. and JOHNSON, P. (1987c): *The development of task analysis as a design tool: Task modelling – identifying the structure and rules for modelling tasks.* QMC report No. 3 to ICL October.

JOHNSON, P., DIAPER, D. and LONG, J. (1984): Tasks, Skill & Knowledge: Task analysis for knowledge based descriptions. In B. Shackel (ed.), *Interact84*, London, North-Holland.

JOHNSON, P. (1985): Towards a task model of messaging. In Johnson, P. and Cook, S. (eds.), *People and Computers: Designing the user interface*, Cambridge, Cambridge University Press.

KEANE, M. and JOHNSON, P. (1987): Preliminary Analysis for Design. In Diaper, D. and Winder, R., *People and Computers: 1987*, Cambridge, Cambridge University Press.

KIERAS, D. and POULSON, P. (1985): An approach to the formal analysis of user complexity. *International Journal of Man Machine Studies*, 22, 365–394.

KELLY, G.A. (1955): *The psychology of personal constructs*, New York, Norton.

ROSCH, E. (1978): Principles of categorisation. In Rosch, E. and Lloyd, B. (eds.), *Cognition and Categorisation*, New Jersey, Lawrence Erlbaum Associates.

PAYNE, S.J. and GREEN, T.R.G. (1986): Task-Action Grammars: a model of the mental representation of task languages. *Human Computer Interaction*, 2, 93–133.

The architecture of an automated Quality Management System

J.G. Walker
STC Technology Ltd.
B.A. Kitchenham
Centre for Software Reliability, The City University, London EC1V OHB

Abstract

This paper gives an overview of the Quality Management System which is being developed as part of the work of the Alvey Test Specification and Quality Management project and of the ESPRIT REQUEST project. This system will provide a set of tools to support the quality management of software development projects.

These tools are concerned with the specification of requirements for the non-functional attributes of a product, and the evaluation of such specifications; the creation of quality plans, verification and validation plans, and metrics plans; the monitoring of quality during product development; and the assessment and prediction of final product quality.

The paper summarises the aims of each tool, and the anticipated benefits to the user. It also indicates the current progress with the development of the prototype system.

1 Introduction

The authors of this paper have been involved in two research projects concerned with the issue of software quality: the Alvey-backed Test Specification and Quality Management (TSQM) project, and the ESPRIT-backed REQUEST project.

One of the goals of TSQM is to develop an automated Quality Management System (QMS), which will assist a quality or project manager throughout the life of a software project. It is intended to have a common interface to a set of subsystems which assist project initiation, project monitoring, and project assessment.

One of the goals of the REQUEST project is to develop a constructive quality model, called COQUAMO [PETE87], which will assist a manager to achieve specified quality requirements in a software product by providing improved methods for modelling software quality.

Both TSQM and REQUEST have contributed to the general concept of the QMS; TSQM is developing tools which support quality management planning and specification activities, while REQUEST is developing models which assist quality assurance and control by providing quantitative assessment of product quality.

This paper describes the tools which the QMS is intended to provide, and how these can help project and 'quality' staff.

2 The QMS as an aid to quality assurance

Readers of this Journal are presumed to be familiar with the concepts of QA, QC and QM, where Quality Assurance embraces all the planned or systematic actions necessary to provide adequate confidence that a product or service will satisfy given needs; Quality Control provides the means by which quality is achieved; and Quality Management is the aspect of the overall management function that determines and implements the quality product.

The authors of this paper believe that the quality-related goals of a project will be more readily achieved if there are suitable tools whose use *automatically* supports these goals.

Thus the intention in designing the QMS is to provide tools which make it easier for project staff to carry out their tasks, and at the same time ensure that they do so in a way which will satisfy quality assurance requirements.

3 Tools for quality management

The provision of tools to support QA and QM must explicitly or implicitly help to train staff in the use of established standards; must offer working practices based on these standards; and must provide a method of auditing conformance to these standards. These objectives provide the justification for the work which TSQM and REQUEST have given to developing the QMS.

The QMS provides a common interface to a set of subsystems intended to assist the tasks of quality management and quality control, by providing tools which are geared to the requirements of quality assurance. These subsystems assist project planning and initiation, project monitoring, and project assessment. An overall diagram of the system is shown in Fig. 1.

In summary the QMS is intended to provide the following tools:

- Project Planning and Initiation, which includes three subsystems
  - the Planning Assistant Subsystem: this results in the creation of quality plans; verification and validation plans and testing strategies;

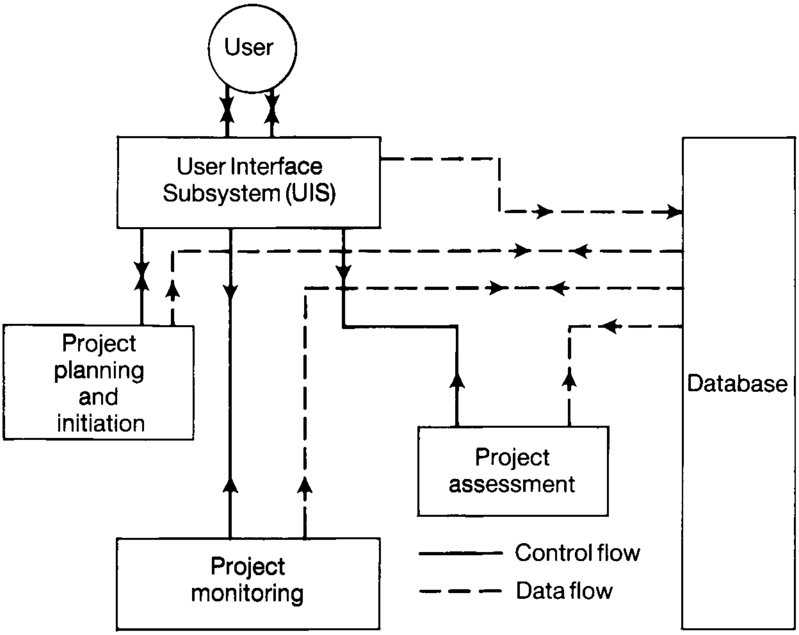


Fig. 1 The major components of the QMS and the links between them

and metrics plans, identifying the metrics which will be gathered to support project monitoring.
- the Quality Requirements Specification Subsystem: this results in the establishment of quantified and measurable targets for the quality factors (non-functional attributes) of a product; it also offers advice on the viability of these targets, and on the techniques which could be used to achieve them.
- the Quality Prediction and Feasibility Analysis Subsystem: this uses quality "drivers" critical to the success of the project, e.g. the experience of project personnel, leadership style, professional environment, product complexity, etc.; to provide a prediction of final product quality in terms of the quality requirements specification information.

- the Project Monitoring Subsystem uses the metrics data to provide reports and advice on the status of the project in quality terms during the course of its development.
- the Project Assessment Subsystem uses data obtained during the early life of the product to report on the achieved quality of the product, and on the anticipated overheads for maintenance and support.

In what follows, we look at some of these tools in greater detail, and describe how they are intended to help the user. The description frequently implies that the tool already exists; this is not necessarily the case. A fuller discussion of the state of development of the prototype QMS is given in section 8 below.

4 Requirements specification and evaluation

Attempts to specify the non-functional attributes of a product are frequently couched in vague, ill-defined terms. Expressions like 'highly reliable', 'easily maintainable', and 'very user-friendly', are all too common. In order to get away from such subjective expressions, a more disciplined and quantified approach is called for.

4.1 The Quality Requirements Specification Subsystem (QRSS)

In the authors' experience, people have had the greatest difficulty in defining the non-functional attributes of a product, which historically have been regarded as contributing to the 'quality' of a product, in objective and measurable terms.

The Quality Requirements Specification Subsystem (QRSS) addresses this by providing a set of definitions for each of nine quality factors. The factors are extendability, integrity, maintainability, performance, reliability, reusability, security, survivability, and usability.

Each factor is considered in turn, and the QRSS offers the user a quality factor template which contains a set of standard headings, some of which define or provide other information about the factor, and some invite the user to insert the information which specifies the measurable target (or targets) for that factor.

For each factor definition there is one or more 'explosion'. An explosion is a way of viewing a particular quality factor which enables a measurable planned value to be set for that factor. Each explosion is described, and has its own specifiable measuring unit; a tool for measuring this; conditions which apply to the measurement process; and planned and worst values.

By systematically reviewing the explosions for all nine quality factors, and setting specific quantified and measurable requirements for any or all of these according to what is most appropriate for the product concerned, the user is enabled to build up a full requirements specification for these non-functional attributes.

As an example of this, Fig. 2 shows an extract from the quality factor template for Usability. In Fig. 2, ⟨⟩ denotes a field into which the user types the information specific to the particular product. In this example the user is given a choice of two definitions of usability, for each of which an explosion is given. Users can set targets for usability against either or both of these. In addition, or instead, they can specify further explosions of their own choosing, and/or further definitions as well, if they so wish.

In Fig. 3, an example is given which shows how this template might be filled in for a hypothetical case where usability is defined in terms of the time taken by graduate recruits to learn to use the product.

Usability template	
Classification	general quality – application dependent
Level required	⟨level⟩
Associated qualities	
– synonyms	learnability, operability, user friendliness
– related concepts	understandability
Definition 1	the ease with which users can learn to use the system.
Explosion 1.1	the average time for identified classes of user to achieve a specific level of competence with a system.
Measuring units	⟨time units⟩
Measuring tool or data source	⟨tool⟩
Measurement conditions	the classes of user to be included are – ⟨experienced⟩⟨naive⟩⟨operators⟩⟨other⟩ the number of people required for the test is ⟨integer⟩
Worst case	⟨value⟩
Planned level	⟨value⟩
Best case	⟨value⟩
Current level	⟨value⟩
Justification	⟨text⟩
Consequences of failure	⟨consequences⟩
⟨further explosions⟩	
Definition 2	the extent to which users of the system are satisfied with its facilities.
Explosion 2.1	proportion of users expressing themselves reasonably satisfied with the system
Measuring units	percentage
Measuring tool or data source	⟨tool⟩
Measurement conditions	features to be assessed are ⟨text⟩ method of conducting survey is ⟨text⟩ the minimum number of replies required is ⟨integer⟩ the criteria for establishing an overall assessment from survey replies are ⟨text⟩
Worst case	⟨value⟩
Planned level	⟨value⟩
Best case	⟨value⟩
Current level	⟨value⟩
Justification	⟨text⟩
Consequences of failure	⟨consequences⟩
⟨further explosions⟩	
⟨further definitions⟩	

Fig. 2

Benefits to the user: it provides a method and a tool for producing quantified and measurable targets for the non-functional attributes of a product.

4.2 Coarse evaluation of a specification

Additionally, the QRSS will offer an initial assessment of this specification in terms of any potential conflicts that it may contain, and of the notional

Usability example	
Classification	general quality – application dependent
Level required	high
Associated qualities	
– synonyms	learnability, operability, user friendliness
– related concepts	understandability
Definition 1	the ease with which users can learn to use the system.
Explosion 1.1	the average time for identified classes of user to achieve a specific level of competence with a system.
Measuring units	days
Measuring tool or data source	survey forms and results of survey
Measurement conditions	the classes of user to be included are – new graduate intake the number of people required for the test is 20
Worst case	8
Planned level	4
Best case	2
Current level	10
Justification	the HMI improvements proposed in report ABC/123.
Consequences of failure	cost

Fig. 3

orders of cost that its implementation will involve. It will also offer advice on the implementation techniques that may be appropriate to achieve the specified levels of the different quality factors. On the basis of the techniques selected, it will give a broad assessment of the probability of success of achieving the specified requirements. More information is available in [WALK87a].

Benefits to the user: it provides a (speculative) method and tool for predicting the percentage chance of success of implementing a product with a given specification of non-functional attributes, where these have been assigned a priority on a scale from low to very high; it indicates the likely order of magnitude of 'cost', expressed as a percentage of the basic cost of implementing the product without regard for these attributes; it also offers advice on development techniques likely to assist the realisation of the specification.

4.3 The Quality Prediction and Feasibility Analysis Subsystem

The Quality Prediction and Feasibility Analysis Subsystem offers a more finely tuned prediction of the likely achievement of the specified requirement targets. This is based on the REQUEST project's COQUAMO model (COQUAMO-1), which uses information about quality "drivers" (such as product complexity, project leadership style, working environment, etc.) and how these affect the different quality factors. The values for these quality drivers are derived from the plans and constraints for the project.

Benefits to the user: it provides a method and a tool for predicting the likely achievement relative to quantified targets for the non-functional attributes of a product, based on ranking various 'environmental' aspects of the project on a scale of 1–4.

5 Planning for quality

People will usually accept that it is reasonable to have some sort of a plan to guide them on a project. Something to say what the project is aiming to achieve in functional terms, the timescales involved, what resources are needed and available, and perhaps some details of the main dependencies.

The same people, unfortunately, may not see the same need to plan the management of quality. Somehow, they feel, that will take care of itself.

Within the Planning Assistant Subsystem, the QMS offers tools to enable the user to create Quality Plans, Verification and Validation Plans, and Metrics Plans, which together go some way towards correcting this omission. The following notes briefly introduce each of these tools in turn, and the plans which they produce. More information is available in [FREW86] and [GRES87].

5.1 The Quality Planner

The Quality Planner presents the user with a structure for a Quality Plan, and a template for each section of the structured plan. The templates offered by the prototype QMS reflect most of the information required by the AQAP [AQAP1,13,14] and IEEE [IEEE84] standards. It is expected that in due course the QMS will offer a range of standards for users to choose from, including the possibility of standards tailored to their own organisations' specific requirements.

The structure includes sections to describe project details, codes of practice, quality control, problem reporting, configuration management, and other aspects of quality planning which need consideration.

Each section of the quality plan is introduced by a statement explaining its purpose and what it should contain. This appears only on the QMS screen, and is not reproduced in the finished plan. In most cases there is then a template which provides standard narrative for that section, embedded in which are fields into which the user puts information specific to the project being planned. 'Help' text is associated with each template, in case further guidance is needed as to precisely what is intended to be put in these various fields.

As an illustration of what is provided, the following is an excerpt from what the user sees on screen when considering section 2 of the Quality Plan: the description of the project. The user is prompted as to the purpose of this section by the following text:

"This is where a link is made between the various documents available and their relative importance. You should explain the nature of the project itself and how it fits into an overall strategy. This section should however be brief and adhere to the following guidelines as shown in the example template."

The template for this section is as follows:

"2. Description of the Project.

Project aims: ⟨text⟩
Requirements specification: ⟨reference⟩
Quality requirements specification: ⟨reference⟩
Other documents to be read in conjunction with this document:
⟨e.g. Feasibility studies⟩
Deliverable items: ⟨list⟩
Dependencies: ⟨list⟩
Project contracts: ⟨text⟩"

where ⟨ ⟩ denotes a field into which the user types the information specific to the particular Quality Plan.

There is further information available as to what is required, in the form of 'help' texts. For example, if the user is unsure of what is meant by 'Dependencies' in the template shown above, the following 'help' will be displayed:

'Dependencies: in this section you have the opportunity to state dependencies made on other projects, software, hardware, etc. You may also highlight products, etc. that are dependent on this project. You should identify each item by its identifier as well as its name.'

The prototype Quality Planner is intended to demonstrate the viability of the method; the detailed contents (e.g. plan format and structure) can in principle be tailored to meet an organisation's requirements.

Benefits to the user: it provides a method and a tool for producing a Quality Plan to a standard format.

5.2 The Verification and Validation Planner

The Verification and Validation Planner enables the user to create a specific Verification and Validation (V & V) Plan for a project. It provides a structure for the plan based on the IEEE standard [IEEE86], and presents a template for each section of the plan. The user is required to identify the 'deliverables' for each stage in the development life cycle, and state which V & V activities are associated with each deliverable at each stage. As with the Quality Planner, templates are provided for the various sections, together with 'help' information.

There is also a tutorial component which offers the user advice on general aspects of V & V: the application of V & V at each stage in the development cycle; the checking and testing methods available at each stage; a strategy for V & V; and so on.

To support the V & V Planner there is a database of information on various checking and testing methods. This gives, for each named method: a brief description; the quality factors most affected; the life cycle stage(s) to which it applies; entry, exit and evaluation criteria; constraints and implications, etc.

Benefits to the user: it provides a method and a tool for producing a V & V Plan to a standard format. It also provides advice on appropriate checking and testing methods to use at different stages in the development cycle, and on a strategy for verification and validation.

5.3 The Metrics Planner

The Metrics Planner enables the user to plan the metrics which are to be collected during the development of a product. Unless there is a plan to collect the required metrics, it is most unlikely that they will be available when needed to help monitor the progress and quality of the developing product.

It enables the user to create a specific metrics plan, identifying the metrics to be obtained at each phase of the development cycle. There is also a tutorial component which offers the user advice on the reasons for gathering metrics, the types of metric data which should be considered at each phase, and how these can be used, including specifically their role in relation to the QMS project monitoring subsystem.

Benefits to the user: it provides a method and a tool for producing a Metrics Plan to a standard format. It also provides advice on appropriate metrics which can be obtained at different stages in the development cycle, and the use that can be made of these.

6 Monitoring projects for quality

During the development of a product, there is much information potentially available by which the state of the product can be assessed. The QMS Project Monitoring Subsystem is intended to use such information to provide reports for project and quality management. It is based on the REQUEST project's COQUAMO model (COQUAMO-2). The reports produced will indicate the current status of the project in summary form; any trends which have been detected; and any specific exceptional situations which appear to need attention, as indicative of potential quality problems.

In assessing whether there is any reason to highlight an exception, the subsystem can potentially compare the status of the product with its plans, or with historic data on other similar products; it can additionally compare data on all elements of the product which have reached the same stage of development, in order to indicate any which may appear to be anomalous. Any such exceptions are good reasons to alert management to the need to investigate the situation further.

In addition to providing summary, trend and exception reports, the subsystem will offer advice on the possible causes of any exceptions identified. For example, consider a program module which shows a much higher bug count per 100 statements than other modules at the same stage of development. This may be because:

- it is an intrinsically difficult module to code
- it may have been badly designed
- it has been produced by someone with little experience
- it has been more thoroughly tested than other modules.

The Project Monitoring Sybsystem cannot say for sure which of these reasons, if any, is or are correct, but it can offer a list of possibilities; in some cases it may also be able to use environmental data (see section 4.3) to infer which of these possibilities are the most probable. In the above example, if it is known that the project staff are of above average experience, then the second and third possible reasons are less likely than the other two.

The subsystem does not pre-empt management judgement: it provides what 'advice' it can about the reasons for an apparent anomaly, and offers ideas for appropriate short and long term remedial action. The responsibility then rests with management to determine what, if anything, to do about it.

More information on this approach is contained in [KITC86] and [KITC87b].

Benefits to the user: it provides a tool to monitor the metrics that are available during product development, in order to alert management to any potential quality problems which may be identified from these metrics, their possible causes and appropriate remedial action.

7 Project assessment

During the system testing and early operational use of a product, the behaviour of the product can be observed directly. It is then possible to verify whether it does indeed conform to its stated quality requirements. The QMS provides a separate Project Assessment Subsystem to assist with this verification process, because the metrics and models needed to verify final product characteristics are very different from those which may be used to monitor project progress. This subsystem is based on the REQUEST project's COQUAMO model (COQUAMO-3).

The assessment subsystem permits the requirements specification to be compared with the observed characteristics of the final product. The metrics and models provided will therefore be determined by the general product qualities defined in the requirements specification. However, it is intended that the subsystem does more than simply verify that requirements have been met. It is also intended to include the following features:

- provision for analysing and possibly reporting software failures and faults, to conform with the requirements for establishing a problem analysis procedure stated in various quality standards (e.g. AQAP13 and BS5750),
- provision for estimating the elapsed testing time necessary to demonstrate that reliability requirements have been met; if they have not yet been achieved, the subsystem should be able to provide an estimate of the elapsed testing time needed to do so.
- provision for using the data collected to verify final product characteristics, and to estimate future support characteristics of the product (e.g. the costs of support, the effort required for fault diagnosis, the cost of enhancements, etc.).

Benefits to the user: it provides a tool to help assess whether the product is fit for release, and to estimate what the likely support characteristics will be.

8 The implementation

8.1 The approach to QMS development, and progress to date

The QMS is being developed and made available in incremental stages. For each component of the system, work progresses from theoretical studies, through specification and design, to prototype implementation; the implementation is then evaluated, using appropriate skills from outside the project, and the prototype is thereafter refined and improved as necessary in the light of the evaluation reports.

The QRSS (see section 4.1 above) has gone through the full process just described. It benefited greatly from the feedback gained from an early evaluation, which resulted in the production of a much improved version (as confirmed by a further evaluation). Some information about how it has been implemented is given in the next section, which also covers other elements of the system not described elsewhere in this paper.

The coarse evaluation of a specification (section 4.2) has been designed and awaits implementation effort, as does the Quality Planner (section 5.1). Work is currently in hand to design the Verification and Validation Planner (section 5.2), and the Metrics Planner (section 5.3). All the above work has been or is being carried out within the Test Specification and Quality Management project.

The work on Quality Prediction and Feasibility Analysis (section 4.3), Project Monitoring (section 6), and Project Assessment (section 7), is still mainly in the theoretical stage, and is being progressed primarily within the ESPRIT REQUEST project at present.

8.2 The implementation so far

As shown on Fig. 1 in section 3 above, the user's entry to the QMS is via the User Interface Subsystem (UIS). The UIS and the QRSS have been implemented in the prototype QMS so far, along with some very high level 'help' in the form of an introduction to quality management and to the facilities of the QMS.

The intention in implementing this prototype QMS is to demonstrate the viability and, it is hoped, the utility of the various methods and tools proposed. It is anticipated that these will be refined as a result of pilot use via the prototype, and will subsequently be improved in any 'production' version.

This prototype runs under UNIX* on the ICL DRS300. The elementary User Interface Subsystem which currently exists is a UNIX shell script, which controls the user's initial interface to the QMS by means of a menu. This offers the user the choice of whatever facilities are available on the system: at present this is limited to the QRSS and the Introduction to Quality Management. The user indicates which facility is chosen, and the UIS sets this running. When use of that facility is ended, control returns to the UIS, which re-displays the original menu so that the user can make a further selection, or else terminate that QMS session.

The Introduction to Quality Management talks about the terms associated with quality management, quality assurance, and quality control, and describes the facilities of the QMS – both those implemented and those planned. It is provided as a set of text files which are displayed under the control of a UNIX shell script. This offers the user a menu listing the 'chapters' of text. The user selects one of these at a time and browses through it, page by page. The order of chapter selection is entirely at the user's discretion. When the user has had enough of this, there is a quit option which returns control to the UIS menu.

The specification part of the QRSS has been implemented using a customised version of the STC Generic Toolset, itself the product of an Alvey project, see [HOOK86]. The Toolset is a structured editor generator, so the QRSS implementation is achieved via the construction of a grammar, which includes the text of the various quality factor templates. This is a mixture of narrative (e.g. the definition of a quality factor), and the framework of embedded fields for which the user will subsequently supply values. The types of value which may be inserted into any specific field are prescribed by the QRSS grammar.

*UNIX is a registered trademark of AT&T in the USA and other countries.

Using the generated editor, the user is able to create and browse through a specification, starting with any one of the nine quality factors. By using the cursor movement keys, the user can either work through the text line by line; skip from one specifiable field to the next; or return to a higher level (to another quality factor, say) and move on from there. The user creates a specification step by step, inserting values into the quality factor templates.

As an example, the planned level for a particular 'explosion' appears on the screen as

planned level: ⟨value⟩

The user has the option of specifying the 'value'. In this particular implementation, 'value' has been defined so that the user has the choice of specifying it as an integer, or a real number, or as a form of approximation – which may be the most realistic value that can be specified at that stage in the project. Examples of the sort of approximation allowed are: 18 ± 2; 250?; 75–130.

Once the user has specified the 'value' for this particular field, that is what will appear – on screen or in a hard copy print-out – until the user either deletes it or alters it, e.g.

planned level: 60

The user may indicate that a factor does not apply.

The system allows a short help text to be displayed at each point, and extended help information can be provided where necessary (by accessing separate text files). The current implementation provides short help in the form of a brief prompt as to the permissible types of entry for the current field. The longer form of help, obtained by invoking a specific text file, provides on-line extracts from the QMS User Guide.

The user can create a specification, save it when partially or totally complete, subsequently amend it, print it out, and eventually delete it. As the printable form of the specification is an ordinary text file, it can be edited further outside the QMS. This makes it relatively simple to embed the specification in a report format of the user's choosing.

Further information about the implemented system is available in the QMS User Guide [WALK87b].

The implementation to date has been achieved by using a mixture of UNIX shell scripts, text files, and the STC Generic Toolset. This in no way constrains the choice of tools for implementing further components of the QMS. It is intended, rather, to select whichever tools are felt to be most appropriate for each subsystem; blending the resulting modules into the existing system via the UIS.

9 Conclusions

With the current emphasis on the development of IPSEs (Integrated Project Support Environments), there has been a good deal of agreement that software engineers need better tools as well as better techniques to achieve significant improvements in software quality and productivity. However, software development, like other complex industrial activities, demands proper project management and quality assurance as well as good working practices, but there has been less appreciation of the need for better project management and QA techniques and tools.

One of the authors has argued elsewhere [KITC87a] that the nature of software development makes greater demands on management than other more conventional engineering projects, and that conventional management techniques and tools are insufficient to deal with the problems of software development. The arguments applied to software project management apply with equal force to software QA, which, traditionally, has been undertooled, even in conventional engineering projects. The Quality Management System described in this paper is an attempt to provide some of the automated techniques needed to support software QA. The QMS provides a framework within which both traditional QA activities can be automated, and the more novel techniques being developed by TSQM and REQUEST can be made available to quality managers and project managers.

In particular, the QMS ensures conformance to standards by building the standards directly into a tool which assists the production of documents, and/or the performance of activities, to that standard. This means that the tasks of learning a standard, and learning to produce an item, or to perform a task, in conformance to that standard, are combined into the single task of learning to use the QMS. This approach is particularly relevant to the planning subsystem, and the part of the project monitoring and project assessment subsystems devoted to recording and analysing product problem reports.

In addition, auditing conformance to standards is simplified, since conformance to standards can be demonstrated by verifying that the tool has been used (either by providing a software audit trail or by inspection of date stamped output from the tool), rather than by reading documents and/or observing various development activities.

The more novel elements of the QMS such as specification and evaluation of quality requirements, monitoring project progress using software metrics, and utilising reliability and/or support cost models, cannot be considered as traditional QA and QM techniques. However, we believe that there is a clear need to develop improved techniques and that the concepts being developed by TSQM and REQUEST provide some useful advances. The inclusion of these techniques in the QMS allows the ideas to be evaluated by practising quality managers, and will hopefully lead to useful dialogue between the TSQM and REQUEST researchers, and the potential users of that research.

Acknowledgements

The authors of this paper were partially funded by the Alvey Directorate and the ESPRIT programme.

The authors of this paper would like to acknowledge the contribution that Tom Gilb's ideas have made to their thinking on how to specify non-functional attributes.

References

[AQAP1] AQAP-1: NATO requirements for an industrial quality control system.

[AQAP13] AQAP-13: NATO software quality control system requirements.

[AQAP14] AQAP-14: Guide for the evaluation of a contractor's software quality control system for compliance with AQAP-13.

[BS5750] British Standard on Quality Systems.

[FREW86] Quality management – procedures and practices, G.D. Frewin and B.J. Hatton, IEE/BCS Software Engineering Journal, Jan. 1986, pp. 29–38.

[GRES87] Design of a QMS subsystem for quality planning, C.A.E. Greswell, Alvey deliverable from the TSQM project, Report A26.1, 1987.

[HOOK86] VDM Toolset Userguide, P. Hook, 725 07831, Ed. 1, 1986.

[IEEE84] IEEE standard for Software Quality Assurance Plans, Std. 730, 1984.

[IEEE86] IEEE standard for Software Verification and Validation Plans, Std. 1012, 1986.

[KITC86] An information model for software quality management, B.A. Kitchenham and J.G. Walker, Alvey deliverable from the TSQM project, Report A24, 1986.

[KITC87a] Controlling software projects, B.A. Kitchenham, Electronics and Power, May 1987, pp. 312–315.

[KITC87b] A quantitative approach to monitoring software development, B.A. Kitchenham and J.G. Walker, draft submitted for publication 1987 (copies available from the authors).

[PETE87] Software quality: the COnstructive QUAlity MOdelling System, P.G. Petersen, ESPRIT'86 – Results and Achievements, Directorate General XIII (Editors), Elsevier Science Publishers B.V. (North Holland), 1987.

[WALK87a] Quality requirements specification and evaluation, J.G. Walker and B.A. Kitchenham, presented at the CSR symposium on Measurement for Software Control and Assurance, 1987, and to appear in the published proceedings.

[WALK87b] User Guide to the QMS at version 1.1*, J.G. Walker and C.A.E. Greswell, Alvey deliverable from the TSQM project, Report A25(1.1*), 1987.

ICL Company Research and Development Part 2: Mergers and Mainframes, 1959–1968

Martin Campbell-Kelly
Department of Computer Science, University of Warwick

Abstract

This paper describes computer developments during the years 1959–68, which culminated in the formation of ICL. The period 1959–63 saw a series of mergers in the British computer industry which resulted in the formation of three firms: ICT, English Electric-Leo-Marconi and Elliott-Automation. In 1964, IBM launched its highly successful System/360 range of computers, to which the two larger British firms responded with the ICT 1900 series and the English Electric-Leo-Marconi System 4. As American manufacturers came to dominate the British scene, however, the Government intervened to achieve a final rationalization of the British computer industry, by promoting the formation of ICL in 1968. To achieve this final consolidation, the Government provided research and development funding for a new range of computers for the 1970s.

1 Introduction

A previous article[1] described research and development in the two British punched card machine companies, the British Tabulating Machine Company (BTM) and Powers-Samas. In January 1959, the two companies merged to form International Computers and Tabulators Limited (ICT). The new company was the largest non-American supplier of data processing equipment, with annual revenues of about £25 million, nearly 20 000 employees, and over 20 factories. Apart from the economies of scale to be derived from rationalizing the development, manufacturing and selling operations of the merged companies, a prime objective of the merger was to diversify into electronic computers and computer peripherals.

At the time of the merger, however, ICT was by no means the dominant British computer manufacturer. During the 1950s, Elliott-Automation, Ferranti, Leo Computers and English Electric had all secured larger market shares. Also, by the end of the 1950s, most of Britain's electronics companies had decided to enter the computer business: these included AEI, Decca, EMI, GEC, Marconi, Plessey and STC.

Thus, by the early 1960s, the scene was ripe for merger activity for two reasons. First, there were simply too many firms competing for a domestic market that, in 1960, amounted to a mere £5 million; it is doubtful if any firm was so much as recouping its research and development costs, let alone making money in computers[2]. The second cause of merger activity was the onset of American competition. In the 1950s, American penetration of the UK computer market had been negligible, but by 1967 – perhaps the most critical year in the development of the industry – US manufacturers such as IBM, NCR, Univac, Burroughs and Honeywell had come to take nearly 70 per cent of the British market; IBM alone had over 40 per cent of the market in that year[3].

There were essentially two merger waves (Fig. 1). The first occurred in 1962–63, which resulted in three British firms: ICT (which absorbed the computer interests of GEC, EMI and Ferranti), English Electric-Leo-Marconi, and Elliott-Automation. The second merger wave took place in 1967–68: Elliott-Automation was first absorbed by English Electric, and then ICT and the EDP computer interests of English Electric Computers were merged to form International Computers Limited.

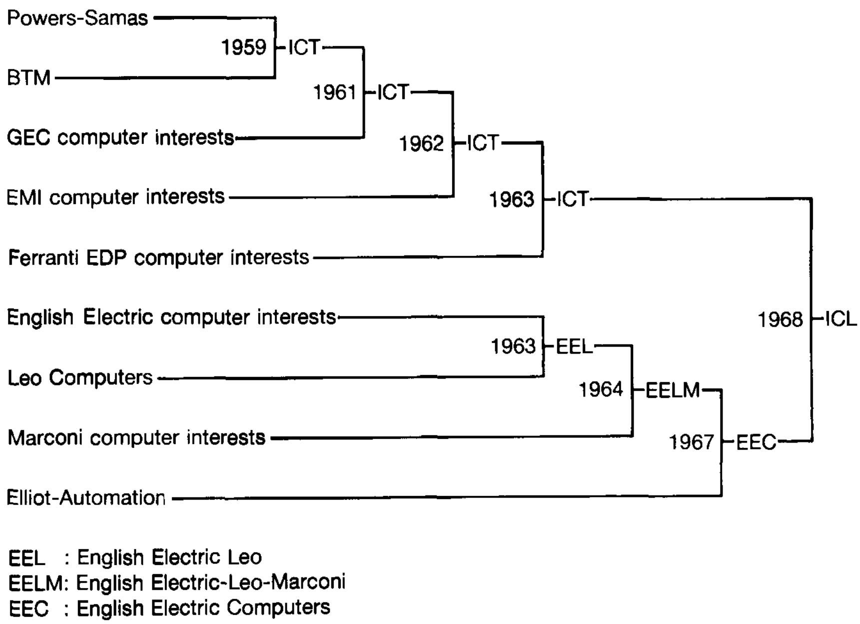


Fig. 1 Evolution of ICL, 1959–68

In fact, the progress of the mergers was less rational than Fig. 1 perhaps suggests. Talks were held between all the players on many occasions, and the partnerships and sequence of events that materialized could easily have been very different. However, the eventual outcome – a single British flagship computer company – was perhaps inevitable, given the political and economic climate of the 1960s. The political and economic imperatives that underlay the mergers, and the personalities involved, will be described in more depth in the author's forthcoming book on the history of ICL; this paper will focus largely on the technical issues.

2 Tabulators and second generation computers

Although the eventual ascendancy of computers over tabulators was implicit in the name International Computers and Tabulators, punched card machinery took much longer to fade away than is commonly supposed. At the time of the 1959 merger, ICT derived 90 per cent of its revenues from the punched card machine business and only 10 per cent from computers. Even in America, which was at least two years ahead of the UK in the diffusion of computers, two thirds of IBM's domestic business was based on traditional electro-mechanical accounting machinery. Thus, although it was recognized that the future lay with computers, it was vital for ICT to protect the short term competitive position of its punched card machines in order to secure the foundations on which its future computer business would be built.

At the time of the merger, BTM had seen great promise in the Powers-Samas Samastronic tabulator, which printed at the then unprecedented speed of 300 lines per minute. It did not take long after the merger, however, for experienced former BTM engineers to discover that the electro-mechanical design of the Samastronic arithmetic units was an amateurish mess, and that there was no real prospect that the machine could ever be made to work with acceptable reliability. Many machines prematurely released to the field in the last months of Powers-Samas's existence had either to be withdrawn or maintained at enormous cost. Even the print unit was unable to deliver reasonable print quality, and so ICT's hopes for developing a computer printer from the Samastronic had to be abandoned, leaving it without an acceptable product for several years. In November 1960 the ICT board took the decision to put an end to the Samastronic once and for all. So far as the balance sheet was concerned, the total cost of the debacle was put at about £1·5 million, a figure that exceeded ICT's entire annual research budget. In fact the position was to prove even worse in the long term, because in order to fulfil the cancelled Samastronic orders – there were well over 200 – output of 80 column equipment had to be increased at great cost, or the business would have been lost to IBM. The failure of the Samastronic also cost ICT dearly in goodwill, and a product that might well have secured some competitive advantage against IBM, albeit temporary.

Approximately half of ICT's research and development funds were allocated to punched card machine developments. The biggest development was the model 975 calculating tabulator. This machine was intended to be the functional equivalent of IBM's second generation and highly successful model 628 calculating punch. The calculating tabulators – the main first generation examples of which were the BTM 555, the Powers-Samas PCC and the IBM 604 – were an important bridge between the traditional electro-mechanical accounting machine and the new stored-program computer. But because they occupied 'that ill-defined boundary which divides computers from calculators'[4], they were generally ignored by analysts of the 1960s computer scene. This tends to indicate a much sharper divide between computer and non-computer EDP equipment than was truly the case, and ICT classed these machines as 'small computers', which from a marketing point of view they were. Indeed, ICT derived more revenue from this class of machine than from all its computers put together during the first half of the 1960s.

In 1959, ICT's only marketable stored program computers were the models 1201 and 1202; these were small and slow first generation machines based around a drum memory with a maximum capacity of 4096 words (Fig. 2). BTM, at the time of the merger, had a number of computer developments nderway, whereas Powers-Samas had nothing of any significance. This was one of the main factors which determined the relative valuation of the two companies at the time of the merger: whilst the tangible assets contributed by each company were approximately equal, their respective shareholders received 62 and 38 per cent of the equity of the new company.

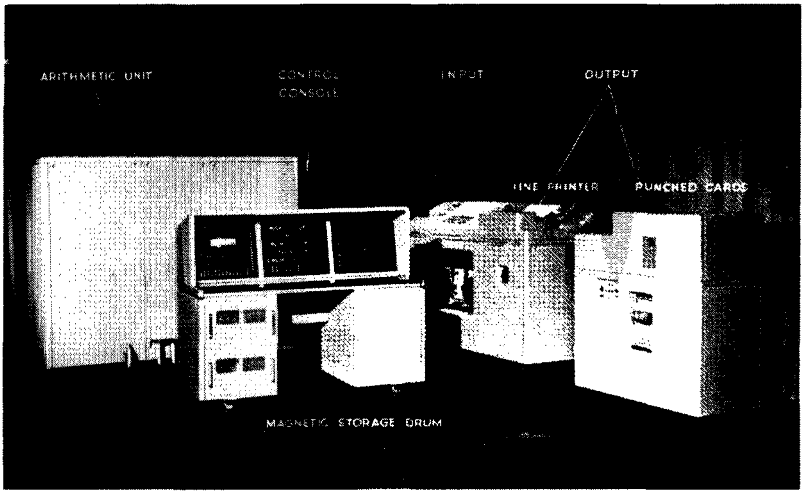


Fig. 2 ICT 1201 computer, c. 1959

Computer related research and development had two aspects. First, the development of small/medium EDP computers, which it was expected that the more advanced punched card machine users would gravitate towards during the next few years. Second, the development of computer peripherals for ICT's own computers and for sale on an OEM basis to other manufacturers; the manufacture of electro-mechanical peripherals was the ideal exit path from making traditional punched card machines.

The computer plans inherited from BTM were based on two machines, the 1300 series and the 1400 series, and a random access file drum. The 1300 series was to be offered initially in two models: the 1301, a small/medium system, and the 1350, a random access version; smaller and larger versions, the 1300 and 1302, were to be introduced subsequently. The 1400 series was to be offered in two models: the 1400, a medium-sized tape based system, and the 1450, a random access version. Had these plans matured, ICT would have had an excellent range of equipment, but during 1960 much of the commercial potential evaporated due to technical obsolescence and development delays.

The 1400 series 'balanced data processing computer' was ICT's prestige computer project. It had been heavily publicized since 1958, but during 1959 not a single sale had materialized. The 1400 was a first generation computer based on thermionic valves, and this technology was being rapidly overtaken by the new transistor electronics. The 1400 could not compete against second generation machines such as the IBM 7090 or the EMI 2400. The entire project had to be scrapped and the prototype sold off to Dr Andrew Booth at Birkbeck College for a token £5000.

Another major disappointment occurred with the random access file drum derived from BTM's collaboration with the Laboratory for Electronics in Boston. Up to 1960 this had seemed competitive against the IBM RAMAC disc store used in the IBM 305 computer. But in that year IBM announced the model 1301 disc drive with a capacity of 56 million characters and an average access time of 165 milliseconds[5]; against this specification, BTM's file drum with a capacity of 2 million characters and an average access time of 1 second, had no real commercial potential, and the plans for random access machines had to be put into abeyance. ICT was, in fact, one of a number of casualties of IBM's pre-eminence in disc drive technology: RCA was developing a magnetic card based random access store (the RACE), and Univac a drum based system (the FASTRAND drum), neither of which were successful[6].

The 1301 computer, although developing slowly, was in considerably better shape, in that it was a second generation machine based on transistors. The computer had been designed by the ICT and GEC joint subsidiary Computer Developments Ltd (CDL) formed in 1956, and was being developed and manufactured in GEC's Coventry telecommunications factory. The relative success of the 1301 served to underline the value of collaboration with an electronics manufacturer with an awareness of the component business. The 1400 had been developed entirely in ICT's own Stevenage electronics laboratory. In addition to the 1301 computer project, in 1959 ICT had a string of peripheral developments in progress: card readers and punches, paper tape devices, printers and console typewriters. ICT also had a fifty per cent shareholding in Data Recording Instruments Limited (DRI), a small company set up to develop magnetic storage systems. So, despite the disappointments of the 1400 series and the file drum, in terms of the traditional time scales of the British punched card machine industry, ICT's position did not seem unduly alarming. Although the short term position in computers was weak, since the first generation 1200 series was rapidly approaching obsolescence, computers accounted for such a small fraction of the business that it appeared of little immediate consequence. The fact was that in 1959 and 1960 the punched card machine business was extraordinarily buoyant. For example, two year delivery times for ICT 80 column equipment were being quoted, and the main anxiety within ICT was to increase production to avoid sales losses to IBM, who were able to quote six month deliveries.

It would be a harsh judgement to accuse ICT of complacency in computer development during 1959. Although computers accounted for only 10 per cent of ICT's output, it was devoting a full 50 per cent of its research and development budget to them.

3 The switch to computers

The event which completely transformed the outlook for computers, and precipitated the collapse of the punched card machine market, was the announcement of the IBM 1401 computer in October 1959. The 1401 was originally intended by IBM to be a second generation successor to its first generation model 650, in much the way that ICT's 1300 series was intended as a successor to its 1200 series. But the 1401 captured the American EDP computer market to an extent that took IBM by surprise, and exceeded all forecasts: a thousand orders were taken in the first few weeks following the announcement, and the machine went on to sell a total far in excess of ten thousand installations. The success of the 1401 has often been attributed to the model 1403 chain printer that accompanied it; printing at 600 lines per minute, it enabled a single 1401 to replace four conventional tabulators.

The IBM 1401 was an instant success in the UK too, and in May 1960 ICT was forced to make a premature announcement of the 1301; this was partly for prestige reasons and partly to ensure that at least some sales of this class of computer accrued to ICT. The credibility of the 1301 announcement was tempered more than a little by the two to three year delivery times, and the modest specification: the basic 1301 was promised for mid 1962; the magnetic tape version would not be available until 1963; and there were no plans to announce a random access version at all. By contrast, delivery for the IBM 1401 was 6–12 months, and magnetic tape and random access storage were immediately available.

Thus during 1960–61, the whole tempo of development quickened. IBM's own prediction was that by 1964 40 per cent of its European revenues would be derived from electronic data processing machines[7]. ICT was ill-prepared for an escalation on this scale and found itself short of both competitive second generation computer designs and an electronic manufacturing capability. In the short term, the former could only be obtained from the United States, and the latter by buying out another British manufacturer.

The fact that ICT had to go to America for the most advanced computer technology was a manifestation of the 'technology gap' that was a major European political concern throughout the 1960s. Although British computers had been on a par with those in America in the early 1950s and there had been some notable recent successes (such as the Atlas), the fact was that in the mainstream of practical EDP computers, Britain had fallen lamentably behind[8].

American computers were considerably in advance of those available from European manufacturers in terms of the three components of a computer installation: processors, software and peripherals. By 1960, virtually all computers on the American market had second generation transistorized processors. Software – a term which came to use in 1959 or 1960 – was also considerably in advance of that available in the UK: the programming languages FORTRAN and COBOL were becoming widely available, and operating systems and real-time applications were far in advance of anything routinely offered by UK vendors. In terms of peripherals, the most significant difference was in magnetic tape and disc storage technology. The former was very well established with several suppliers, and the latter was maturing rapidly. By contrast, in Europe there was little capability in magnetic tape drive manufacture and none whatever in disc drive technology.

During 1960–61, ICT talked with most of the major American manufacturers. A deal was eventually struck with RCA in September 1961 which gave ICT a long term non-exclusive licence to use RCA computer technology; the overall cost of this agreement, which was related to ICT's turnover, was to be approximately £10 million over the 15 year term of the agreement. This was not an insignificant fraction of ICT's research and development budget.

Another arrangement made with RCA was to import its model 301 computers, which were resold as the ICT 1500; well over a hundred were eventually sold. Slightly later, in autumn 1962, a deal was made with Univac to import and resell their model 1004 calculating tabulator (Fig. 3). Nearly five hundred ICT 1004s were sold between 1963 and 1966, which was an important factor in enabling ICT to survive its financial crisis of the mid 1960s. The 1004 deal also enabled ICT to drop the 975 calculating tabulator development, allowing the research and development resources to be diverted to computer and peripheral development. Of course the substitution of imports for direct manufactures was bad for morale and also led to the shedding of thousands of the direct workforce, but there was really no alternative at the time.

A major strengthening of ICT's computer development capability was achieved in March 1961 by taking over the GEC staff who were developing the 1300 series at the GEC Coventry telecommunications works. The 1300 team was transferred to Stevenage and merged with ICT's own research and development staff as a subsidiary company, ICT (Engineering) Ltd, in which GEC took a 10 per cent share holding. At the same time, CDL, which had now served its purpose in specifying the 1300, was taken over and became the

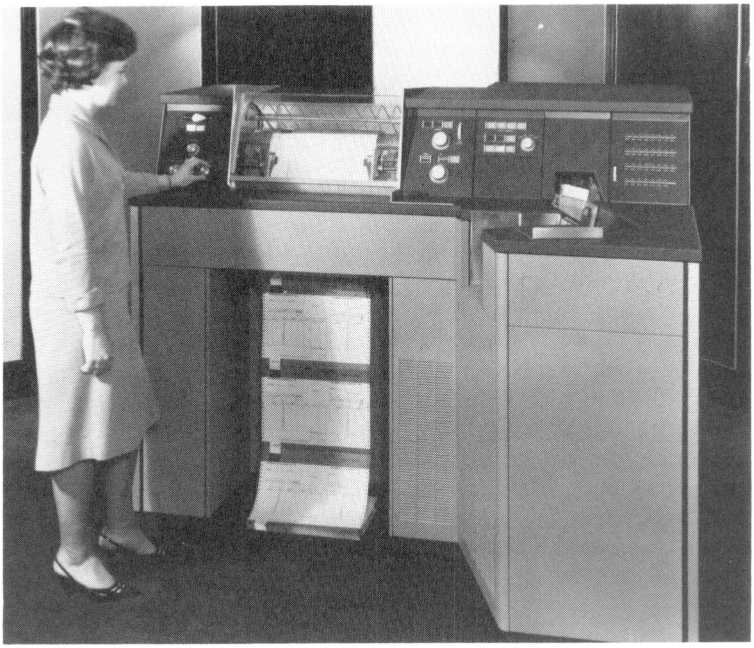


Fig. 3 ICT 1004 calculator, c. 1963

Fig. 4 ICT 1301 computer, 1962

nucleus of a new product planning group. The 1301 itself, however, was to be built in the GEC works, since ICT lacked the manufacturing capability; over 200 machines were eventually delivered, making it outstandingly the best selling second generation EDP computer in Britain (Fig. 4).

In order to acquire an electronics manufacturing capability, more or less serious top level talks were held between ICT and most of the British electronics companies. One possibility actively explored in 1961 was a merger with the English Electric computer division. This would have made good strategic sense, since their EDP marketing and technical strengths were complementary. English Electric had begun making computers in 1947, and during the 1950s and early 1960s it developed an excellent range of scientific machines, the most recent of which was the KDF9, which had an elegant stack-based architecture. Unfortunately the merger talks foundered, largely on the question of whether it should be ICT or English Electric that controlled the resulting company. Once the talks broke down in 1961, the impetus for a closer relationship was lost and not regained until 1967.

In fact, English Electric did manage to acquire the EDP and marketing expertise it was seeking by acquiring control of Leo Computers Limited in April 1963. The formation of English Electric Leo Computers Limited constituted a real second force in the British EDP computer industry. In 1964 the company absorbed the computer interests of the English Electric Marconi division, and became known as English Electric-Leo-Marconi.

4 The EMI and Ferranti acquisitions

In mid-1962 an opportunity occurred for ICT to acquire the computer interests of EMI. At the time, EMI was probably the only British electronics company actively seeking to exit from the computer business, since it recognized that it would call for vast capital investment, and even then commercial success was far from assured. This made the negotiations fairly straightforward.

EMI had some substantial attractions for ICT. It was a highly competent manufacturer; it had a team of about one hundred development staff, which would practically double ICT's capability; it had a small well-trained sales staff and some prestige customers, including the banks; and finally it had no less than three second generation computers at a time when ICT had none.

EMI's smallest machine, the model 1100, was broadly in the price/performance class of ICT's 1300 series, but it had the advantage of being available, with magnetic tape, at a time when the ICT machine was still in the prototype stage. Although ICT had arranged to buy in the RCA 301, the 1100 was slightly more powerful and in any case a home-built product was always to be preferred to an import. EMI's second machine was the 2400. This was a highly competitive large-scale EDP machine that promised to be a useful addition to ICT's product range. The EMI 2400 had been developed under the auspices of the National Research Development Corporation (NRDC), and was intended to be Britain's answer to the IBM 7090 large-scale EDP computer. This it would have been, had it not had profound reliability and software problems. This, of course, was unknown to either EMI or ICT at the time of the negotiations. In the event only three 2400s were ever sold[9]. EMI's third machine was the 3400. This machine had been developed with the aid of a £250 000 development contract from the NRDC and, along with the Ferranti Atlas, was intended to be Britain's answer to giant computer developments in America, the IBM STRETCH and the Univac LARC[10]. The giant machine market was one that it was ICT's policy to eschew, however, so the 3400 had little attraction, and it was never developed beyond the prototype that ICT inherited.

The EMI acquisition remained but a step in the right direction. A much more significant rationalization of the British computer industry came with ICT's acquisition of the Ferranti computer department. Talks were held on several occasions, but nothing materialized until early 1963 when Ferranti decided to make an exit from the computer business, for much the same reasons as EMI had decided to withdraw the previous year.

In fact, ICT's reaction to Ferranti's overtures were initially lukewarm. Although Ferranti's computer research and development and production were outstanding – unquestionably the best in Europe – its current range of products were not at all attractive to ICT. Ferranti's first generation machines were no longer being actively sold and promised to be a maintenance liability rather than an asset, and it only had two second generation machines, Orion and Atlas, neither of which was attractive to ICT.

The Orion, although it was a large-scale EDP machine, was too close in performance to the EMI 2400 which ICT was intending to sell. Moreover, Orion had major reliability problems owing to a novel electronic technology known as the 'neuron'[11]. The neuron eventually had to be abandoned and a successor, Orion II, developed.

The Atlas was even less of an attraction than the Orion. The Atlas, like the EMI 3400, was a very high-speed scientific computer intended to compete with the American STRETCH and LARC, and it had been partially underwritten by £300 000 funding from the NRDC[12]. Developed in collaboration with Manchester University, the Atlas was a major technical triumph doing much to recapture Britain's lost prestige in advanced computer design (Fig. 5). It had pioneered techniques in virtual storage and operating systems that were perhaps two years ahead of American manufacturers. But although the Atlas was strong on prestige, it was an asset that ICT did not want any more than it had wanted the EMI 3400. At the time of the merger talks, only two Atlases had been sold.

What the Ferranti acquisition therefore offered ICT was not a range of machines, but research and development potential and manufacturing

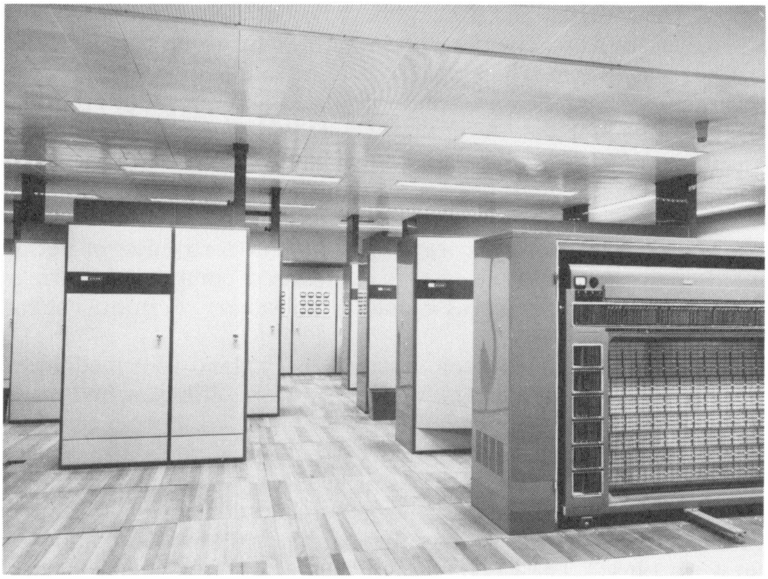


Fig. 5 The London University Atlas installation c. 1963 (Lower floor, showing main processing units)

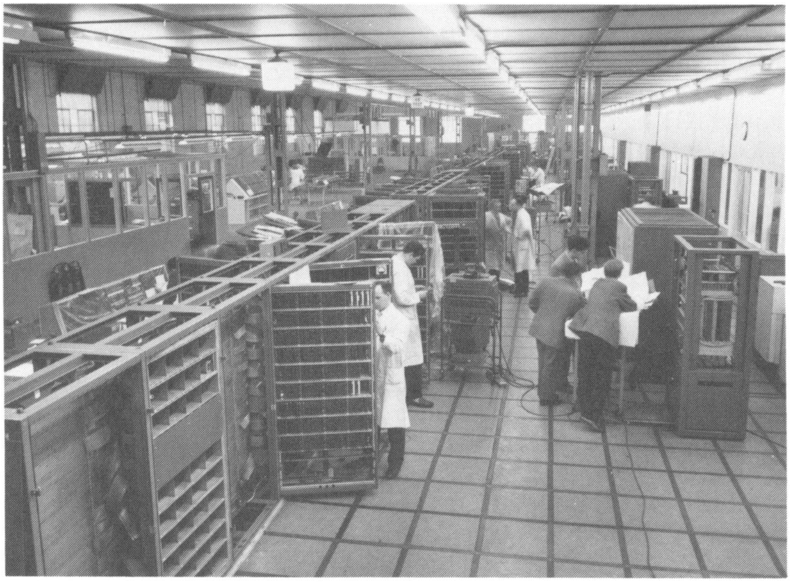


Fig. 6 Orion production line, c. 1963

capability. The research and development potential was certainly very high, particularly in software. The Ferranti programming group had developed the Atlas operating system which was well in advance of anything offered by American manufacturers, and it had produced a commercial programming language, Nebula, for the Orion that was at least equal to those offered by American manufacturers. As a computer manufacturer, Ferranti's computer plant in West Gorton, Manchester, was the largest in Europe (Fig. 6).

One particular event, however, transformed the attractiveness of Ferranti to ICT. This was the discovery of a medium-sized computer known as the FP6000 developed by Ferranti's Canadian subsidiary, Ferranti-Packard.

The FP6000 had originally been specified in England as a medium-priced EDP computer by one of Ferranti's salesmen, Harry Johnson, and the design had much common ancestry with the Ferranti Pegasus. But because Ferranti had most of its resources committed to the Orion and Atlas, it was not developed in England. The design was picked up by the Canadian subsidiary, however, where it was developed into a product during 1962. The FP6000 was first described publicly at the American Spring Joint Computer Conference in April 1963[13]. The FP6000 was a very advanced machine, incorporating a multi-programming system, derived from Orion, of unusual sophistication in a machine of its size. Implicit in the design of the FP6000 was that it could support a spectrum of machine sizes, priced from as little as £50 000 up to perhaps £500 000. The performance could be extended in a fairly continuous manner by having core store sizes from 4 Kwords up to 32 Kwords, with speeds from 6 microseconds to 2 microseconds, and various processor options. The software for the system would be invariant with system size, and this would enable users to upgrade their systems without the reprogramming costs of changing to an entirely different architecture.

Although ICT had its own computer projects underway, it was clear that the FP6000 was far more developed than any of them. As a result, it was decided to acquire the Ferranti EDP computer interests. The deal included about 1900 personnel, and the development and manufacturing facilities at West Gorton, Manchester, and Bracknell, Berkshire (the Ferranti Digital Systems Department, which manufactured process-control computers, was not included).

Following the Ferranti acquisition, ICT had an impressive assortment of computers. In the short run, there was little to be done but accept the position as it was. There were too many competing machines in the small/medium range; the medium priced machines were not particularly competitive; of the large machines, the EMI 2400 had been abandoned, and the Orion was unreliable; and such enthusiasm as existed for the Atlas came only from Ferranti. Another problem was that the software and hardware compatibility between them was negligible.

For the longer term, it was intended that all of ICT's medium to large EDP computers would be made from a single 'project set' which would have compatible software and peripherals throughout the range. It was planned that such a range of computers should be available by 1968, and that the range would either be developed in collaboration with RCA, or by development of the FP6000.

Which way to go – the RCA route or via the FP6000 – was still under active investigation when IBM astounded the computer industry by announcing System/360.

5 The response to System/360

The IBM System/360, announced on 7 April 1964, was a compatible family of third generation computers. The range consisted of six distinct processors and forty peripherals, which were intended to replace all of IBM's current computers, except the smallest and largest. The scale of the announcement was entirely unprecedented, and all the evidence is that it took the rest of the industry largely by surprise.

IBM's decision to launch System/360 is generally regarded as one of the great business success stories of the second half of the twentieth century. Certainly, within the IBM culture it has taken on a symbolic significance as the true beginning of the computer age. The publicity surrounding System/360, however, has lent IBM an aura as a technological pioneer which was scarcely justified. The fact is that the concept of a compatible family was well understood within the computer industry; IBM's achievement was to be the first in the market place with such a range. Similarly, although the term 'third generation' was first used in connection with System/360, the IBM machines in fact used hybrid circuits known as Solid Logic Technology (SLT), which might more fairly be described as 2½ generation.

The rationale for System/360 (and indeed the compatible ranges of all the other manufacturers) was to tackle the problems associated with marketing many different computer designs. When System/360 was conceived in 1961, IBM had no less than seven incompatible computer architectures. This proliferation of computer models was causing major problems in marketing, manufacturing and software development[14].

In terms of marketing, the large number of architectures necessitated multiple selling forces, each expert in a particular machine. Users also had problems upgrading their installations by more than a factor of about two without changing to an entirely different range of machines, with all the attendant problems of reprogramming. Manufacturing problems were acute, and were threatening to undermine IBM's traditional economies of scale. For example, some 2500 different circuit modules were used in the different processors. Peripherals also represented a problem since it was necessary to develop a special purpose controller to interface each peripheral with each processor: given m processors, and n peripherals, this represented a possible $m \times n$ controllers; the only way to contain the problem was to artificially limit the number of peripherals offered with each processor. The same combinatorial explosion was occurring in software, where each processor required a full portfolio of systems programs and applications. In practice, the programming effort was kept within bounds, but software was beginning to dominate development costs and it was realized that the problem would have to be tackled sooner rather than later.

It should be emphasized that these problems were common throughout the industry. ICT, for example, had almost as many incompatible architectures as IBM and was responding to the problem in its own way. By 1962, for example, it had already done important pioneering work in collaboration with the National Physical Laboratory in developing a 'standard interface', so that any peripheral could be attached to any processor[15]. ICT was also labouring under no less than 24 different magnetic tape formats on its various computers, so that it was acutely aware of the need for the standardization of file and data formats.

It was against this background that in November 1961 IBM's top management set up the SPREAD Task Group (Systems Programming Research and Development) 'to establish overall plans for data processing products'[16]. The SPREAD Task Group included several of IBM's most senior managers and technical staff, from various divisions, both domestic and world trade. The group worked quickly, and its recommendations were made in their final report, dated 28 December 1961.

The SPREAD Report concluded that a compatible family of computers was both commercially desirable and technically feasible, and recommended the development of a family of computers, to serve a spectrum of users from small to large, both scientific and commercial. The different processors would have identical instruction codes, but they would be constructed using different technologies and with core memories ranging from slow to the fastest available. By means of a standard interface, most processors would support most peripherals. It was recognized, however, that it would not be possible to accommodate very small and very large machines within the technological framework of the computer range, and this remained a serious competitive disadvantage.

In January 1962, the IBM board accepted the recommendations of the SPREAD Report, and the research and development and manufacturing programs within IBM were reorientated to what was to be called System/360. It was estimated that the research and development budget for System/360 was about $500 million, and that the total cost of development and manufacturing was $5 billion[17].

System/360 produced two major competitive challenges to other manufacturers: first, the concept of a compatible range, and second, a several-fold increase in price/performance over existing computers. In the hiatus of ordering that followed the April 1964 announcement, while the market digested the implications of System/360, all of IBM's competitors formulated their responses. There were three broad strategic responses available:

1 to develop a range of computers fully hardware compatible with System/360, but with a better price/performance and/or technical superiority;
2 to develop a range of computers, not compatible with System/360, but with a better price/performance and/or technical superiority;
3 to focus on 'niche' areas not well served by System/360.

The best known examples of the first strategy, developing a 360 compatible range, were the RCA Spectra 70 and the English Electric System 4 (see later). Several manufacturers adopted the second strategy of launching a non-compatible computer range: these included the ICT 1900 series, the Honeywell 200 series and the Burroughs 500 series; in each case these ranges were developed by extending an existing model upwards and downwards. The most successful exponent of the third computer response, aiming for a niche market, was CDC. By attacking IBM where they were weakest, CDC rapidly came to dominate the large machine market.

This, then, was the competitive environment created by the System/360 announcement. As computer manufacturers in the international market place, the British companies were forced into adopting one or more of these competitive responses.

6 The ICT 1900 series

From about mid-1963, well before the 360 announcement, both RCA and ICT had been independently evolving plans for compatible ranges of computers. While ICT had been contemplating a range based on the FP6000, RCA had quite separate plans that would include some form of IBM compatibility. These were, however, long term plans; in the case of ICT there was certainly no intention of delivering a compatible range much before 1968. The effect of the 360 announcement in April 1964 was therefore to compress into months development programs that had been intended to take years.

In spring 1964, RCA invited ICT to make an appraisal of its long term computer plans in the hope that they would decide to adopt the RCA range (as yet unspecified), and of course assume some of the research and development and manufacturing load. As luck would have it, the System/360 announcement of 7 April 1964 occurred at the very moment of the ICT visit. There was no industrial espionage, and RCA obtained details of the 360 from the publicly available manuals. While the ICT team toured the United States on other business, RCA immediately investigated the implications of System/360 for the RCA range. When the ICT team returned a week later, RCA had decided to make the new line fully 360 compatible. The new RCA range was subsequently announced as Spectra 70.

ICT was entitled to manufacture the RCA series under licence, but declined to do so on two main grounds. First, on the policy of IBM compatibility, and second on the question of lead-times.

IBM compatibility was seen to be a poor competitive strategy for ICT. The only logical argument for a user buying an IBM compatible computer in preference to a machine manufactured by IBM, it was argued, was because it had a better price/performance, or technical superiority. Given IBM's relative economies of scale, a price advantage could only be achieved at the cost of very low profits, if it could be achieved at all. And given IBM's research and development resources, and the fact that an IBM compatible manufacturer would have to follow IBM developments since it could scarcely anticipate them, technical superiority would also be extremely difficult to achieve. There was also the question of the damage that copying IBM would do to ICT's (or rather Ferranti's) image as an innovator.

The question of lead-times was at least as decisive. Although the RCA planners believed they could bring machines onto the market in 18–24 months, the ICT team was highly sceptical. And in any case, they would be left without any product at all during the development period. RCA was a rich company and willing to withstand a short term loss for the eventual high rewards, but ICT did not have this luxury.

The FP6000 was less architecturally advanced than System/360, but it had the great advantage of being available, working and already in the field. The ICT team was convinced that it could be developed into a compatible family and be delivered ahead of System/360. Probably the main perceived disadvantage of the FP6000 – and this was to be brought up time and again over the next decade – was that it used a 6-bit character, where System/360 used an 8-bit byte. But even this, in 1963, was seen as far from being a disadvantage, since it meant that the core storage requirement – which accounted for perhaps 25 per cent of processor costs – would be six eighths that of a byte-organized machine. The ICT range needed to have a 10 to 15 per cent price/performance advantage over System/360, if it was to sell at all, and this went no small way to achieving it. Similarly, although the FP6000 was in electronic terms a second generation discrete component machine, the technology was very well established and therefore cheap.

Before flying back to England, the ICT team had decided to recommend that the FP6000 be developed into what was to become the 1900 series. The recommendation was accepted by top management, and in a matter of days all of the major processor and peripheral projects were redirected to the fulfilment of the 1900 series. The original FP6000 – already announced as the ICT 1900 – now became the 1904, the middle of the range. The two main Stevenage processor developments were reorientated to the 1902/03 (there were two models but only one processor), and the 1901, the smallest member of the 1900 series. At West Gorton, projects were established for the large 1906 processor, and for scientific variants of the 1904 and the 1906 (known as the 1905 and 1907). During the following months, private presentations of the 1900 series were made to potential customers and the decision was made for a public announcement in the autumn.

The press launch for the 1900 series took place on 29 September 1964. It had been realized that something special would be called for in order to match the very high profile launch of System/360. A bright new PR firm, which had spun-off from the 'Tonight' television team, was brought in to organize the presentation, with a script by Tony Jay (now Sir Anthony Jay, of 'Yes, Minister' fame). The presentation, which was given simultaneously around the world, was a tremendous success.

Several models of the 1900 series were announced at the press launch, priced from £40 000 up to £750 000 (Table 1 and Fig. 7). The small 1901 was not announced in order to protect the 1004 and to minimize the development load, but the numbering of the range was designed to imply that such a low cost model would eventually become available.

Table 1 ICL 1900 series announcements, 1964–65

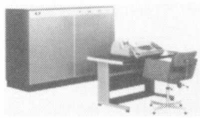
Model	Price	Announced	Delivered
1901	65	Sep 1965	Oct 1966
1902	105	Sep 1964	Sep 1965
1903	175	Sep 1964	Aug 1965
1904	260	Sep 1964	May 1965
1905		Sep 1964	Jan 1965
1906	700	Sep 1964	mid 1967
1907		Sep 1964	mid 1967
1909		Sep 1964	Oct 1965

Notes
Prices: average system price in £000s.
Deliveries: approximate date first delivered to a customer according to *Computer Survey* and ICL sources; some own use machines may have been delivered earlier.
Models 1905, 1907 and 1909 were scientific versions of the 1904, 1906 and 1903, respectively, equipped with a floating point processor.

In addition to the processors, a total of 27 different peripherals were announced. All the basic punched card peripherals and printers were ICT's own make, including a very competitive 1350 lines per minute printer that went on to sell on an OEM basis. All the fast magnetic tapes and discs, however, were to be imported from America – a fact which was not advertised at the launch. A full range of software was announced, including a multi-programming executive, programming languages and application packages.

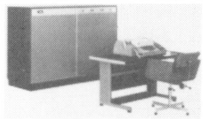
The keynote of the press launch, however, was to emphasize the availability of the 1900 series – that 'it's here, you can see it' – and deliveries of under one year were promised[18]. At the Business Equipment Exhibition the following week, at Olympia, two prototype models – the 1902 and the 1904 – were

1900 SERIES CENTRAL PROCESSORS



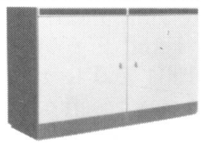
1902

16,384 - 65,536 characters
4,096 - 16,384 words
6 microsecond core speed
8 input/output channels



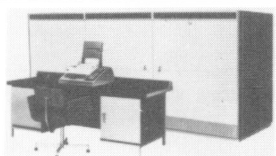
1903

32,768 - 131,072 characters
8,192 - 32,768 words
2 microsecond core speed
8 input/output channels



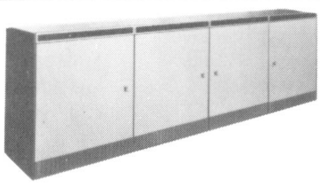
1904

32,768 - 131,072 characters
8,192 - 32,768 words
2 microsecond core speed
23 input/output channels
Multi-Programming



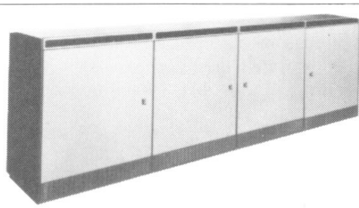
1905

32,768 - 131,072 characters
8,192 - 32,768 words
2 microsecond core speed
23 input/output channels
Multi-Programming
Autonomous Floating-Point Hardware



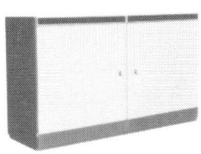
1906

131,072 - 1,048,576 characters
32,768 - 262,144 words
1.1 - 1.25 microsecond core speed
Unlimited input/output channels
Multi-Programming
Floating-Point Hardware



1907

131,072 - 1,048,576 characters
32,768 - 262,144 words
1.1 - 1.25 microsecond core speed
Unlimited input/output channels
Multi-Programming
Autonomous Floating-Point Hardware



1909 PACKAGED SCIENTIFIC SYSTEM

65,536 - 131,072 characters
16,384 - 32,768 words
6 microsecond core speed
8 input/output channels
Multi-Programming
Autonomous Floating-Point Hardware

Fig. 7 Publicity slide for the ICT 1900 series, 1964 demonstrated; this was unusual for any product announcement, but unique in the history of ICT. The timely delivery of computers had never been one of ICT's strong points, so the demonstrations went a long way to restoring its credibility.

The impact of the 1900 series launch exceeded all expectations and orders poured in, both from Britain and around the world. Morale in ICT soared. The 1900 series, once thought of as a stop-gap, now looked to be a major success.

7 ICT 1900 series development

The 1900 series processors proved to be the least troublesome aspect of design and production. The mid-range machines were quickly derived from the FP6000, and rushed into production. The first production 1905 was delivered to Northampton College, London, in January 1965 – only four months after the 1900 series announcement – and was officially inaugurated by Lord Bowden on 15 March 1965. The short lead-time of the 1900 series proved to be a major competitive advantage over System/360. Although the first machines from the IBM range were delivered in the United States in spring 1965, there were production problems that held back deliveries in the UK until spring 1966.

Since one of the main competitive advantages of the 1900 series over System/360 was its lower price, ICT placed a major emphasis on investment in automated manufacturing and the use of standard modules to reduce manufacturing costs, rather than technical innovation in circuitry. Several Milwaukee-matic numerically controlled milling machines were bought from America at enormous expense, and 'fine-blanking' metal shaping machines installed (Fig. 8). The ICT plants became something of a show-piece of industrial automation, attracting ministerial attention[19], although the cost effectiveness of some of the investment was debatable. Capacity of the West Gorton computer plant was ramped up to about 300 processors per year, and investment in automation reduced manufacturing costs to about one fifth of those of the Orion.

Major problems were encountered with peripherals, but mainly those imported from the United States. ICT's punched card peripherals and printers were already well developed before the 1900 series program was begun, and were selling well on an OEM basis to other manufacturers. Because no random access memory devices were available from the UK or European manufacturers, ICT had arranged to import discs from Anelex in the United States, and to use the RCA RACE magnetic card file, both of which were still under development at the time of the 1900 series announcement. The Anelex disc in the event never materialized, and discs were eventually bought from CDC – but not until mid-1966. This lost ICT a lot of ground in the growing transaction processing market, in which IBM and Univac had been strong since 1960; ICT had yet to put a computer with a

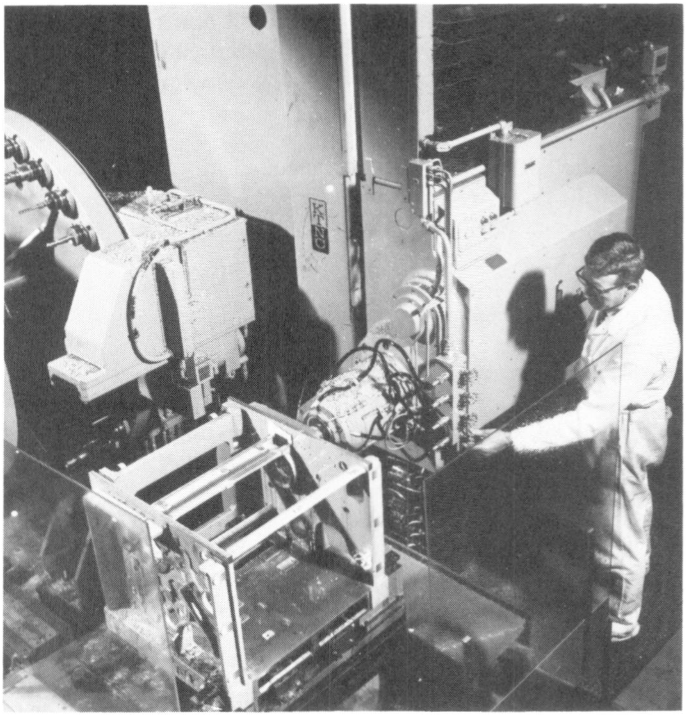


Fig. 8 Publicity shot of a Milwaukee-matic drilling a printer frame, c. 1965

disc store onto the market. The RCA RACE also had problems – it was very late arriving and never really worked well, particularly in competition with the IBM Datacell.

The biggest development problems, however, were encountered with software. A new Programming Division was established in January 1965, but ICT, like every manufacturer, was unprepared for the escalation in user demand for software on third generation computer systems, and the lack of reliable tools for estimating costs and production time-scales. Another major problem was the shortage of programmers in Britain. An urgent advertising campaign was begun in early 1965 to attract one hundred additional programmers, but it took the full year to get that number of recruits; in the meantime several of the applications packages promised for the 1900 series were shelved, or deliveries lengthened up to two years. The ICT Programming Division built up to about 600 people by 1966, built largely on the expertise of Ferranti which had world-class skills in compiler writing and operating systems that ICT had lacked before the merger.

By January 1965, ICT had taken 124 orders for the 1900 series, a small figure in world terms, but unprecedented for a British computer. This success, inevitably, caused the order book for the older models to collapse, and ICT was soon caught up in a major financial crisis that threatened the 1900 series research and development.

It was becoming clear that ICT's £4 million annual research and development budget was insufficient to maintain the momentum of the 1900 series, especially to keep it competitive with System/360. During 1965, IBM had announced major operating systems for System/360 to which ICT had to respond for the 1900 series. This spawned projects for the GEORGE 1 and 2 operating systems in 1965, and for GEORGE 3 in 1966. Software costs escalated to £1·7 million during 1965, which was about 50 per cent more than budgeted. In November 1964, IBM announced a low-cost addition to the 360 range, the model 20. To respond to this announcement ICT was forced to bring forward the 1901 development, which was announced in September 1965. The projected research and development costs for the next two or three years were now about £5·5 million a year. This still represented a little under 10 per cent of revenues, but ICT's profits were so low that the additional expenditure would have meant ICT operating at a loss. Harold Wilson's Labour Government was, however, well disposed towards ICT, and the company was able to secure a £5 million development loan from the NRDC in May 1965. It is doubtful whether the 1900 series development could have continued without the NRDC loan.

It is interesting to compare the research and development cost of the 1900 series of about £20 million over 4 years (approximately $56 million) with the $500 million IBM expended on System/360 research and development over a comparable period. It has often been observed that IBM's research and development expenditure is greater than ICL's total turnover, and this has led at least one observer to note that 'the efficiency of their research and development process is therefore of commendable quality'[20]. This is certainly true, but it is not the ten times greater efficiency that the research and development spending ratios might imply. One major difference for example, was that IBM was a far more integrated manufacturer, and made all its own peripherals and most of its electronic components. By contrast, ICT bought in virtually all of its magnetic storage peripherals, semiconductors and core memories, so that their development costs did not form a component of ICT's overall research and development budget. Again, IBM put far more resources into applications development than did ICT, which simply did not compete with IBM in the more esoteric applications. On the other hand, there is plenty of evidence that ICT was more cost conscious and design minded. Probably the most well known example of IBM's profligacy was its operating system OS/360 which was said to have taken 5000 programmer-years of effort, and to have had over 1000 programmers working on it at its peak[6]. The ICT GEORGE 3 operating system involved no more than about one hundred programmer-years of effort, and was generally considered to be a much better system. In 1968, ICT received the Queen's Award for the development of 1900 series software.

8 The English Electric System 4

Soon after the merger between the English Electric computer division and Leo Computers had taken place in April 1963, planning activity began on a range of third generation machines.

There were essentially three options available: to base the new range on the KDF9, to base it on LEO III, or to develop a completely new architecture. In early 1964, because the KDF9 and LEO III both had technical limitations that prevented their easy enhancement, it was decided to develop an entirely new range, which was known internally as Project KLX.

During the following months, the KLX architecture was developed in detail, software was specified and some engineering prototyping done. An important aspect of the latter was the decision to make use of Marconi integrated circuits rather than discrete transistors. Marconi, a division of English Electric, had made a strategic move into integrated circuits in 1962, and during 1963 had produced the Myriad process control computer; the Myriad was probably the first computer in the world to use integrated circuits.

With the announcement of System/360 and the 1900 series during the course of 1964, the pace and scale of innovation was increased, and it was clear within English Electric that there was a need to contain development costs within realistic bounds. As it happened, English Electric had a longstanding technology sharing arrangement with RCA, and it was decided, therefore, to investigate whether it might not be possible to integrate KLX with the Spectra 70.

An English Electric study team visited RCA for a three week period during November/December 1964. RCA, in fact, made the Spectra 70 announcement during their visit, on 8 December 1964[21]. The Spectra 70 range, as then announced, consisted of four machines. The two smaller models were to use conventional discrete components, and were offered with a twelve month delivery, and the larger models, which were to use integrated circuits, were offered with an eighteen month delivery. Although RCA adopted the System/360 instruction code so far as users were concerned, the 'hidden' operations for multi-programming and time-sharing were considerably more elegant and advanced.

It was clear to English Electric that by adopting Spectra 70 it would greatly accelerate the introduction of its new range, and would greatly reduce the development costs, especially of software. There were, however, two considerations that made the decision not entirely straightforward: the first was the question of IBM compatibility, and the second was the apparent loss of technical leadership.

English Electric did not regard IBM compatibility as being a particular advantage or disadvantage in the UK market. On the one hand, because

IBM did not dominate the UK market to anything like the extent it dominated the American market, the attractions of adopting IBM standards were minimal. On the other hand, there was no doubt in English Electric that they would be able to achieve both technical superiority and a price/performance advantage over System/360. (Of course, when it came to actually *marketing* System 4, the advantages of IBM compatibility were trumpeted for all they were worth.)

Both English Electric and Leo Computers had enjoyed a strong technical position in the British computer industry. The KDF9 was regarded as a highly innovative computer and sold well in universities, and the LEO III also had great prestige. In order to secure Government orders, English Electric had to be seen as something more than a manufacturer of computers under licence. This, in short, was the not-invented-here syndrome.

In fact, the Spectra 70 range offered ample scope for Anglicization. First, English Electric would be able to develop a big machine which the RCA range lacked; this would be sold to large university and commercial customers. Second, by using integrated circuits for the small machines, where RCA had used discrete components, English Electric would be perceived as being highly innovative by offering a 'true' third generation range in hardware terms, unlike System/360 or the 1900 series. Only the middle range machine would be directly derived from RCA.

The decision was therefore taken to adopt the RCA Spectra 70 architecture for the English Electric range, which was to be marketed as System 4. The full range of four machines was announced in September 1965 (Table 2), with deliveries promised for early 1967. The model 4-50 was derived directly from the Spectra 70/45; it was therefore the first to be delivered, although deliveries in quantity were much later than originally had been planned. The two small machines, the 4-10 and 4-30, were developed by the Marconi division of English Electric. Although the functional specifications from RCA were used for these machines, almost everything else was 'invented here'. The use of integrated circuits, however, meant that the machines were not particularly economic. The 4-10, in particular, was not price competitive

Table 2 English Electric System 4 announcements, 1965

Model	Price	Announced	Delivered
10	100	Sep 1965	Cancelled
30	172	Sep 1965	Jun 1967
50	271	Sep 1965	May 1967
70	600	Sep 1965	mid 1968

Notes
Prices: average system price in £000s.
Deliveries: approximate date first delivered to a customer according to *Computer Survey* and ICL sources; some own use machines may have been delivered earlier.
Models 40, 45, 60, 65, 72 and 85 were added to the range during 1966–70.

against the IBM 360/20 or the ICT 1901, and was eventually cancelled before any were delivered.

The large 4-70, however, was a considerable technical triumph. Unlike System/360, it had been designed in the expectation that the future growth in large-scale computing would be in real-time transaction processing and multi-access systems. During 1966–67 many of the prestige orders for large systems from the British Government and nationalized industries went to the 4-70: these included several machines for the Post Office, the National GIRO Bank, the electricity boards, the UK Atomic Energy Authority and universities (Fig. 9). The 4-70 completely outclassed the ICT 1906, which had to be withdrawn from the market until a faster model (the 1906A) could be announced in 1967.

Fig. 9 ICL System 4-70 at the National Giro Centre, c. 1968

The reaction within ICT to System 4 was initially fairly sceptical. The idea of a British company, with only 15 per cent of the home market and no real export market, competing with IBM head on was considered ill-advised, to put it at its best. English Electric's delivery schedules were also considered over optimistic, since it was well known in the industry that System/360 deliveries were running three or four months late due to manufacturing problems with its SLT circuits; in the event volume deliveries of System 4 did not begin until well into 1968. Although integrated circuits were attractive from a marketing viewpoint, ICT took the view, like IBM, that technology was secondary to reliability and prompt delivery. This was evidently also the view of the market: in the year following the September 1965 System 4 announcement, just £3 million worth of orders were taken;

this was equalled by the orders for the ICT 1901 alone, which was announced the same month.

Nonetheless, a major effect of System 4 was to force ICT into introducing integrated circuits into the 1900 series as rapidly as possible, particularly to restore the competitiveness of the 1906. One interpretation of this situation was that ICT and English Electric were not so much competing against the Americans, as nibbling at each other's share of the British market.

9 ICL and the feasibility of a new range

The decisions of ICT and English Electric to independently embark on their own third generation computer ranges took place against a backdrop of growing political concern at the increasing dominance of the high technology industries by American multi-national companies.

When, in October 1964, Harold Wilson's Labour Government came into power, one of its first acts was to establish a Ministry of Technology, envisaged in *The New Britain* an organization to 'guide and stimulate a major national effort to bring advanced technology and new processes into British industry'. Wilson placed the British computer industry at the very top of Mintech's agenda:

> My frequent meetings with leading scientists, technologists and industrialists in the last two or three years of Opposition had convinced me that, if action was not taken quickly, the British computer industry would rapidly cease to exist, facing, as was the case in other European countries, the most formidable competition from the American giants. When, on the evening we took office, I asked Frank Cousins to become the first Minister of Technology, I told him that he had, in my view, about a month to save the British computer industry and that this must be his first priority[22].

Accordingly, in November 1964, the newly appointed Minister of Technology held talks with both ICT and English Electric, in what was to be the first of many attempts to persuade them to bring together their computer interests. These talks came to nothing, and during 1965, Mintech therefore put the rationalization of the computer industry to one side in favour of piecemeal initiatives. These initiatives included: the establishment of a Computer Advisory Unit to encourage and advise on the use of computers in the public sector; the formation of the Flowers Committee to report on the use of computers for research; and the creation of the National Computer Centre.

Another Mintech initiative was to stimulate technological innovation in computers by sponsorship of research and development. The NRDC was considerably expanded, one of its first actions being to grant the £5 million loan for ICT's 1900 series in May 1965. Basic and pre-competitive research were stimulated by the Advanced Computer Technology Project (ACTP) which funded advanced computer research on a cost-shared basis with industry. ICT had a number of programs under this initiative, the most important of which was J.K. Iliffe's BLM project, for an advanced computer architecture. The 'timely support' of the ACTP funding helped sustain the BLM project at a critical time[23], and it was later to be a major input to the ICL 2900 series.

A fifth initiative, shrouded in some secrecy at the time, was to invite ICT to lead a European consortium in making a proposal for a very large computer. During 1963–64 the Atlas, then Europe's largest computer, had been eclipsed by several American high-speed computer projects, and there was a wish to restore Britain's position in this most prestigious area of computer development. The initial proposal was for a machine of a staggering 100 Atlas power, and the development cost was put at about £10 million. In the event, the Government's advisors recommended a much more modest program, the upshot of which was that ICT obtained limited Mintech support for a computer of about 10 Atlas power. This computer – to be known as the 1908, but known internally as Project 51 – in fact never materialized, but perhaps succeeded in generating more column inches of press speculation than any other computer project of the 1960s.

On 31 March 1966, Wilson's Labour Government was re-elected with a safe 97 seat majority, ever more determined to revitalize Britain's industrial base. The two most important instruments for this revitalization were Mintech, whose scope was greatly expanded, and the Industrial Reorganization Corporation (IRC). The efforts of the IRC were initially focused on the electrical and electronics industries. During 1966–68, it encouraged the acquisition of AEI by GEC, the merger between English Electric and Elliott-Automation, the formation of ICL by the merger of ICT and English Electric Computers, and finally the merger between GEC and English Electric.

By spring 1966, the condition of the three main British computer companies – ICT, English Electric-Leo-Marconi and Elliott-Automation – had changed considerably. ICT's position, following the success of the 1900 series and stronger management, had markedly improved to the point where it was making profits and was easily the strongest of the three companies. The position of English Electric-Leo-Marconi, following the launch of System 4, had worsened considerably; the development costs of the new range had produced the anticipated heavy losses during 1966, and the delivery of the new range in early 1967, on which success depended, was problematical. The third company, Elliott-Automation, was in still deeper trouble. A sizeable proportion of Elliott-Automation's business had been defence contracts associated with the TSR-2 aircraft program which had been cancelled in April 1965, and the company was actively seeking some form of merger.

With the changing fortunes of ICT and English Electric-Leo-Marconi, the possibility of a merger surfaced again within Mintech. In order to gain a clearer picture, Mintech commissioned an independent enquiry into the affairs of the two companies. The enquiry was led by S. John Pears, a senior partner of Cooper Brothers accountants, and his report appeared in September 1966[24]. The Pears Report came down firmly against the idea of a merger between ICT and English Electric-Leo-Marconi. The main reason for this recommendation was the incompatibility of the 1900 series and System 4, and the fact that both developments had passed the point of no return.

In July 1966, however, Anthony Wedgwood Benn replaced Frank Cousins as Minister of Technology. Wedgwood Benn was determined to see a large scale rationalization of the British computer industry and referred the whole problem to the IRC. The IRC effectively dealt with the problem in two stages: first the rationalization of the process control computer industry, and second the EDP computer industry. The former proved relatively straightforward and English Electric absorbed Elliott-Automation in June 1967. The new English Electric subsidiary was named English Electric Computers Limited.

It now remained to rationalize the EDP sector of the industry. In April 1967, Wedgwood Benn and his technical advisors called a meeting with the top management of ICT and English Electric to persuade them to merge their EDP computer interests. Mintech accepted the conclusions of the Pears Report that the main impediment to a merger was the incompatibility of the current ranges, and therefore offered *inter alia* a non-repayable grant in the region of £25 million towards the development of a new range of computers for delivery in the early 1970s.

Before a merger could proceed, however, there was a major technological uncertainty. Would a new range be possible? Would it be possible to design a new range of computers that was not only competitive with System/360, but was also compatible with both the 1900 series and System 4 in order to lock in existing customers? The argument ran that if compatibility with the existing ranges was not achieved, then customers replacing their machines would inevitably look at all available machines on the market and would be prey to the American manufacturers. It was not at all obvious, in 1967, that compatibility with the two very disparate architectures could be achieved without affecting technical performance, so that a joint ICT/English Electric working party was established for a feasibility study.

The working party met in secret for an intense three day session in the Hotel Cavendish, London, 3–5 July 1967. The working party was not, in fact, able to give a categorical assurance on the compatibility criterion for the new range, but was guardedly optimistic:

> We are agreed that there is no prima facie reason why it should not be possible to plan a range of systems meeting the basic requirements of competitiveness and of acceptable compatibility with the current ranges of both companies.

> The overall success of such a planning operation cannot be predicted, in that sacrifices in competitiveness directly attributable to securing satisfactory transfer from both current ranges cannot yet be estimated. However, there are good reasons (e.g. the general rapid advance of technology, the successful current activities in emulation, the increasing use of high level languages) to believe that the chances of an acceptable outcome are good[25].

The working party had in fact gone as far as it could in determining the feasibility of a new range. To proceed further it would be necessary to involve many more people from the technical and the planning sides of each company. Since secrecy would then be impossible, future progress implied some form of announcement of the new range studies, with the obvious implication of an eventual merger. This was a step that neither company was yet quite ready to take, and the top level negotiations continued their leisurely pace.

On 24 August 1967, however, the calm of ICT and English Electric's deliberations was shattered when ICT received a communication from the Plessey Company. Plessey had become convinced that the future lay in computer networks linking real-time mainframe computers and insisted on taking a major share in any new British computer company.

Although Plessey's intervention threw Mintech's plans into disarray, these events did not displease the Treasury. Now that cash-rich Plessey had joined the merger talks, a Government subvention of the order of £25–30 million was seen as politically unacceptable, and the Treasury was now thinking in terms of about a half of that amount.

A compromise deal was finally hammered out between ICT, English Electric, Plessey and Mintech, by which they would each take a share of the equity of a new company, International Computers (Holdings) Limited. ICT's existing shareholders received 53·5 per cent of the equity in the new company; Plessey and English Electric each took 18 per cent of the equity; the remaining 10·5 per cent of the equity was taken up by Mintech. Altogether, the Government put £17 million into the company: £13·5 million as a non-repayable grant to develop the new range and the large computer project, and £3·5 million in shares. A White Paper on the computer merger was presented to the House of Commons on 28 March 1968, and ICL was vested on 9 July 1968[26].

The design of the new range – later known as the 2900 series – began in earnest in early 1969. Alas, the research and development funds provided by the Government, and from ICT's own resources, were to prove entirely inadequate for the task, and eventually a total of £40 million was provided from Government funds. The development of the new range will be described in a future paper.

References

This article is based on interviews with past and present ICL personnel, documentary records in the ICL Archives (Putney) and the ICL Historical Collection (Stevenage), and on the open literature. The references below are intended to be representative of the major documentary sources used in this paper, rather than an exhaustive listing. Full references will be given in the author's history of ICL, currently in preparation.

1 CAMPBELL-KELLY, M.: 'ICL Company Research and Development', Part 1: 1907–1955 ICL Technical Journal 5 (May 1986) 2–17.

2 FREEMAN, C., HARLOW, C.J.E., FULLER, J.K. and CURNOW, R.C.: 'Research and Development in Electronic Capital Goods', National Institute Economic Review 34 (November 1965) 40–91.

3 UK market shares are given in the appendices of: Select Committee on Science and Technology (Sub-Committee D), Session 1969–70, Minutes of Evidence (HMSO 1970).

4 'IBM Computers – the story of their development', Data Processing 2 (April–June 1960) 90–101.

5 STEVENS, L.D.: 'The Evolution of Magnetic Storage', IBM Systems Journal 25 (September 1981) 663–75.

6 FISHER, F.M., McKIE, J.W. and MANCKE, R.B.: IBM and the U.S. Data Processing Industry, 1983.

7 SHEEHAN, R.: 'Q. What Grows Faster Than I.B.M.? A. I.B.M. Abroad', Fortune (November 1960) 166–170, 236, 241–242, 243–244.

8 OECD: Gaps in Technology: Electronic Computers, 1969.

9 CROOK, K.: 'In the beginning ... EMIDEC 2400s are still giving sterling service', Computer Weekly (1 May 1969).

10 HENDRY, J.: NRDC and the Early British Computer Industry, unpublished ms, 1987.

11 SWAN, B.B.: The Ferranti Computer Department, unpublished ms, Ferranti Archives. 1975.

12 HENDRY, J.: 'Prolonged negotiations: the British fast computer project and the early history of the British computer industry', Business History 26 (1984) 280–306.

13 MARCOTTY, M.J., LONGSTAFF, F.M. and WILLIAMS, A.M.: 'Time Sharing on the FP6000 Computer System', Proc AFIPS SJCC (April 1963).

14 EVANS, B.O.: 'System/360: A Retrospective View', Annals of the History of Computing 8 (1986) 155–179.

15 WESTWATER, F.L.: 'The Interface Concept', Data Processing J. (November–December 1964) 326–327.

16 [SPREAD Report]: 'Processor Products – Final Report of SPREAD Task Group, December 28, 1961', reprinted in Annals of the History of Computing 5 (1983) 6–26.

17 WISE, T.A.: 'I.B.M.'s $5,000,000,000 Gamble', Fortune (September 1966) 188–123, 224, 226, 228; 119.

18 'ICT 1900 Presentation', 16 mm. film (1964) ICL Historical Collection.

19 '... Minister of Technology Opens ICT's New R. & D. Laboratories', Computer Bulletin 9 (1965) 63–64.

20 STONEMAN, P.: 'Merger and technological progressiveness: the case of the British computer industry', Appl. Economics 10 (1978) 125–140.

21 'RCA's New Spectra 70 Series', Datamation (December 1964) 34–36.

22 WILSON, H.: The Labour Government 1964–1970 (1971) 8.

23 ILIFFE, J.K.: Basic Machine Principles (1968).

24 [Pears Report]: 'Rationalisation of Certain Companies in the British Computer Industry' (September 1966) ICL Archives.

25 'Report of the Working Party on a New Range of Computing Systems' (July 1967) ICL Archives, paragraph 20.

26 Industrial Investment: The Computer Mergers Project (Cmnd. 3660, 1968).

Notes on the authors

R.J. Allison

Dick Allison has worked for ICL since 1963, when he left the RAF and exchanged analogue radar sets for digital electronics. He started life in Computer Engineering on the EMIDEC 1100. He continued in this role on to the early 1900 systems, culminating in 1970 running the 1906A hardware support unit for Southern Division. In 1975 he moved to the 2900, where he led a team supporting the P Series of 2960/70/80. After an involvement with introducing ADEMS to the field he moved into S39, where he is now manager at the S39 Southern Region Support Centre.

V. Bodsworth

Having gained a degree in theoretical physics from Cambridge University, Vince Bodsworth joined the UK Marketing arm of ICT, as it then was, as a graduate trainee in 1967. He clearly had the wanderlust as, after the initial few years, he spent the next fifteen in ICL International Division, variously in East Africa, New Zealand, the USA, the Caribbean and at the International HQ in Putney. He has always had a strong technical bias but preferred to work in various marketing activities until settling at last in the UK, he became Director of Technology for F International in 1986. He now works as an independent consultant in marketing computer and communications products and services, and has recently become a member of the Computer Board for Universities and Research Councils.

Dr. M. Campbell-Kelly

Martin Campbell-Kelly graduated from the University of Manchester in 1968 with a B.Sc. in Computer Science, and received a Ph.D. in History of Science in 1980. He is now Lecturer in Department of Computer Science at the University of Warwick. He is presently engaged in a number of computer history projects, including a corporate history of ICL. He is Editor of the Charles Babbage Institute *Reprint Series for the History of Computing* and the *Collected Works of Charles Babbage*, and is also an editor of the Annals of the History of Computing.

G.M. Coiley

Glenn Coiley has worked for ICL since February 1987; previously he worked for STC as Repair and Refurbishment Manager in their Business Systems company. He had joined STC in 1972 as an electronics technician, becoming responsible for the repair centre in Data Systems Division in 1979. More recently he has been involved in product repair strategy and is now responsible for the development of repair methods, strategies and support functions for worldwide spares. He is particularly interested in planning and in tools for planning, and believes this to be crucial to the smooth introduction of products with ever-reducing timescales.

Dr. A. Flatman

Alan Flatman received a first-class honours degree in Electronics Engineering at the North Staffordshire Polytechnic in 1971, and later gained a Ph.D. in Electronics in conjunction with CNAA and ICL. He joined ICL formally in 1975 and has since held several positions in the development of display and communications products and technologies. Since 1981 he has been responsible for LAN technology and standards and for the last 4 years he has been ICL's consultant designer for LAN technology, in which role he has been an active participant in UK, US and European standards committees. He is a Member of the Institution of Electrical Engineers.

M.D. Griffiths

Dave Griffiths has worked in the computer industry since 1975, when he changed career after two years teaching English in Hampshire. His initial time in computers was spent with Portsmouth Polytechnic, first as an operator and subsequently (after two years of HNC night school) as a systems programmer in VME/B and VME/K. He joined ICL in 1980, when he combined his English and computer experience to work as a technical author. His last three years have been spent as Manager of the Customer Information Unit, which is responsible for producing the technical information needed by users of the Series 39 range of computers. His current particular interests are in the fields of electronic publishing and electronic distribution of information.

J. Houldsworth

Jack Houldsworth is Manager of the OSI Migration Project within ICL: his mission is to ensure that nothing impedes the ICL commitment to Open Systems. He has been involved in the standardisation process for many years and is widey regarded as the "Father of OSI". He is Chairman of the BSI committee on Data Transmission, has represented the UK at ISO/TC97/SC6 for over 20 years and has led the UK delegation since 1973.

He is the author of the OSI Handbook, quoted as Ref. 1 in this paper.

Dr. H. Johnson

Hilary Johnson graduated from Salford University in 1979 with a first degree in Psychology and Statistics and went on to receive a PhD in Psychology from Birmingham University. After four years carrying out research at the Medical Research Council's Cognitive Development Unit she joined the Computer Science Department at Queen Mary College, University of London, in October 1986 as a post-doctoral research assistant working on the current ICL-funded project. Her main research interests lie in the learning, planning and representation of knowledge in tasks.

Dr. P. Johnson

Peter Johnson is currently a lecturer in the Computer Science Department at Queen Mary College, University of London. He gained a BSc in Psychology and Statistics from Salford University in 1978 and a PhD in Cognitive Psychology from the University of Warwick. In 1981 he became a research fellow at the Ergonomics Unit, University College London, and there proposed a method of task analysis for knowledge descriptions (TAKD). He is involved in a number of Alvey research projects in Human Computer Interaction (HCI). He took up his project position in 1984 and has continued his research into HCI.

Dr. B.A. Kitchenham

Barbara Kitchenham received her Ph.D. from Leeds University in 1972. After working from some years as an applied statistician she joined ICL as a systems software programmer, and later designer and team leader, on the VME operating system. In 1980 she moved from program production to quantitative program evaluation and designed and introduced into the VME development group a detailed change-recording system. Her involvement with metrics research increased with the initiation of the Alvey and ESPRIT programmes, and in October 1986 she joined City University as a Reader in the Centre for Software Reliability. She has published over 20 papers on software metrics, Quality Assurance and management.

J.D. Mitcalf

Dave Mitcalf obtained his degree in Management Sciences at UMIST and joined ICL in 1979, where he worked in the STAGE II project for the DHSS, developing front-end processor software. In 1983 he joined the newly-formed Knowledge Engineering Business Centre, where he worked on the design and development of ADVISER, the ICL expert system shell. Since that time he has been involved in the implementation of a variety of knowledge-based systems and has also collaborated in the ESPRIT project EUROHELP, researching in intelligent help systems.

B. Parker

Brian Parker has worked in many areas of ICL's Customer Service organisation since joining Ferranti in 1960. He provided engineering support worldwide initially for Ferranti's first transistorised computer, the Sirius, and then for ICL's first large 1900 series machines after working on their development in Manchester. He then managed a small engineering software team developing early diagnostic and prognostic software. This was followed by a secondment to Australia to set up and manage the local Customer Service support and training function. He now operates as a consultant providing evaluation of high technology tools and their application to the solving of system problems. He received the Chairman's Excellence Award in 1986.

A.J. Russell

Fred Russell joined ICL in 1967 as a Systems Engineer, subsequently transferring to selling 1900 systems in the National Accounts Area. On leaving the sales force he had a varied career within ICL, ranging through business planning, market introduction and programme management of 2900 development and introduction. He then became involved in the emerging office automation strategy, and in particular in the human factors aspect of this. Currently he manages the Human Factors Group in the Systems Strategy Centre at Bracknell and is responsible for the ICL component of an ESPRIT programme aimed at developing knowledge-based user-centred software development tooling.

J.G. Walker

John Walker graduated from Cambridge University in 1959 with a degree in mathematics, took a Diploma in Numerical Analysis and then joined the computer division of English Electric (later part of ICL). Work on the design and development of software was followed by a variety of management posts. Deeply involved in software "quality" since 1976, he has worked for the past three years as a Principal Software Engineer on the Alvey Test Specification and Quality Management project. As a result of ICL incorporation into STC he now works for STC Technology Ltd. He is a Fellow of the British Computer Society and of the Institute of Quality Assurance.

J. Young

Jim Young has worked for ICL since 1973. He has been involved with various aspects of system implementation and support on 1900, System 4, 2900 Series and Series 39. As a former member of the Wokingham Systems Maintenance Centre he was responsible for much of the early VME diagnostic documentation on scheduling systems. His present job is the management of the Software & Services Unit in Reading.