The **Systems** *Journal*

ICL

Volume 14 Issue 1 *Autumn 1999 Edition*

ICL Systems Journal

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.

Published twice a year by Group Technical Directorate, ICL, Stevenage.

1999 subscription rates (including postage & packing):

	UK and Europe	Rest of World
Annual subscription	£72	$120
Single issues	£43	$72

ISSN 1364-310X

ICL Systems Journal

Volume 14 Issue 1

Contents

Interactive Digital Television
A Systems Overview

John V Panter

ICL Fellow Emeritus

Abstract

1999 has seen the introduction of a number of digital television services in the UK, delivered by satellite, terrestrial broadcast, and cable. Whilst initially restricted to television programming, pay per view movies, and an electronic program guide, all these systems plan to provide a wealth of interactive services via the television set that have until now required a personal computer. These interactive services will include electronic mail and electronic commerce as well as access to the Internet World Wide Web, and will also evolve to provide a unique range of new services that combine the attributes of television and interactive services.

This paper provides a background to interactive digital television and the technologies involved. It then goes on to provide an overview of the systems which are being developed to deliver these services. This involves the integration of systems from the broadcast and IT industries, an example of where digital technologies are forcing a convergence between two industries.

Introduction

In the UK, during 1999, we have seen the introduction of digital television services; delivered by satellite, terrestrial broadcast, and cable. Initially these services delivered television channels, pay per view movies, and an electronic program guide. However, all the service operators plan to introduce a range of interactive services. Many of these services will be familiar to personal computer users who access the Internet through an Internet Service Provider (ISP), but new services will also be provided which combine the characteristics of television and interactive services.

The original stimulus for the introduction of digital television was the more efficient use of bandwidth. This was essential for terrestrial broadcasting, where the demands on the electromagnetic spectrum had become critical. It was becoming necessary to release bandwidth for purposes other than the broadcast of entertainment, and is the motivation behind recent announcements about the timing of the switch-off of analogue services.

At the same time there was a growing demand to broadcast more entertainment channels, and to provide better quality audio/visual content. This is easier to achieve with digital technologies. Although the current digital services will not incorporate high definition television, the quality of the sound and pictures will still be better than in the analogue services.

As a result of these demands, in Europe the broadcasters and consumer electronics manufacturers came together as the Digital Video Broadcasting (DVB) group to agree a set of standards. Similar groupings occurred in the US and, as usual, a different set of standards have emerged. This paper will concentrate on the European standards.

In parallel with the above activities of the broadcasters, the Internet phenomenon has swept the world. This has been linked with many initiatives and strategies, grouped under the general banner of the Information Society. The initial concern was that these services were restricted to those who owned a personal computer and that, therefore, many would be disenfranchised from this information revolution. However, with the ongoing march of technology, there is now talk of accessing these services from a growing range of products.

The obvious target was to offer access to Internet services through the television set. There is a 98% penetration of television sets in UK households and, therefore, an opportunity to link many new users to the Information Society. This is achievable in conjunction with the introduction of digital television.

However, a television is a very different platform to a personal computer, particularly with respect to the display characteristics and the manner in which it is used. The TV viewer behaves very differently from the personal computer user. These differences need to be taken into consideration when designing an interactive television system, particularly in regard to the design and presentation of the interactive content. This paper tries to indicate these problems and how systems are being designed to solve them.

The subject matter in interactive digital television is very wide in scope. It has therefore been necessary in this paper to over simplify grossly the details of the technology in order to explain the basic principles.

What is Interactive Digital Television?

Before looking at the technology it is necessary to ensure that the reader has an appreciation of what Interactive Digital Television is from a user perspective.

Digital Television

The transmission of digital television is by digitally encoded signals. MPEG-2 has been universally adopted as the standard for the encoding of digital television broadcasts. This is obviously transparent to the end user, except for the fact that it requires a Set Top Box (STB) to connect to the service. Eventually the STB will become integrated with the television, at least in the case of terrestrial digital broadcasts, but, for the present, a STB is required to receive the digital signals and convert them to the analogue form required by the television set input.

There will be some improvement of picture and sound quality, but otherwise the major differences that the user will notice are:

- The number of television and audio channels available. This could be in the range of 30 to 100 channels dependant on the broadcaster and the subscription paid by the end user
- The availability of pay per view (PPV) events, particularly movies. These may be individual events requiring the user to book through the broadcaster, but increasingly the events will be impulsively bought through the Electronic Program Guide (EPG), with the charges being automatically added to the monthly bill. There will also be near video on demand services, where a small set of movies are continually broadcast at short intervals to allow the end user to select when the movie is viewed
- An Electronic Program Guide. This is an interactive application that runs in the STB. In its basic form it provides the user with information about all the programs being broadcast, and the ability to select immediately the program they want. The information presented is normally in the form of what is on now and on next, as well as what will be broadcast over the next few days. For pay per view programs, the EPG supplies details of the movie or event, as well as the ability to buy the event. In more sophisticated forms the EPG can provide more functionality, such as allowing the user to select which programs they want to watch over a period of time, and then providing a reminder when the event is about to be broadcast.

Interactive Services via the TV Screen

The EPG is an example of an interactive service that runs entirely within the STB, using data that is provided as part of the MPEG-2 broadcast stream. This same principle can be used for other interactive services, but they are limited in scope and restricted by the amount of data that can be continuously broadcast over the MPEG-2 streams. This issue is discussed in more detail below.

The real requirement for interactive services is a bi-directional communications connection between the STB and the delivery systems. This is why for satellite and terrestrial digital broadcasts the end user is requested to connect the STB to their telephone line. This is not required for cable connections, since the requirement is met from bandwidth on the cable connection itself, leaving the telephone connection untouched.

With this "secondary" bi-directional connection, the way is open for true interactivity, limited only by the bandwidth available. Under these conditions the user will be offered a range of interactive services such as:

- E-mail, although this may be with less functionality than is normally available with personal computer connections
- A range of services that have been designed for the medium. These will normally be derived from the Web service of a content provider, but with a "look and feel" that is matched to the television environment. These services will often be structured so that they can easily be navigated from the User interface. The range of applications offered is potentially endless, but most will include:
  - o E-commerce applications within the service list, providing such things as home shopping and home banking
  - o News services
  - o Thematic news and information services such as sport, entertainment, travel, etc.
  - o On-line games
  - o Bulletin board and chat environments
- Internet World Wide Web access.

In many instances it will also be possible to continue watching the television channel whilst using the interactive service.

Enhanced Television

With the ability to broadcast data within the audio/video streams, and bi-directional data communication over the "secondary" connection, the door is open to a wide range of new applications. Some of the possibilities are:

- Providing the ability to select different camera angles at sports events
- Displaying advertisements from which the user can follow through for more information, or even buy
- Ability to display additional information associated with the television event
- Complementing educational programs with additional material
- Ability to vote on-line in real time.

The services will be further enhanced when video material can be broadcast to complement interactive services. This will be a major application once true video on demand systems become available, particularly for educational and instructional applications.

The Delivery of Digital Television

Since broadcasters were building the systems, it was natural that they should design them as an evolution of their analogue systems. This approach is also mandated by the commercial agreements between the film studios and television program providers, who insist on the use of conditional access systems originally designed for the analogue systems. Whilst this has dictated less than ideal architectural solutions for interactive digital television, it has had the benefit that digital television can be launched ahead of the availability of interactive services.

This section of the paper looks at the basic technology for the delivery of digital television.

Standards

As previously stated, the industry came together to agree on the standards to be used for digital television. In Europe, this grouping was the DVB group, which has used existing standards where possible, and has released many of its documents through ETSI, the European Telecommunications Standards Institute.

One of the first fundamental decisions taken was the selection of MPEG-2 as the coding of video and audio, and the creation of program streams. However, the MPEG-2 standard is too generic and too wide in scope, and DVB therefore provided details on how the standard should be used. As well as defining the standards for video and audio streams, it has also included the broadcast transmission of data streams. This latter covers application areas such as data piping, data streaming, multi-protocol encapsulation, and data carousels. A subset of DSM-CC from the MPEG standards is defined for layers 2 and 3 of the protocol stack.

Other parts of the DVB work covered the technical specifications of the baseband transmission system; for the broadcast of MPEG-2 transport streams over satellite, terrestrial broadcast, microwave, and cable systems, including the use of telecommunications networks such as PDH, SDH, and ATM to distribute the baseband signals.

Most service providers already have a Subscriber Management System (SMS) that they are unwilling to throw away, and Conditional Access Systems (CAS) tend to be closely guarded secrets by the companies that own the technology. Therefore DVB decided not to standardise on the SMS or the Subscriber Authorisation System (SAS) that encrypts and delivers code words that enable the descrambler in the STB. However, in order to allow content material to be supplied across multiple systems they defined the common scrambling algorithm, and a common interface which can be used by the different conditional access systems. The latter includes an interface to the smart card used by conditional access systems.

Even in the basic forms of digital television there is the need for some interaction between the user and the delivery system. DVB have therefore specified the lower layers of the ISO model for the use of PSTN, ISDN, and cable networks.

The Principles of Operation

The following is a simplified description of the basic principles behind the delivery of digital television.

The Efficient use of Bandwidth

In Europe the bandwidth used for the transmission of television services is 8MHz. For analogue services this is used for the transmission of a single television channel. However, digital television uses Quadrature Amplitude Modulation (QAM) with forward error correction, and with 64QAM (where the modulation provides 64 uniquely identifiable states) it is possible to transmit 38Mbits per second over this single 8MHz channel, normally referred to as a multiplex. With MPEG-2 encoding, good quality television pictures can be transmitted at around 3.6 to 4.0 Mb/s. In crude terms, therefore, there is a tenfold increase in the use of the bandwidth. However, in practice the situation is more complex, with the MPEG-2 stream containing not only television channels, but audio channels and data streams as will be described below.

In the UK, the Government has released six multiplexes for use by terrestrial digital television broadcasting. Satellites usually provide a number of 33MHz transponders, each therefore capable of delivering four multiplexes. Cable systems usually use bandwidth up to 550MHz for analogue transmissions, but modern networks are capable of working to 750MHz or above, and therefore have the capacity of providing many multiplexes.

In the near future it is expected that the technology will move to 256QAM modulation, providing another 4-fold increase in the use of available bandwidth.

Structure of the Transport Stream

MPEG-2 defines the concept of program and transport streams. A program stream is normally associated with a single television channel, or movie, or individual service. It is a single data stream made up by multiplexing any combination of elementary component streams: video, audio and data. All components in a program stream therefore have a common time base. A transport stream is the basic data stream transmitted over the multiplex, and is constructed by multiplexing together all the elementary component streams associated with the services. This can include program streams, which will be incorporated into the transport stream as individual component streams.

This provides the mechanism for carrying a number of services and data streams over a single multiplex, as is illustrated in Figure 1.

The detailed manner in which transport streams are constructed is illustrated in Figure 2. Each elementary data stream is packetised into PES packets, with the header containing a description of the data and the number of bytes in the packet. The transport stream is constructed by multiplexing the elementary data streams, whether provided individually or as part of the program streams. The transport stream is itself packetised, the payload being the PES packets of the elementary streams. Since the size of a transport stream packet is 188 bytes, including the header, a number of packets will be required to deliver each PES packet. The transport stream packets that carry the PES packets associated with a particular elementary stream are given a unique Packet Identifier (PID) in their header. This allows the STB to identify and extract the necessary packets to reconstruct a particular elementary stream.

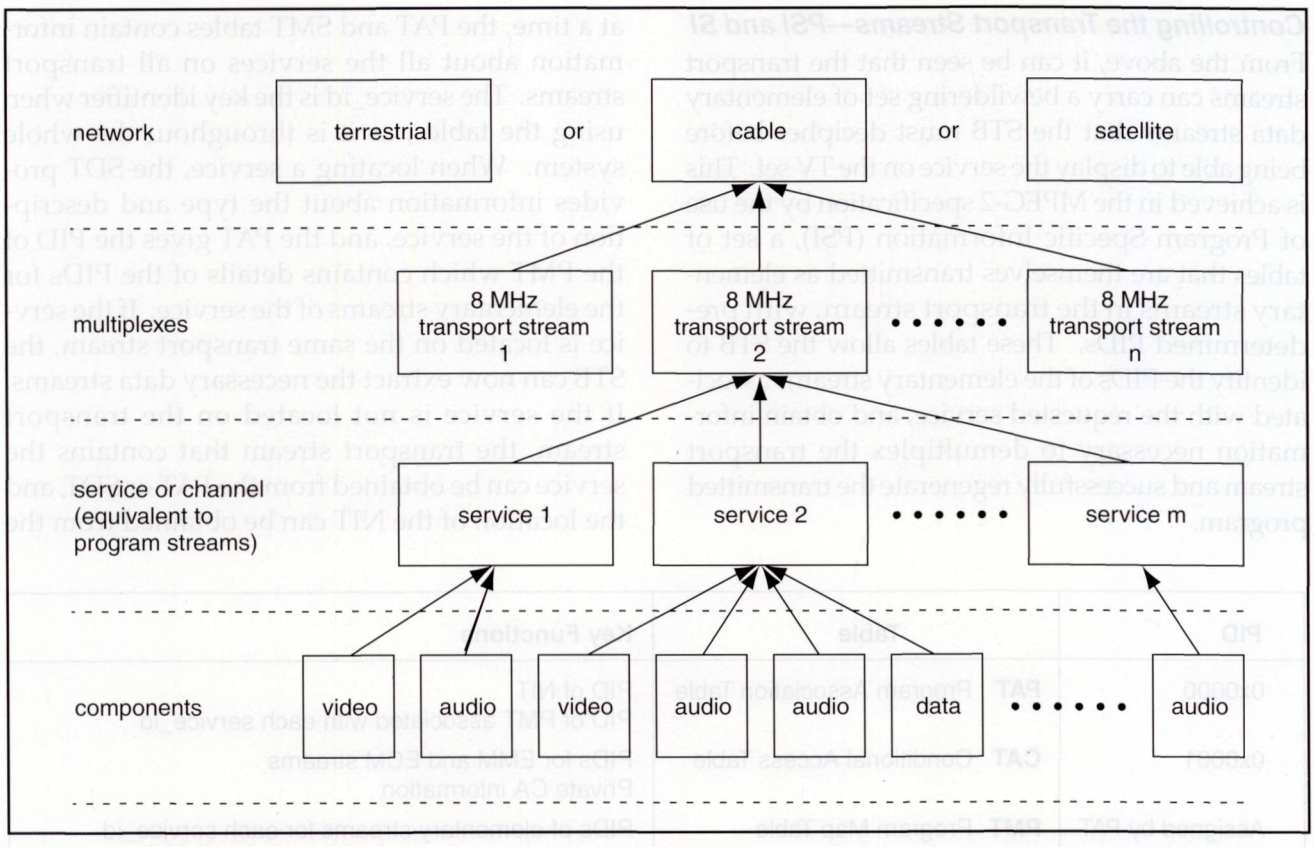


Figure 1: Carrying Multiple Services in a Digital Television Service

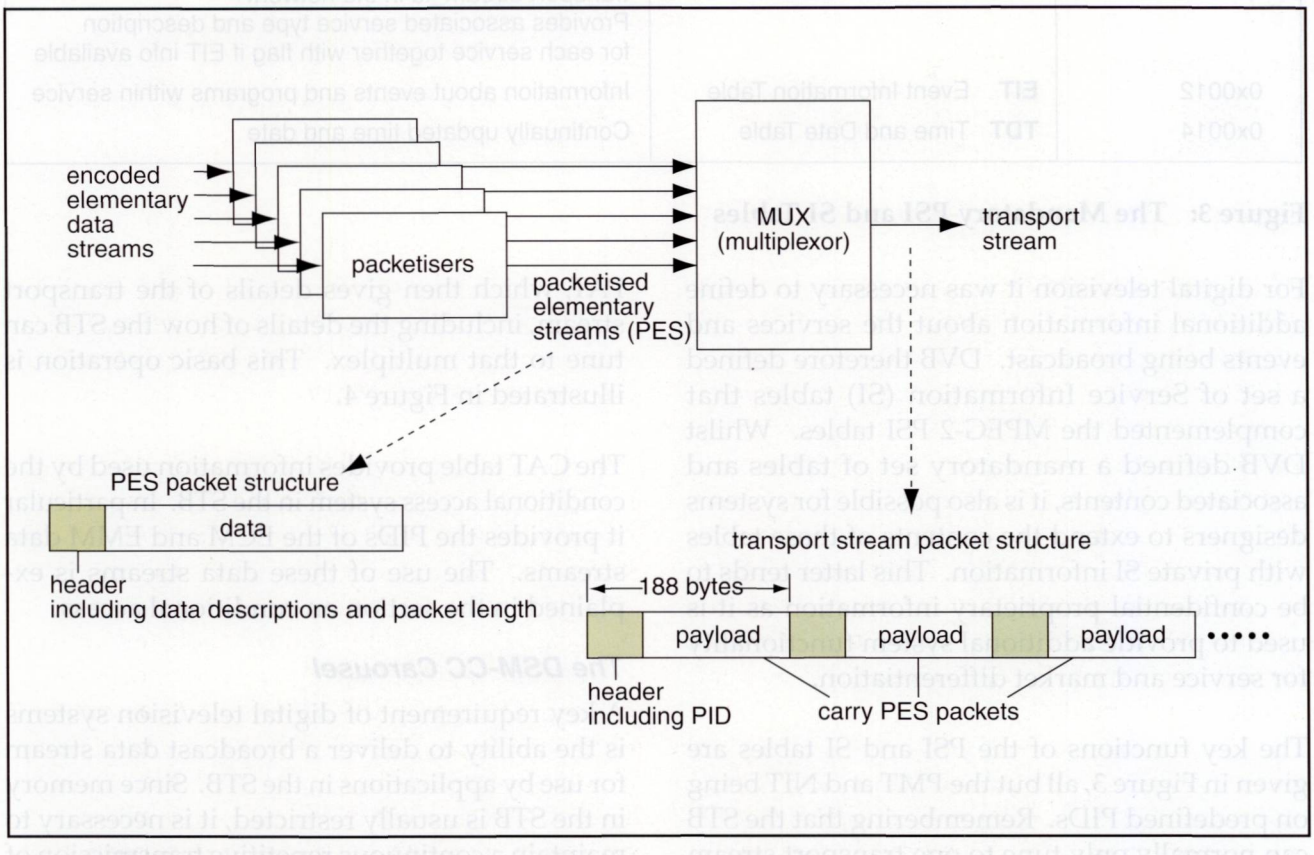


Figure 2: The Structure of the Transport Stream

Controlling the Transport Streams—PSI and SI

From the above, it can be seen that the transport streams can carry a bewildering set of elementary data streams that the STB must decipher before being able to display the service on the TV set. This is achieved in the MPEG-2 specification by the use of Program Specific Information (PSI), a set of tables that are themselves transmitted as elementary streams in the transport stream, with predetermined PIDs. These tables allow the STB to identify the PIDs of the elementary streams associated with the requested service, and obtain information necessary to demultiplex the transport stream and successfully regenerate the transmitted program.

at a time, the PAT and SMT tables contain information about all the services on all transport streams. The service_id is the key identifier when using the tables, as it is throughout the whole system. When locating a service, the SDT provides information about the type and description of the service, and the PAT gives the PID of the PMT which contains details of the PIDs for the elementary streams of the service. If the service is located on the same transport stream, the STB can now extract the necessary data streams. If the service is not located on the transport stream, the transport stream that contains the service can be obtained from the PAT or SDT, and the location of the NIT can be obtained from the

PID	Table	Key Functions
0x0000	**PAT** Program Association Table	PID of NIT PID of PMT associated with each service_id
0x0001	**CAT** Conditional Access Table	PIDs for EMM and ECM streams Private CA information
Assigned by PAT	**PMT** Program Map Table	PIDs of elementary streams for each service_id
Assigned by PAT	**NIT** Network Information Table	List of transport stream_ids in network, with tuning information
0x0011	**SDT** Service Description Table	List of all service_ids associated with each transport stream_id in the network Provides associated service type and description for each service together with flag if EIT info available
0x0012	**EIT** Event Information Table	Information about events and programs within service
0x0014	**TDT** Time and Date Table	Continually updated time and date

Figure 3: The Mandatory PSI and SI Tables

For digital television it was necessary to define additional information about the services and events being broadcast. DVB therefore defined a set of Service Information (SI) tables that complemented the MPEG-2 PSI tables. Whilst DVB defined a mandatory set of tables and associated contents, it is also possible for systems designers to extend the contents of these tables with private SI information. This latter tends to be confidential proprietary information as it is used to provide additional system functionality for service and market differentiation.

The key functions of the PSI and SI tables are given in Figure 3, all but the PMT and NIT being on predefined PIDs. Remembering that the STB can normally only tune to one transport stream

PAT, which then gives details of the transport stream, including the details of how the STB can tune to that multiplex. This basic operation is illustrated in Figure 4.

The CAT table provides information used by the conditional access system in the STB. In particular it provides the PIDs of the ECM and EMM data streams. The use of these data streams is explained in the section on conditional access.

The DSM-CC Carousel

A key requirement of digital television systems is the ability to deliver a broadcast data stream for use by applications in the STB. Since memory in the STB is usually restricted, it is necessary to maintain a continuous repetitive transmission of

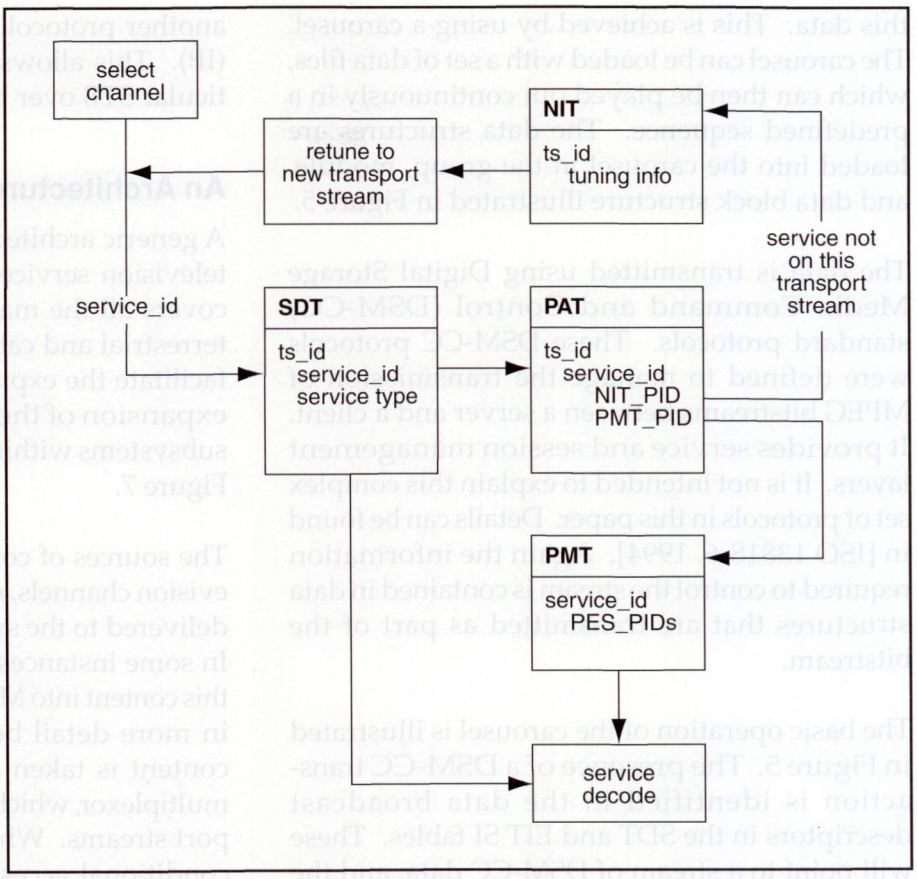


Figure 4: Using the PSI/SI Tables to Decode a Selected Service

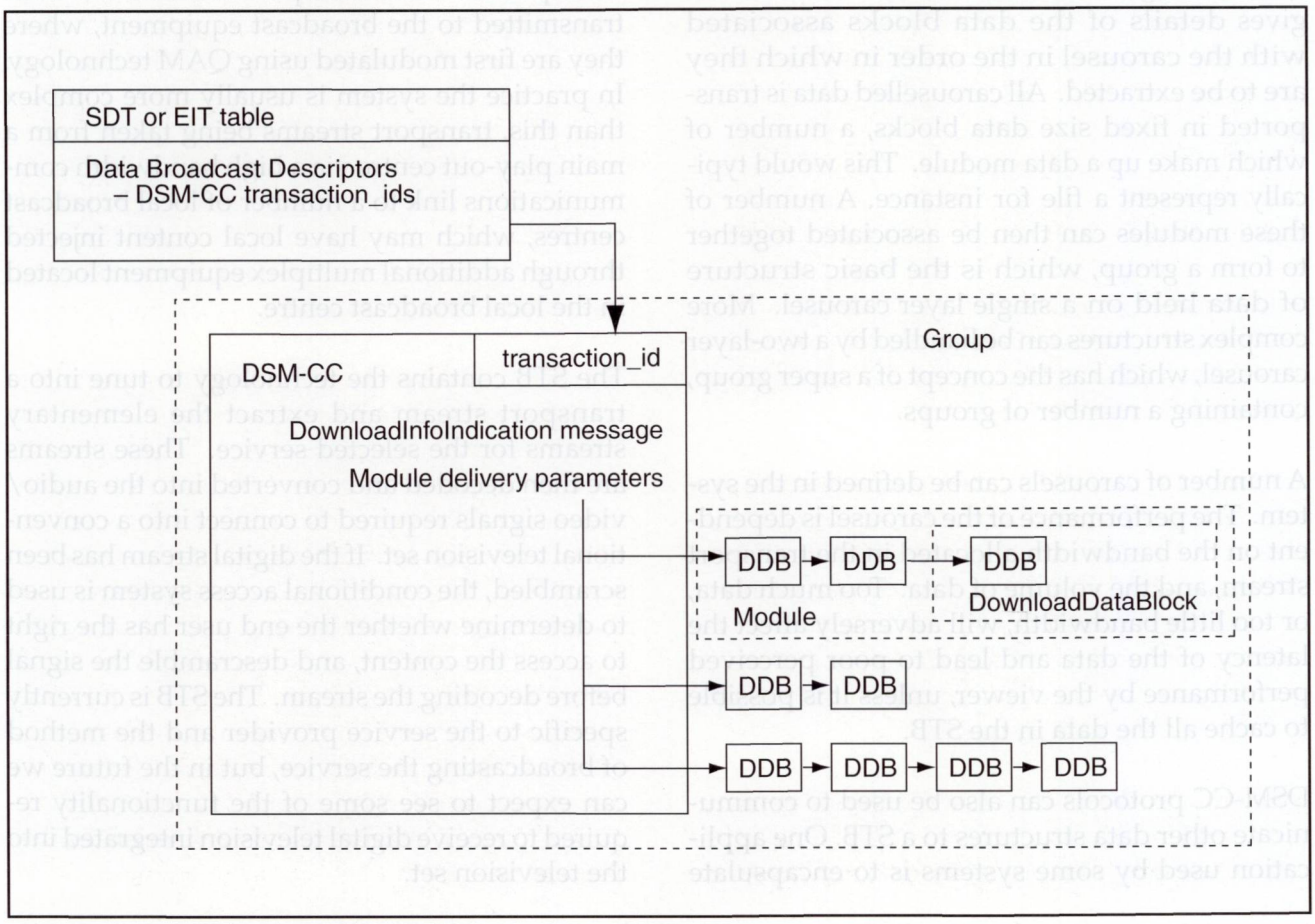


Figure 5: The DSM-CC Data Carousel

this data. This is achieved by using a carousel. The carousel can be loaded with a set of data files, which can then be played out continuously in a predefined sequence. The data structures are loaded into the carousel in the group, module, and data block structure illustrated in Figure 5.

The data is transmitted using Digital Storage Media Command and Control (DSM-CC) standard protocols. These DSM-CC protocols were defined to manage the transmission of MPEG bitstreams between a server and a client. It provides service and session management layers. It is not intended to explain this complex set of protocols in this paper. Details can be found in [ISO 13818-6, 1994]. Again the information required to control the stream is contained in data structures that are transmitted as part of the bitstream.

The basic operation of the carousel is illustrated in Figure 5. The presence of a DSM-CC transaction is identified in the data broadcast descriptors in the SDT and EIT SI tables. These will point to a stream of DSM-CC data, and the transaction_id of a Download-Info-Indication (DII) message within that stream. The DII then gives details of the data blocks associated with the carousel in the order in which they are to be extracted. All carouselled data is transported in fixed size data blocks, a number of which make up a data module. This would typically represent a file for instance. A number of these modules can then be associated together to form a group, which is the basic structure of data held on a single layer carousel. More complex structures can be handled by a two-layer carousel, which has the concept of a super group, containing a number of groups.

A number of carousels can be defined in the system. The performance of the carousel is dependent on the bandwidth allocated in the transport stream, and the volume of data. Too much data, or too little bandwidth, will adversely affect the latency of the data and lead to poor perceived performance by the viewer, unless it is possible to cache all the data in the STB.

DSM-CC protocols can also be used to communicate other data structures to a STB. One application used by some systems is to encapsulate another protocol, particularly Internet Protocol (IP). This allows the sending of data to a particular STB over the broadcast system.

An Architecture for Digital Television

A generic architecture for the delivery of digital television services is illustrated in Figure 6. It covers all the major delivery systems; satellite, terrestrial and cable, but is greatly simplified to facilitate the explanation of the principles. The expansion of the functionality of some of the subsystems within this architecture are shown in Figure 7.

The sources of content will be multifarious (television channels, music channels, films, etc.), and delivered to the system in many different forms. In some instances it will be necessary to encode this content into MPEG-2 format. This is discussed in more detail below. The MPEG-2 encoded content is taken as elementary streams into a multiplexor, which assembles them into the transport streams. Where the content is protected by conditional access, the data in the elementary streams is scrambled before being passed to the multiplexor. The transport streams are then transmitted to the broadcast equipment, where they are first modulated using QAM technology. In practice the system is usually more complex than this, transport streams being taken from a main play-out centre via a high bandwidth communications link to a number of local broadcast centres, which may have local content injected through additional multiplex equipment located in the local broadcast centre.

The STB contains the technology to tune into a transport stream and extract the elementary streams for the selected service. These streams are then decoded and converted into the audio/video signals required to connect into a conventional television set. If the digital stream has been scrambled, the conditional access system is used to determine whether the end user has the right to access the content, and descramble the signal before decoding the stream. The STB is currently specific to the service provider and the method of broadcasting the service, but in the future we can expect to see some of the functionality required to receive digital television integrated into the television set.

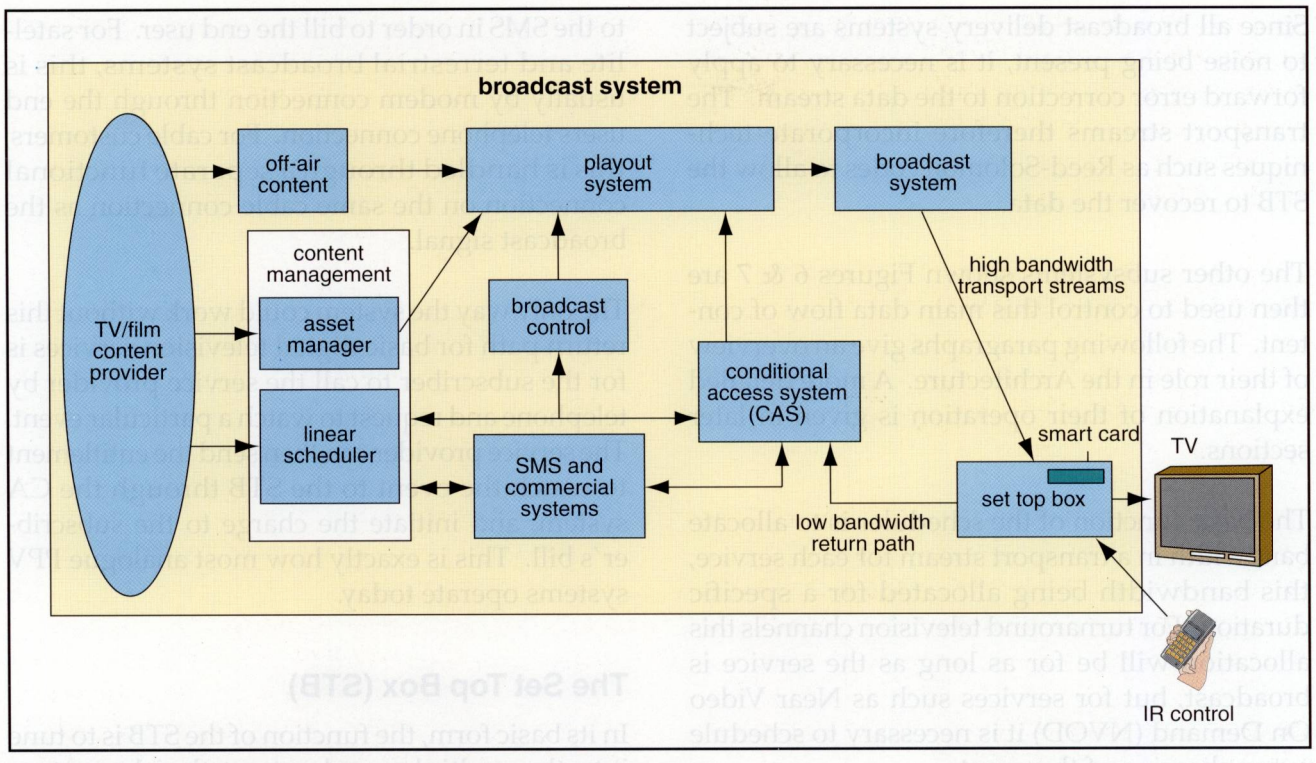


Figure 6: Generic Architecture for Digital Television System

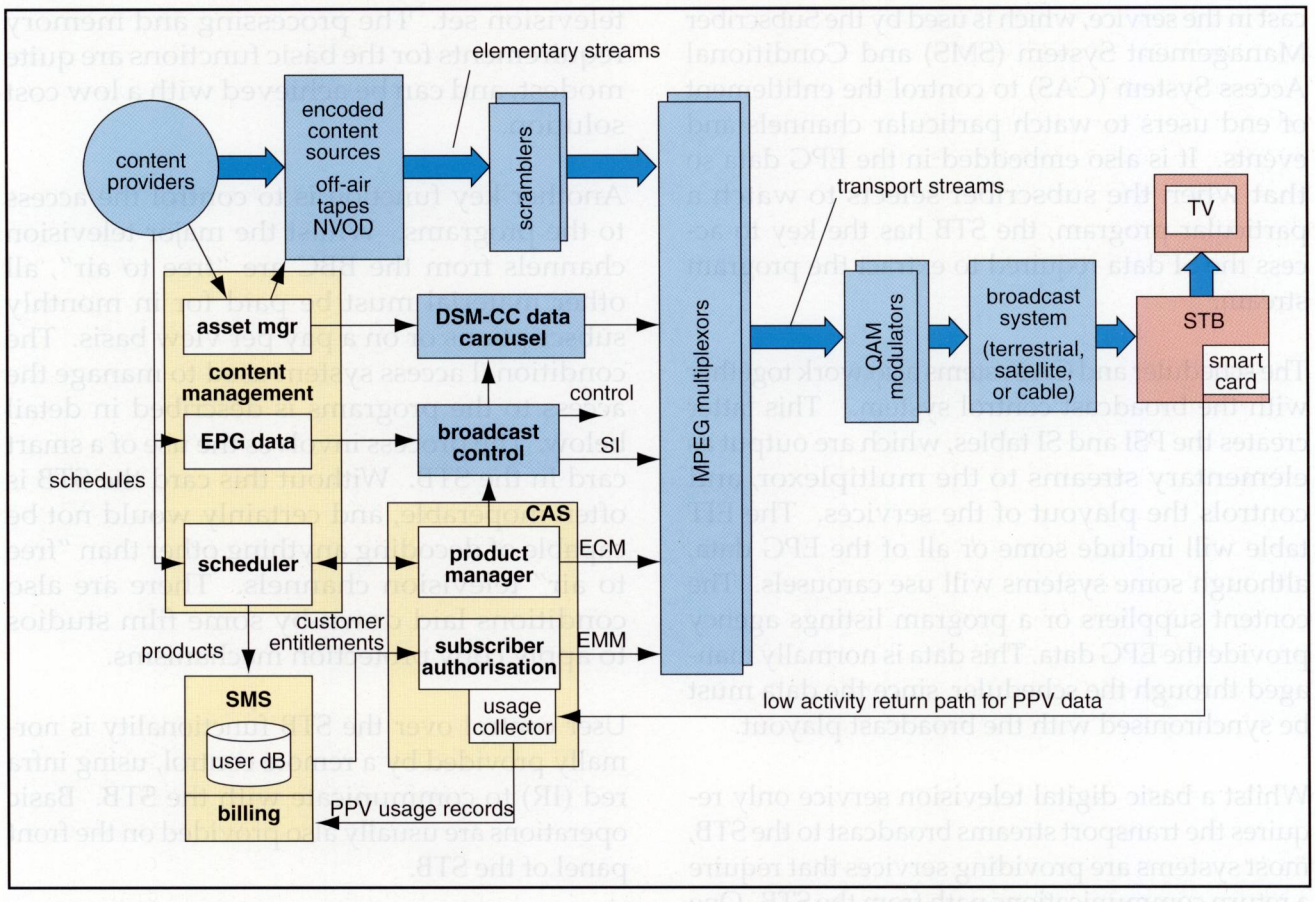


Figure 7: The Delivery of Digital Television

Since all broadcast delivery systems are subject to noise being present, it is necessary to apply forward error correction to the data stream. The transport streams therefore incorporate techniques such as Reed-Solomon codes to allow the STB to recover the data.

The other subsystems shown Figures 6 & 7 are then used to control this main data flow of content. The following paragraphs give an overview of their role in the Architecture. A more detailed explanation of their operation is given in later sections.

The basic function of the scheduler is to allocate bandwidth in a transport stream for each service, this bandwidth being allocated for a specific duration. For turnaround television channels this allocation will be for as long as the service is broadcast, but for services such as Near Video On Demand (NVOD) it is necessary to schedule every showing of the event.

The scheduler also allocates a product identity, service_id, for each channel or event to be broadcast in the service, which is used by the Subscriber Management System (SMS) and Conditional Access System (CAS) to control the entitlement of end users to watch particular channels and events. It is also embedded in the EPG data so that when the subscriber selects to watch a particular program, the STB has the key to access the SI data required to extract the program stream.

The scheduler and CA systems also work together with the broadcast control system. This latter creates the PSI and SI tables, which are output as elementary streams to the multiplexor, and controls the playout of the services. The EIT table will include some or all of the EPG data, although some systems will use carousels. The content suppliers or a program listings agency provide the EPG data. This data is normally managed through the scheduler, since the data must be synchronised with the broadcast playout.

Whilst a basic digital television service only requires the transport streams broadcast to the STB, most systems are providing services that require a return communications path from the STB. One example is the "on-line" impulse buying of a PPV event. Once the event has been bought and watched, a data record must be sent from the STB to the SMS in order to bill the end user. For satellite and terrestrial broadcast systems, this is usually by modem connection through the end users telephone connection. For cable customers, this is handled through a separate functional connection on the same cable connection as the broadcast signal.

The only way the system could work without this return path for basic digital television services is for the subscriber to call the service provider by telephone and request to watch a particular event. The service provider can then send the entitlement to watch the event to the STB through the CA system, and initiate the charge to the subscriber's bill. This is exactly how most analogue PPV systems operate today.

The Set Top Box (STB)

In its basic form, the function of the STB is to tune into the multiplex and extract the elementary MPEG streams for the selected program, and then decode the streams into video and audio signals that can be seen and heard on a conventional television set. The processing and memory requirements for the basic functions are quite modest, and can be achieved with a low cost solution.

Another key function is to control the access to the programs. Whilst the major television channels from the BBC are "free to air", all other material must be paid for in monthly subscriptions or on a pay per view basis. The conditional access system used to manage the access to the programs is described in detail below. The process involves the use of a smart card in the STB. Without this card the STB is often inoperable, and certainly would not be capable of decoding anything other than "free to air" television channels. There are also conditions laid down by some film studios to apply copy protection mechanisms.

User control over the STB functionality is normally provided by a remote control, using infra red (IR) to communicate with the STB. Basic operations are usually also provided on the front panel of the STB.

The other key function of the STB is to provide an electronic program guide.

The software in the STB is complex and will require modifications and enhancements over time. Most STBs therefore have a minimal set of embedded code that controls the downloading of the software from a data stream in one of the transport streams.

The Electronic Program Guide (EPG)

The EPG is a key application provided by all digital television services. It not only provides information about what is on now and on next, but also provides full details of all programs and services that will be broadcast over the next few days. Often this will include subsidiary information about the program or event. The amount and depth of data available is dependent on the amount of bandwidth that can be spared in the broadcast transport streams, and the amount of caching available in the STB. The minimum now and next information of all programs must be included in the EIT table on every multiplex, but if a rich amount of information is required for all programs and events to be broadcast over the coming week, it is likely to be broadcast from carousels and would need most of a multiplex unless a significant amount of caching was available in the STB in order to reduce latency.

The EPG not only provides information about the programs, but is also used as a prime navigation tool through the services. With the very large number of channels and events that will be broadcast, it is necessary to provide user-friendly methods of finding and accessing the programs. This is achieved using the remote control and navigational help built into the EPG. More sophisticated EPG programs provide the ability to set up viewing requirements for several days to come, and a system to alert the viewer that a selection is about to start.

Conditional Access

The overall system design of any subscription television service is dominated by the security systems required to prevent unauthorised access to the system and the broadcast content. Since this is a running battle between the systems designers and the hackers, much of the detail is kept secret. The key systems are housed in secure rooms, and only those who really need to know are in possession of all the details. This section can therefore only describe the basic mechanisms that are in the public domain.

The MPEG-2 elementary streams are scrambled using a common scrambling algorithm defined

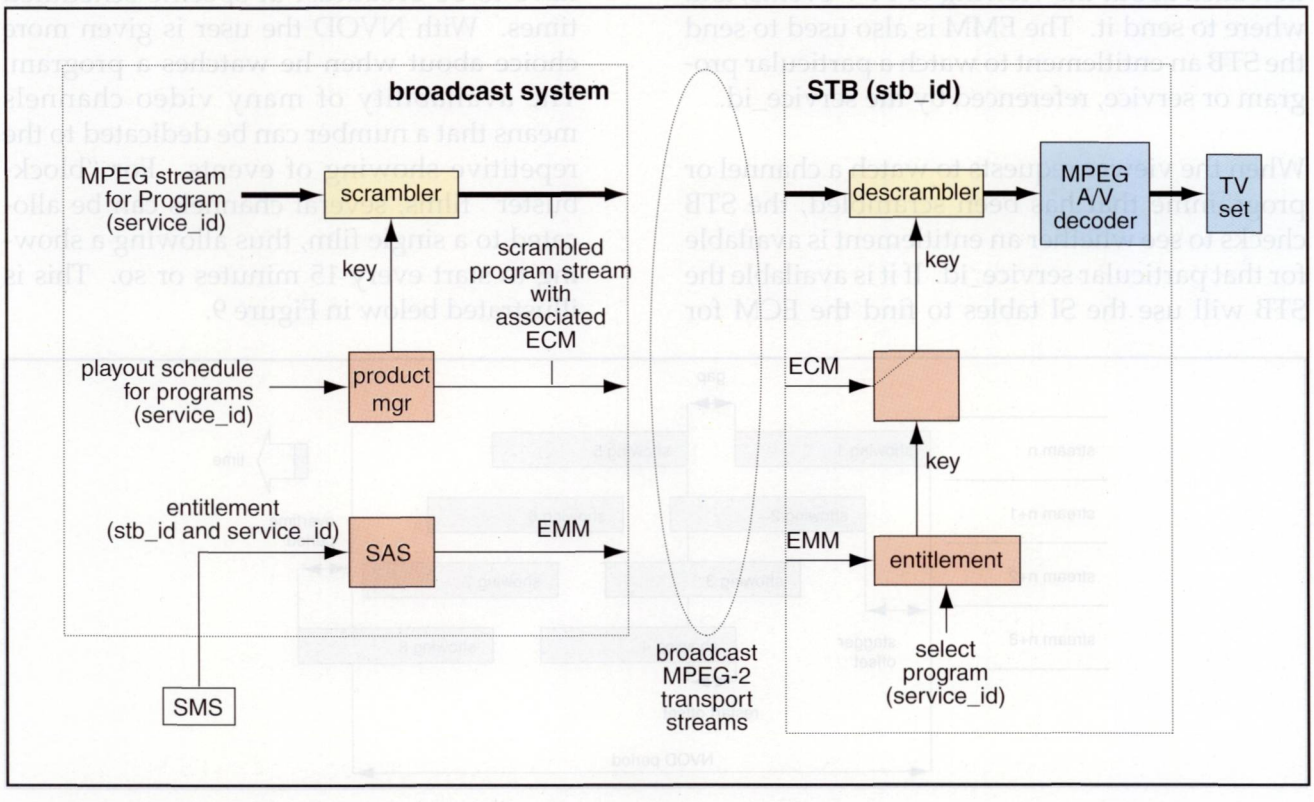


Figure 8: The Basics of Conditional Access

by DVB. It is therefore necessary for the STB to have a control word with which to descramble the stream before it can be decoded into the audio and video signals required by the television. The function of the CAS is to manage the scrambling of the elementary streams and the passing of the unscrambling key to specific STBs. This overall process is illustrated in Figure 8.

The system makes use of the Entitlement Control Message (ECM) and the Entitlement Management Message (EMM) specified in the MPEG standards, although much of the content of these messages remains private and secret.

The ECM contains specific CA information about the scrambling of the program stream. The ECM is broadcast on its own PID synchronously with the program it refers to. The ECM is repetitively broadcast during the program, and the conditional access content is often dynamic in nature.

EMMs are also broadcast, but are addressed to a specific STB or smart card. The SMS knows the STB or smart card identity, and can therefore send an EMM to a particular customer through the SAS in the CAS. EMM's can be used to send information and control a number of functions in the STB, for instance when to send information back about the viewing of PPV events, and where to send it. The EMM is also used to send the STB an entitlement to watch a particular program or service, referenced by the service_id.

When the viewer requests to watch a channel or programme that has been scrambled, the STB checks to see whether an entitlement is available for that particular service_id. If it is available the STB will use the SI tables to find the ECM for that service_id, and the information from the entitlement and ECM is sufficient to enable the descrambler.

Content Sources

The content will be supplied from a number of sources, as described in the following sections.

- Turnaround television and radio channels that are supplied by third party media companies. These may be delivered via cable, terrestrial broadcast, or satellite. The service provider does not change this content, but schedules it into a transport stream.

- Near Video on Demand (NVOD) events that are played out from servers within the playout system under the control of the scheduler and broadcast control system. The film material is usually delivered on tape and will need to be encoded into MPEG-2 before being loaded onto the delivery servers. These processes are controlled through the asset manager in the content control system. See Figure 7.

  In analogue systems, bandwidth restrictions mean that films and any other PPV events have to be broadcast at specific scheduled times. With NVOD the user is given more choice about when he watches a program. The availability of many video channels means that a number can be dedicated to the repetitive showing of events. For "blockbuster" films, several channels can be allocated to a single film, thus allowing a showing to start every 15 minutes or so. This is illustrated below in Figure 9.

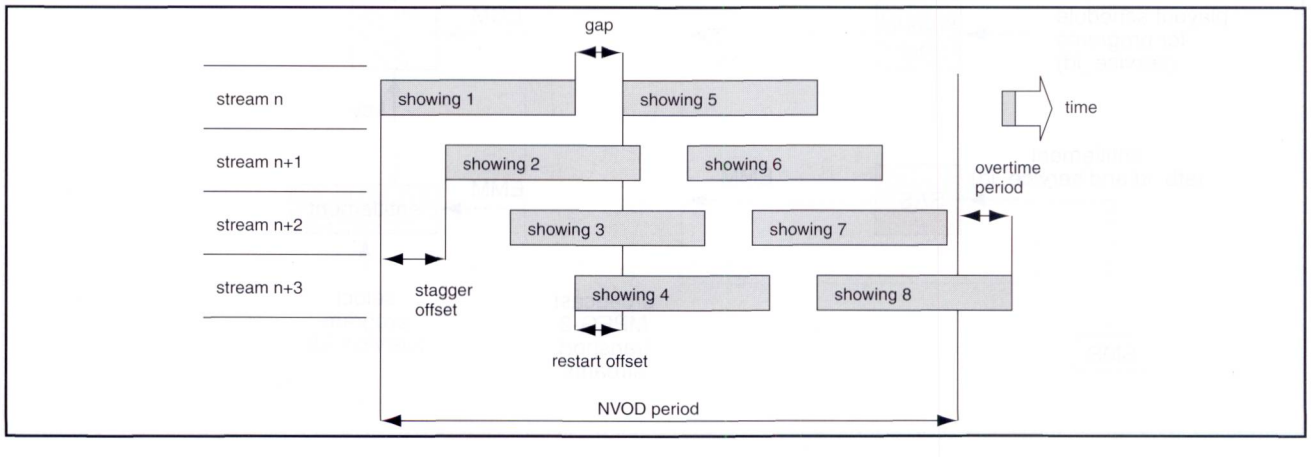


Figure 9: The Playout of a Single Film using NVOD

- Details of turnaround channel programmes are received from listings agencies, which are used as raw EPG data. Information about the films and PPV events is provided by the content providers, and again fed into the EPG data files that are managed through the content management system.

Adding Interactive Services

In the above we have seen how a basic digital television service works. As described, this is a natural evolution from the analogue broadcast systems providing subscription and PPV services. We shall now consider how interactive services can be added to the system.

Client-side Applications

With suitable processing and memory resources in the STB, it is possible to download software and data for a specific application when that service is selected by the user. This can then use additional data broadcast on carousels, or different elementary streams, to give a range of interactive services, albeit the interactivity is at the client level in the STB. In this way it is relatively simple to provide services such as the selection of different camera angles at sports events, and the real-time provision of information about the event, all selected by the viewer.

In order to provide more sophisticated client-side applications it is necessary to find efficient ways of handling the programmatic and display rendering services in the STB. This has led to the development of STB middleware such as OpenTV. These protocols reduce the bandwidth requirements and the processing and memory requirements in the STB.

This method is mainly used where there are restrictions on the use of a reverse channel, for instance where the STB needs to make a dial-up call on the users telephone service. This is currently the case with the satellite and terrestrial broadcast of digital television. The applications operate mainly within the STB, and only require connection to the servers when a transaction is required such as completing an e-commerce (shopping) transaction.

The number and scope of applications that can be offered in the service are restricted by the broadcast bandwidth that can be allocated.

Server-side Applications

Client-side applications will always be restricted by the bandwidth requirements and by the specific characteristics of the middleware. In particular these characteristics make it difficult to provide access to services such as the Internet World Wide Web. It also makes demands on the content providers to conform to a specific set of applications that are very different from their Internet activities.

The answer to this is to provide a fully integrated set of server-side applications and functions through a bi-directional communications path. This "secondary" communications path will be called the return path for the remainder of this paper. The return path could be through a dial-up telephone connection, although this is problematical for a television environment. Whilst users accept the need to connect a PC through a modem on their telephone service, it is less acceptable for a television service. This is one major advantage that a cable system has over the other service providers; the return path can be provided over the same cable that is providing the broadcast service.

Using such a return path, a full interactive environment can be added to the digital television system as shown in Figure 10. This system can provide a full service as would be expected from an ISP; e-mail, hosted applications and services, and a portal to the Internet. The main requirement is that the STB can support an HTML browser.

This type of system has many advantages over a client-side only system, and can still provide the same class of client-side applications if required. Where there is a permanent connection to the Interactive Service System it is possible to have a richer set of functions in support of the user. In particular, the service can support a number of personalised services to a number of named individuals in the household. This is achieved by a close integration with the SMS.

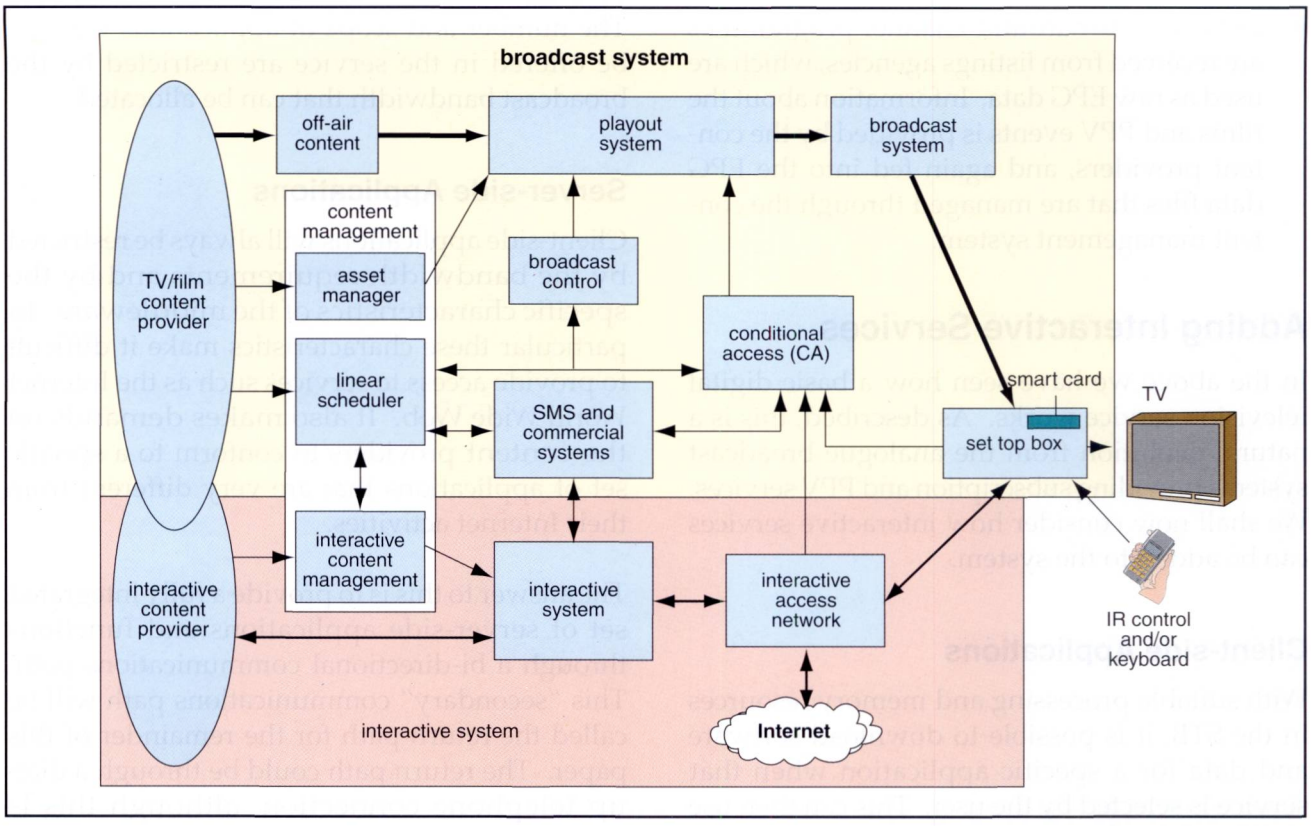


Figure 10: Adding Server-side Interactive Services to a Digital Television Service

The Interactive System is essentially a combination of an ISP service and a Web server. It also has the ability to connect to the Internet to provide access to the World Wide Web. The main difference for a content provider is that for the best display on the television screen, their HTML content may need to be re-purposed. This can be done on-line in real time by content re-purposing machines, but at the moment these are still rather inefficient. In practice it is better to produce new screen templates that display the content more advantageously on a television screen. The system therefore needs to provide content management services that allow content to be re-purposed and published onto the content servers, together with direct communications with content providers sites to allow real time loading of time critical data.

The Return Path

Even in the case of the basic digital television service and client-side interactive applications, there is a need for a return communications path from the STB for infrequent data communications. In satellite and terrestrial systems, this is provided by dial-up connections over the user's telephone connection, using a modem in the STB. As already stated, it is possible to make infrequent use of this type of connection for essential transfer of data, and the completion of transactions such as for home shopping and home banking. However, it is not a preferred method for connection to a full server-side application environment in an interactive digital television system, unless a second telephone line connection is available in the home. Where such a connection is made, it is managed in the same way as by any other ISP.

This is a major advantage of the cable system. The cable can provide a permanent connection over a different set of frequencies, typically 2MHz channels in the range 5 to 130MHz. There are a number of methods in use for the purpose, but all can be considered as providing a LAN type connection for a community of STBs. The number of STBs supported on a segment of this network is determined by the activity level of the users, and the bandwidth of local connection. Typically the LAN connection provides 3Mb/s for both downstream to STB, and upstream to the interactive system. The activity level of real users in interactive digital television systems is still the

big unknown, and will be the subject of much measurement in the early deployment of interactive digital television systems.

The Additional Requirements for the Set Top Box

Whether the STB is supporting client-side or server-side interactive applications, there are significant requirements over and above those required for a basic digital television system. This is in both processing power and memory requirements, and represents a cost premium for the support of interactive applications. However, this is already becoming less of an issue with the continuing technological cycle of increasing capability and reducing costs.

The interactive applications also demand a sophisticated middleware solution, which, together with the EPG application, have been one of the most difficult parts of the system to implement, and the cause of most of the delays in deploying interactive digital television systems.

Where there is a restricted return path and the need to deliver virtually all data over the broadcast streams, designers have looked to use efficient middleware solutions such as OpenTV. However, this requires special applications to be written, and content providers must work within these templates. Where the return path is always open, such as in cable systems, the trend is to use middleware that handles HTML formatted content. The STB provides a basic browser functionality. This facilitates the use of existing content from suppliers who run their own Web sites, and the ability for the STB to access WWW content on the Internet.

Some of the interactive services are difficult to use with a simple remote control, particularly if it is necessary to input text. A keyboard function is required, particularly for applications such as e-mail. Whilst it is possible to use "on screen" virtual keyboards, or multiple key presses such as are used on mobile telephones, these are no substitute for a real keyboard. We are therefore seeing the emergence of IR linked remote keyboards in some interactive digital television services, synergistic with the normal habits of watching television, where the viewer is usually some 2–3 metres away from the television screen.

Managing Content in the System

The system as described has separate content management systems for the linear content typical in a broadcast environment, and the interactive content typical of an Internet Web service. However, it is also necessary to consider how to integrate these systems together to provide a range of mixed systems. Some of the situations that will need to be managed are discussed below.

- The ability to provide enhanced television services, where the television service may include an application to manage a number of different video channels (for instance different camera angles at a sports event), or include access to additional information, or a link to a Web site.

  This requires "triggers" to be inserted in the MPEG-2 streams that provide prompts to the end user, as well as details of how to locate the additional data. These triggers need to be synchronised with the video stream and, if the data is broadcast, this must also be synchronised.

  Some of the synchronisation issues are relaxed if the data can be delivered over the return path from an interactive server. In this case it is necessary that the triggers can be interpreted by the STB to where the data can be found.

  If the data is accessed through the interactive servers, then co-operative processes are necessary between the two content management systems. Currently this is managed by manual processes, but in future we can expect some of these processes to be automated.

  Enhanced TV services will be produced by a number of studios, for delivery by a number of digital TV service operators. It is therefore necessary to have standards. Currently there are groups in ATVEF (Advanced Television Enhancement Forum) and DVB working on the necessary standards, but until they have completed their work, a number of proprietary systems are being used by the service operators.

- The ability to manage the insertion of advertisements, whether into the broadcast video streams, or through banners in the interactive services.

  The insertion of advertisements into video streams is handled in much the same way as enhanced TV. Banner advertisements are normally delivered by an advertising server in the interactive system, using standard Web technology. These advertising servers are capable of delivering banner advertisements according to configurable business rules, related both to the rate of delivery and to whom they should be delivered.

- The ability to access streaming content such as video or audio from an interactive service. This is an enhancement of interactive content, where the enhanced content is delivered through the broadcast streams. It is the inverse of enhanced TV from a content management perspective.

In all cases, decisions must be taken as to whether the content should be delivered by broadcast streams, or accessed from the interactive servers. This is a complex question which is dependent on the availability of broadcast bandwidth, the bandwidth in the return path, and the caching capabilities of the STB. In general, the interactive servers give access to more data in an application but, if a small number of screens are very regularly accessed, it may be worth delivering the content over the broadcast streams.

In those systems using a dial-up return path, the general principle is to use only the return path when necessary. This is because it probably uses the end user's only telephone line. Apart from clashes in use, it also involves telephone charges to use the return path, and has an initial latency associated with dial up and authentication. This leads to virtually all content being delivered in the broadcast streams.

The management of interactive content will depend upon the middleware supported by the STB. For the class of middleware (such as OpenTV) that is orientated towards the efficient delivery of content data over the broadcast streams, it is likely that the content provider will have to enter material into specially commissioned applications. Such material is unlikely to be obtainable directly from HTML content on a Website.

Where the STB provides an HTML engine and browser, it is possible to make direct use of content produced for a Website. However, the TV screen has different display characteristics to a PC monitor for which most Web sites are designed. It has lower resolution, less active area on the screen and different colour characteristics. Once interactive digital TV has become established, I am sure that content providers will design their applications to be effective in both environments. Until then, for key services, it is likely that the screen templates will be redesigned for the TV environment.

A number of suppliers have designed repurposing proxies, which are intended to provide real-time transformation of HTML content for more effective display on a TV screen. However, these systems are still somewhat limited in their capability, and are computationally very intense. However, they do have a place in situations where efficient caching is possible.

Finally the system must provide all the tools found in the content management systems of any large interactive service provider hosting content, such as AOL, MSN, etc.. There must be facilities for the off-line delivery of content to the service, together with authoring, editorial control, staging, and delivery of content to the live servers. It is also necessary to provide live data feeds where real-time updates of data content are required such as in news and sports services. Many of these concepts were covered in [EDWARDS and STEELE, 1999].

Managing the Subscriber in the System

Apart from the ability to receive "free to air" television programming from terrestrial digital television broadcasts, all other services will be provided on a subscription and pay per view basis. This requires the end user to enter into an agreement with the service provider, who will then bill the customer on a monthly basis. This is the role of the SMS.

In the case of the satellite and cable operators, they have existing Subscriber Management Systems that support their current services. A key

systems design requirement has been to use this legacy system to manage the digital television subscribers. This will allow the service operators to manage the transition from analogue to digital services.

The Subscriber Management System

As a minimum, the SMS must provide the following:

- Setting up an account for a new subscriber on receipt of an order
- Managing the installation of the STB, and the provisioning of service entitlements in the STB
- A billing system that will handle subscriptions and any charges for discretionary use of the services, such as PPV events and interactive services that may be charged on a metered basis
- Managing the account on an ongoing basis. This will include changes to the service as requested by the subscriber, handling the payments, credit control, etc.
- A customer service desk for subscribers to call. This help desk not only answers general queries, but also handles many aspects of account management. For subscribers, this is the human face of the service, and it is essential for the service operators to get this right. A key business metric for all service operators is the percentage of subscribers they lose. A good customer service desk can help minimise subscriber churn
- Financial systems, including reconciliation with content providers.

In order to achieve the above, it is necessary for the SMS to have full details of all the "products" and services, including their price. This is used not only to manage the entitlements of an account, but also to provide the information to the billing system for the charging of discretionary services.

Where the service provider supplies the STB as part of the contract, the SMS will also provide an asset control system.

The delivery systems are capable of collecting a large amount of data. As a minimum there will be general data about the use of the system but, increasingly, data will be included about individual user activity in the system. It will also be possible for the system to collect information about users through interactive dialogue within an application. The main uses of this data will be:

- Financial—providing the data required to bill subscribers for discretionary spend, and to handle the commercial and financial arrangements with content providers
- Operational—providing information on the data traffic in the networks, and the use being made of the equipment. This is an essential tool for assessing performance and service level agreements, and managing capacity planning in a timely manner
- Marketing—providing information on the use of services that can be used for service planning, and eventually providing detailed end user profiles. Information about the end users will be used for many marketing purposes
- Personalisation—the use of data to provide personalisation of the services will be a growing trend, and is covered in more detail below.

Subscriber Facilities

In the original analogue services, subscribers had few facilities at their disposal. Any service queries had to be handled in person, by post, or by telephoning the help desk. However, with the introduction of interactive digital television, many more facilities will be available through the service itself. Examples of the facilities offered by the system are:

- Tutorials and Help—the tutorial is normally aimed at first time users and delivered as a broadcast stream. Help is provided as reference information which can be accessed at any time. This can be provided within a broadcast stream, but is more effective if delivered as an interactive service, particularly if access to the files is context sensitive
- Electronic Program Guide
- Managing multiple users of the STB. Some systems allow a number of users within an account to be identified in the system. This not only provides for a more personalised system, but allows the account holder to apply some control over the activities of the other users. Examples are parental control and budgetary control over discretionary spend

- Parental control over the use of the system
- Self-provisioning, where the user can request and/or activate a change of service from an on-line application
- On-line billing, where the account holder is informed through the television screen that the bill is ready and can be displayed on demand. Provision can also be made for the account holder to pay the bill by using a credit or debit card if a direct debit mandate has not been agreed.

All these facilities are aimed at improving the service as perceived by subscribers, and have the added benefit of reducing the load on the customer service desk. Whilst the customer service desk is still required, it is expensive for the service providers to run it and reducing the demand helps reduce costs.

The System Data Object Model

The system enhancements described above require a more sophisticated data model than that used in the analogue Subscriber Management Systems. Because of the necessity to maintain the SMS legacy, additional facilities have to be provided by the interactive system and client-side applications in the STB. In general the interactive environment must mirror part of the SMS data model, and then be extended to provide the additional facilities associated with the registered users on an account. This can be complicated by having more than one STB on an account, and having the need to track the products and entitlements to support functions, such as self-provisioning. This requires that the interactive system be built around a database which encompasses the required system data model. This is exemplified by the membership server in Microsoft's MCIS (Microsoft Commercial Internet Server), which is deployed in a number of systems in the UK, although Oracle also has an equivalent system deployed in the satellite system.

Personalising the Service

With the extended facilities being offered by interactive digital television, it is possible to begin to "personalise" the service. When entering some of the services, the users will be requested to identify themselves against the list of registered users for that account. Where necessary the identities will be authenticated with PIN numbers. Knowing the identity not only allows controls to be applied such as parental controls, but also allows the interactive services to communicate with an individual known in the system database.

Initially, the systems are likely to be simple but, over time, they will be able to accumulate information about the users. This information can then be used by applications to deliver more appropriate content, or to make suggestions about content from the wealth available, which may otherwise be difficult for the user to locate.

The Future

This paper has concentrated on describing the basic technologies, and general systems principles being deployed for the first generation of interactive digital television services. However, with the experience gained from the operation of these early systems, and the ability to deploy even more advanced technology, it is expected that the systems (and services they can deliver) will develop rapidly over the next few years. In particular, the following can be expected.

- More efficient use of broadcast bandwidth by using 256QAM modulation (and above), and the availability of more multiplexes as the analogue channels are turned off. This additional bandwidth will be used to supply more content, not necessarily more television channels, but more enhanced TV channels or more choice of pay per view events, etc..

- The introduction of true Video On Demand (VOD) services, particularly on the cable networks which can handle the real time control of VCR-like functions of start, stop, pause, rewind, and fast-forward.

- A rich source of enhanced television services will appear, incorporating increasing levels of interactive content. Some of the key genres exploiting the system are likely to be sport, entertainment, and education.

- The emergence of a more interactive service, where the user can control many aspects of the overall service.

- Increasing use of the interactive services for access to information, electronic commerce, etc..

- The emergence of more integrated systems. In particular the integration of the broadcast and interactive systems, making more use of the return path to control the STB rather than depending on the traditional broadcast methods. This will provide the feedback from the STB that is missing from the current broadcast systems, making for a more efficient system. This will require a rethinking of the conditional access systems and the acceptance of the media studios.

- The only other return path technologies that provide the immediate on-line capability of the cable system are ISDN and ADSL. They are also both independent of the customers' telephone service. However, it still remains to be seen whether these technologies can be deployed at a cost commensurate with a mass consumer market.

Conclusions

This paper has covered the general principles behind any interactive digital television system. However, there are some significant differences between the different broadcast systems (cable, satellite, terrestrial) and their associated return paths. In the UK, the satellite service, and to a lesser extent the terrestrial service, has stolen a march on the cable companies, and is making a significant market penetration. But with all three systems eventually being available to most homes, there is likely to be some confusion in the market. What happens next will very much depend on consumer education and the popularity of interactive services to a television audience. There is no doubt that for the delivery of enhanced and interactive television the cable systems have an inbuilt advantage, but it remains to be seen whether they can exploit this advantage in the market place.

Glossary

CA(S)	Conditional Access (System)
DSM-CC	Digital Storage Media Command and Control
DVB	Digital Video Broadcasting
ECM	Entitlement Control Message
EMM	Entitlement Management Message
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
HTML	Hyper Text Mark-up Language
IP	Internet Protocol
IR	Infra Red
ISP	Internet Service Provider
MPEG	Moving Picture Experts Group
NVOD	Near Video on Demand
PID	Packet Identifier
PPV	Pay Per View
PSI	Program Specific Information
QAM	Quadrature Amplitude Modulation
SAS	Subscriber Authorisation System
SI	Service Information
SMS	Subscriber Management System
STB	Set Top Box
VOD	Video on Demand

Bibliography

A 020, "A Guideline for the use of DVB Specifications and Standards" DVB Blue Book A 020, February 1997.

DYER, N., "An Architecture for Commercial On-line Internet Services," ICL Systems Journal, Autumn 1998.

EDWARDS, J., STEELE, H., "Monterey. A Web Content Production System," ICL Systems Journal, Spring 1999.

EN 300 468, "Digital Video Broadcasting; Specification for Service Information (SI) in DVB Systems" Draft standard EN 300 468, European Telecommunications Standards Institute, September 1997.

EN 301 192, "Digital Video Broadcasting (DVB); DVB Specification for Data Broadcasting" Draft standard EN 301 192, European Telecommunications Standards Institute, August 1997.

ETR 154, "Digital Video Broadcasting (DVB); Implementation Guidelines for the Use of

MPEG-2 Systems, Video and Audio in Satellite and Cable Broadcasting Applications" ETSI Technical Report ETR 154, European Telecommunications Standards Institute, September 1996.

ISO 13818-1, "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information. Part 1: Systems", ISO/IEC International Standard IS 13818, November 1994.

ISO 13818-6, "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information. Part 6: Extension for Digital Storage Media Command and Control", ISO/IEC International Standard IS 13818, November 1994.

ATVEF, Various documents can be found at http://www.atvef.com.

Biography

John V Panter is an ICL Fellow Emeritus, having taken early retirement from ICL in 1998.

He joined ICL through Leo Computers in 1961, having gained a BSc Hons in Physics from Bristol University. Working mainly in Kidsgrove and Bracknell, he held a number of senior technical and managerial positions, usually at the leading edge of technology and the formation of new business enterprises. His interest has always been the exploitation of technology for business benefit.

Following up a personal interest in multimedia, he was a founder member of Project VIEW, the corporate task force for business strategies for "multimedia". As this developed into a new business start-up, ICL Interactive, he took the role of Technical Director, responsible for the design and implementation of systems for content management and the delivery of interactive services over a variety of network topologies. He held this position until his retirement.

Since his "retirement" he has been a consultant to a major cable company, working with a team on the design and implementation of an interactive digital television system.

Internet Shopping Services

Stephen F. Picken

ICL, Kidsgrove, UK

Abstract

This paper is an overview of an approach that ICL is taking in order to deliver Internet shopping services using Microsoft Site Server 3.0 Commerce Edition. The influencing factors and the business drivers to the approach are also presented. The approach is derived from the requirements for reducing cost of development, reducing time to market, increasing flexibility of the user interface and use of industry standard components. The resultant technical architecture is one that is flexible enough to meet the needs of a variety of Internet shopping scenarios, from single shop Web sites to shopping malls. Central to the approach is the belief that an Internet shopping solution should be delivered to the customer as an on-going service, rather than a one-off product. Furthermore, the solution must be able to respond to a particularly dynamic environment, in which the demand of both business and shoppers for Internet shopping services are becoming more sophisticated all the time.

Introduction

ICL has been building Internet shopping sites since 1995, and in that time has provided services to prestigious clients such as the BBC, the Arcadia Group, Macmillan Publishers, Rolls-Royce, Woolworths, Waitrose and Northcliffe Publishers. ICL's partners in delivering Internet shopping services include NatWest Bank for payment services and Microsoft for the technology platform. A great deal has been learnt about what makes a good Internet shopping site. This paper draws on some of these experiences, gained from building sites using a variety of platform technologies, to present an overview of how to provide Internet shopping services using Microsoft Site Server 3.0 Commerce Edition.

Businesses are turning to the Web as an alternative sales channel for their goods and services. These businesses can choose from a bewildering array of inexpensive 'off-the-shelf' products that enable a business to build its own commerce enabled Web site from pre-built 'starter shops'. The initial cost of the licence for such products is often low in comparison to the total cost of ownership of the Internet shop. The total cost of ownership includes the design of the look and feel, integration into the existing business processes, deployment to a live service and on-going management of the site.

The fast moving nature of the electronic commerce business and the speed of change means that many businesses cannot afford the effort needed to keep pace with the latest developments, or to provide on-going maintenance of a Web site. The solution is to purchase a service that includes the specification, implementation, deployment, hosting and management of the site. The total cost of ownership will be more transparent as it should be defined in the agreement with the service provider.

Thanks to its highly interactive nature, the Web provides new ways of engaging the customer that are not possible with other media. A successful Internet shop exploits this advantage. For example, Microsoft Site Server 3.0 Commerce Edition has features that enable a well-constructed Internet shop to suggest another product that the customer may be interested in purchasing, based on the contents of their shopping basket and the purchase history of other shoppers. However, there is a risk of exploiting Web technology for its own sake, rather than keeping the needs of the shopper as the top priority. Technical consultancy provided as part of the service assists the business in making the right technology choices for its target market, to ensure that the widest possible audience can use the site.

Many businesses that are not traditionally considered to be retailers are realising the vast potential of the Web and looking to build Internet shopping sites. Examples include publishing and media businesses, schools and colleges and government departments. Such businesses rely even more on service led solutions that include an element of consultancy as part of the solution to help them understand not only the Internet, but also how to use it to sell their products and services using the medium. The needs of these businesses are driving much of ICL's activity in the Internet shopping service business, in addition to providing a service to the larger, more established retailers.

Business requirements

This section presents the requirements that led to the approach described in this paper.

Reduction of development costs

It was clear early on in the development of bespoke Internet shopping services that there were many requirements common from one project to the next, and the development of a completely bespoke solution for each customer was unnecessarily time consuming and costly. Re-use of a core technology platform was established to reduce costs and time to market.

Reduction of time to market

Time to market is crucial to the development of Web sites in general. The speed of change in Web technology means that businesses have had to react to the new medium very quickly. The need to deliver a live Internet shop in six to twelve weeks is commonplace. The development of the group of Web sites for Northcliffe Electronic Publishing in early 1998 demonstrated that this could be achieved more readily by using Microsoft Site Server instead of ICL proprietary technology.

Use of industry standard components

Selection of a core technology platform that is established as an industry standard has several advantages. Others have already used the technology, and problems are more likely to have already been identified. A good support network is also likely to be available both from the product supplier and also the wider development community.

The platform chosen for the development of Internet shops also needed to provide functionality to support a variety of other Web services. This is important because many Web sites offer much more than simply Internet shops; they offer other services such as news stories, chat areas and classified advertising services. An example is the group of Web sites developed for Northcliffe Electronic Publishing, referred to earlier.

Application Flexibility

Internet shopping services offered by ICL range from the provision of single Internet shops, such as the BBC Shop, through Internet shopping malls such as Buckingham Gate, to Internet business parks such as TradeUK. This diversity in customer needs led to the requirement for a standard technical platform able to support a variety of Internet shopping based applications.

Potential for Change

The need for Internet shopping Web sites to grow was identified, as it was thought that many businesses would want to join a shopping mall as an experiment, and move into their own Web site once they were established. This led to the requirement that the architecture should be designed so that a retailer could move easily and quickly from a mall based service into his own service, without having to carry out large-scale modifications to the shop.

What makes a good shopping site?

A good shopping site is one which meets, or exceeds, its sales forecast.

The building of such a site needs to take into account a variety of perspectives, including those of the shopper, the shop manager and the service provider, all of whom will view the problem differently. The provision of service may actually require a number of individual service providers, each offering particular specialisms. For example, Internet shopping Web sites are likely to need the expertise of business consultants, application developers, graphical designers and Internet

service hosts. Each of these groups will view the service from a different perspective and any potential conflict between them must be considered and managed appropriately when constructing particular solutions for customers.

The following sections consider each of the major perspectives on the Internet shopping service, making reference to lessons learned in the development of both bespoke Internet shopping sites and a generic Internet shopping service.

The shopper's perspective

The shopper is primarily focused on the public Web interfaces of the shopping site. The shopper is interested in:

- a compelling reason for visiting the site in the first place. The site must be easy to find and have something of interest once the shopper is there
- a means of browsing the product catalogue, or viewing specific products
- a means of directly locating specific products and information, and ensuring their availability
- a means of making a purchase securely and with confidence
- an opportunity to provide feedback to the shop manager and/or the site host manager
- a simple and intuitive user interface.

Theme and context encourage shoppers

A good theme for a Web site increases the chances of the shopper discovering the site. This is achieved by the use of search keywords that relate to the theme, which enable search engines to match the site when the shopper runs a search. A startling example of this was the 350% increase in traffic on the Buckingham Gate Web site shortly after the tragic death of Princess Diana in September 1997. The underlying reason for this behaviour was that search engines matched the Buckingham Gate site when users searched for phrases such as 'Buckingham Palace'.

It has also been established that the 'Best of British' theme of Buckingham Gate attracts regular visitors to the site to read the editorials on high society events in the United Kingdom, such as the Henley Regatta and Wimbledon fortnight.

Having attracted the shopper to the site with a theme, and theme-based content, a good Internet shop integrates product links into that content. This is called context selling. Buckingham Gate is integrating attractive editorial content with products and services that are also associated with the 'Best of British' theme, from retailers such as Rolls-Royce, Wedgwood and Churchs' Shoes.

Provide early opportunities to buy

A good Internet shopping site provides opportunities to buy as soon as any product information is displayed on screen. It is a common practice to display information about products in two stages as follows:

- The first displays several products together as a summary page, perhaps showing five to ten products
- The second displays more detailed information for a specific product.

Analysis of the use of shopping sites hosted by ICL indicates that for Web sites that are built using this method, the product summary page is likely to be requested much more often than the product detail page. A good Internet shopping service allows the shopper to buy the product from the summary page, even if the shopper must supply further information, for example selection of colour or size, before the product can be added to the basket.

Provide a good product search facility

A poor search facility loses sales. A good search facility within the Internet shop is, therefore, important, as it enables the shoppers to find the products that interest them quickly and easily. Analysis of Web site access logs shows that when a search does not return the expected results, the shopper does not take the browsing session any further.

A good Internet shopping site offers the shopper a search method that is flexible and powerful enough to cope with a wide variety of query types, including the ability to perform wild card searches and word stemming searches. In general, such features are not found in a system that employs a direct database query for its search mechanism. A dedicated search facility, such as the Microsoft Search Server, offered as part of

Microsoft Site Server 3.0 Commerce Edition, provides a more flexible search solution.

Although the Microsoft Search Server is a part of Microsoft Site Server 3.0 Commerce Edition, the 'starter shop' applications provided with the product do not use it. They use a simple database query instead. Such an implementation is likely to be suited only to small Internet shops that have a limited product range. The commerce and search components should be used together to deliver a search mechanism that is suited to larger Internet shops.

Fulfil expectations about the product range

A good Internet shop will fulfil the expectations of the shopper. A well-known physical brand is likely to set expectations about the range of products that should be offered by the Internet shop. The shopper is likely to expect to find at least those products that are available in the High Street, and will hope to find products that cannot be obtained locally. After all, the shop is not constrained by physical limitations imposed by real shops.

Although the Internet shop may not be constrained by physical storage space, other restrictions usually apply to the presentation of the product catalogue. For example, limited availability of the product collateral or a very short time in which to construct and publish a new site.

Anecdotal evidence suggests that unfulfilled expectations of the product range are a major cause of Internet shoppers complaining to the Web site. This not only loses sales, but employing someone to respond to the complaints increases costs.

A good Internet shopping site offers the shopper some means of browsing and purchasing the full product range, even those products for which there is limited collateral available to view online. As long as enough information is available to make a sale, the shopper should not be stopped from purchasing due to constraints imposed by the site design. For example if a product is out of stock, the shopper should be left with a positive message that the Internet shop is normally able to supply the product, although it is not for sale at that point in time. A good site may further inform the shopper of when the product is likely to be available again.

Keep Web page design simple

An elaborate design, perhaps one that includes lots of images or that requires the shopper to be running the latest version of his browser, means that the Web site appeals only to those shoppers who have a fast connection on to the Internet and are able to keep pace with the latest browser release. This leads to further loss of sales as it limits the potential audience.

A good Internet shopping service keeps the needs of the shopper in mind from initial design through to on-going hosting and management. This can be difficult, as there may be millions of shoppers spread around the world. Therefore, the recommended approach is to keep the user interfaces simple, making judicious use of browser features and HTML language extensions, so that the widest audience possible can use it.

A good Internet shop provides user interfaces which inspire confidence in the shopper, not only by keeping things simple and by using an intuitive page navigation structure, but also by ensuring that any expected brand imagery associated with a particular shop is part of that interface. Using recognised brand imagery enables the shopper to identify with the site and builds on trust that has already been established in the real world.

Use standard practice

Standard practice improves the shopping experience. A large number of Internet shopping sites are establishing many common practices and common metaphors. It is, therefore, reasonable to assume that Internet shoppers are coming to expect certain standard features of an Internet shopping Web site to behave consistently regardless of the Web site publisher.

The shopping basket is an example of this. Many web sites now include a virtual shopping basket facility; the shopper sees something that he likes and clicks the 'Add to basket' button next to the product. The virtual shopping basket allows the shoppers to view all the items in their basket, change quantities and remove items.

A good Internet shopping service adopts standard practices whenever possible. Certain practices, such as those required by law, must be adopted. These include information standards, privacy standards and currency standards.

The shop manager's perspective

Shop managers are primarily interested in providing the best possible service to their customers, thereby ensuring that the most sales are achieved and that the brand image of the shop is enhanced. The shop manager's primary responsibilities in the on-going management of the shop include:

- Upload and publication of the product catalogue
- Upload and publication of editorial content for the web site
- Collection of orders placed on the site
- Review and response to shoppers' feedback on the site
- Review of shopper behaviour on the site.

A good Internet shopping service offers interfaces to support at least all of the above activities. The next section illustrates some examples of good practice in the development and operation of Internet shopping Web sites, from the perspective of the shop manager.

Simplicity of Web page design

Elaborate designs restrict the product range. As in many aspects of service delivery, the designer of pages for an Internet shopping service must strike a balance between the needs of the shopper and the needs of the shop manager's business. Often the business has short term cost considerations that may affect the look and feel of the page, and they must be taken into account and planned across the lifetime of the Internet shopping service.

To illustrate this, consider a business that is building an Internet shop. It is very tempting to focus on the shopper's Web interface, and build a beautiful looking Web site with graphical images that, for example, portray products from a variety of angles. This is a shopper's delight, in which a desire for sensory experience can be satisfied. It looks great on paper too. The page designs work well when the sample product is displayed from several angles, with additional images to highlight details on the product.

From the shop manager's perspective, this might not be such a good design as it would first appear, since the cost of producing such collateral for the complete product catalogue may be prohibitive. Even if cost is not an issue, the production of such imagery can be time consuming to generate, and not available when the rest of the Web site is ready to be published.

A good Internet shopping service offers a balanced approach, with a page design that is capable of showing detailed images, but which still works when none is available for a particular product.

When launching a new Internet shopping site it is often appropriate to spend most of the budget available for product images on a small proportion of the product catalogue, thereby providing high quality images for the most popular products in the catalogue. Images for other products can be added as the Web site matures. A good Internet shopping service supports this practice, while still being able to deliver Web pages that are visually appealing as well as enabling purchases from the complete product catalogue.

Provide content management and preview facilities

The shop manager, who may not be technically skilled, must be able to preview the content of the Internet shop before it goes live to the public. This will be both editorial content, such as the shop home page message of the day, as well as catalogue content, such as product descriptions and department hierarchy.

A good Internet shopping service includes a content management aspect that enables the shop manager to control the following facilities:

- Introduction of content into the Internet shopping site
- Modifying and deleting content
- Management of the approval of content prior to publication
- Preview of content prior to publication, including preview of changes to published content
- Archiving and/or purging of redundant and obsolete content.

Separate the look and feel from the business logic

The look and feel, or the shopper interface, of a Web site is usually updated regularly, in order to keep the Web site looking fresh. A good Internet

shopping service allows the shop manager to achieve this without having to modify:

- the business logic that makes up the application
- the business data that is displayed
- the overall structure of the site, which would lead to changing the familiar user experience that has been built up.

A good Internet shopping site achieves this by separating the look and feel of the site from the business logic and the business data, and by providing the shop manager with a method of updating the look and feel.

Service delivery perspective

The service delivery perspective is interested in the efficient production, delivery and on-going management of the Internet shopping service. More than one organisation is often involved in the delivery of the complete service, since there is a wide variety of skills required. The service deliverer is responsible for the following functions:

- Definition of the specific requirements for the Web site
- Implementation of the application
- Design of the look and feel, or the user interface, for the Web site
- Deployment of the application
- Hosting and on-going support of the Web site
- Card payment services
- Fulfilment and delivery services.

Provide support for managing the Web service within the application

Downtime of an Internet shopping service loses sales and an unreliable service damages customer loyalty. A good Internet shopping application provides interfaces to enable diagnostic tools to interrogate the operational state of the service. This facility is needed in order to determine whether there are problems with any aspects of the service, and to alert the service manager to them quickly quickly.

The Internet shopping sites are integrated with ICL's own WebCheck availability program. WebCheck constantly monitors special pages from the Web server across the Internet, and analyses the results to determine the state of various parts of the Internet shopping system.

If there is a problem, the WebCheck program will contact the service manager, using a mobile phone or mobile pager, to alert him to the problem.

Provide tools that support the graphic design team

Tools that support the graphic design team reduce costs. The graphic design team is responsible for designing the look and feel of the Web site, by agreeing the design brief with the customer and producing a prototype of the user interface.

The graphic design team uses HTML files to present the user interface once the original design ideas have been sketched out and agreed with the customer. The HTML files must then be integrated with the Web application. This involves converting the HTML files into HTML templates. HTML templates capture the look and feel in a way that enables them to be integrated with the functional aspects of the site.

The HTML author is provided with the tools that enable the HTML files to be written directly as templates that can then be integrated with the application with minimum effort. This means that the application developer can concentrate on capturing the functional aspects of the Internet shopping site, while the HTML author can concentrate on capturing the look and feel.

Use audit logs to log system events

When things go wrong, the cause of the problem must be found quickly, especially when there are users of the system, from both the service delivery and shop management perspectives, who have access to facilities that could cause problems. A good Internet shopping site maintains audit logs of activities carried out during maintenance of the site. These logs include the identity of the person who carried out the task, as well as date and time stamps.

The recommended approach is to provide a service whereby the day-to-day administration of the Internet shopping site takes place using a set of manager Web pages. These pages are not available to the ordinary shopper; indeed, they will typically be located on a separate secure Web

server to which the public does not have access. The system is implemented using a privilege based administration scheme, linked to the role to which a person is allocated. The system attributes actions to the person who took them and writes actions to an audit log.

Use Microsoft development tools for a Microsoft Environment

To build a good Internet shopping application the developer needs the right tools for the job.

Microsoft Visual Interdev 6.0 and Microsoft SourceSafe should be used to build Internet shopping applications using Microsoft Site Server 3.0 Commerce Edition. These tools offer an integrated environment for large scale and multi-team developments and integrate seamlessly, enabling the development team to carry out client and server side debugging from the same interface.

ICL is working closely with Microsoft to ensure that the next generation of Microsoft products continue to meet current and future needs.

The approach to providing an Internet shopping service

The approach to providing an Internet shopping service has been:

- to understand the business requirements and the shopper's habits by working with customers on bespoke projects
- to analyse the trends in Internet shopping solutions that it and other companies have built in the past
- to determine the strengths and weaknesses of the bespoke solutions, and build upon the strengths.

In doing this, valuable insights into how an Internet shopping service should be constructed and operated have been gained. This knowledge and experience enables a full Internet shopping service to be provided in the form of:

- **Consultancy**—An understanding now exists of how to sell on the Internet, and what makes a good Internet shopping site
- **Industry expertise**—ICL is a leading supplier of electronic business services, and has privileged access to products and services from its suppliers and partners
- **Web site development**—Internet shopping Web site applications are built using Microsoft Site Server 3.0 Commerce Edition and expert capability has been acquired across the range of Microsoft Enterprise products
- **Back-office integration capability**—ICL is a systems integrator with a good track record for providing back-office integration
- **Graphical design and usability expertise**—A graphical design team now exists which has become expert in producing Web sites for businesses, including Internet shopping sites
- **Hosting and management services**—ICL operates Web server farms in which to host Web sites, with expert management capability.

Solution Delivery Framework

As many of the functional characteristics of an Internet shopping site are common, a generic Internet shopping solution has been constructed which enables bespoke customer requirements to be integrated and delivered as part of the overall service. Examples of bespoke customer requirements include the design of a look and feel that makes the best use of the customer's brand image, and the back end integration into the customer's business systems.

The basic Internet shopping functionality is provided by Microsoft products and is enhanced by additional components supplied by ICL, for example:

- the themed mall concept
- staging of the product catalogue and editorial content
- the use of dynamic Web templates to separate look and feel from business logic
- the provision of an electronic payments gateway
- the ability for an installation to be administered remotely as a managed service.

This approach aims to provide approximately 80% of a customer's functional requirements. The final 20% is 'finishing work', which allows the Internet shopping site to be tailored to the customer's specific requirements. In addition to the functionality of the Web site, the service elements are added to the solution in the form of

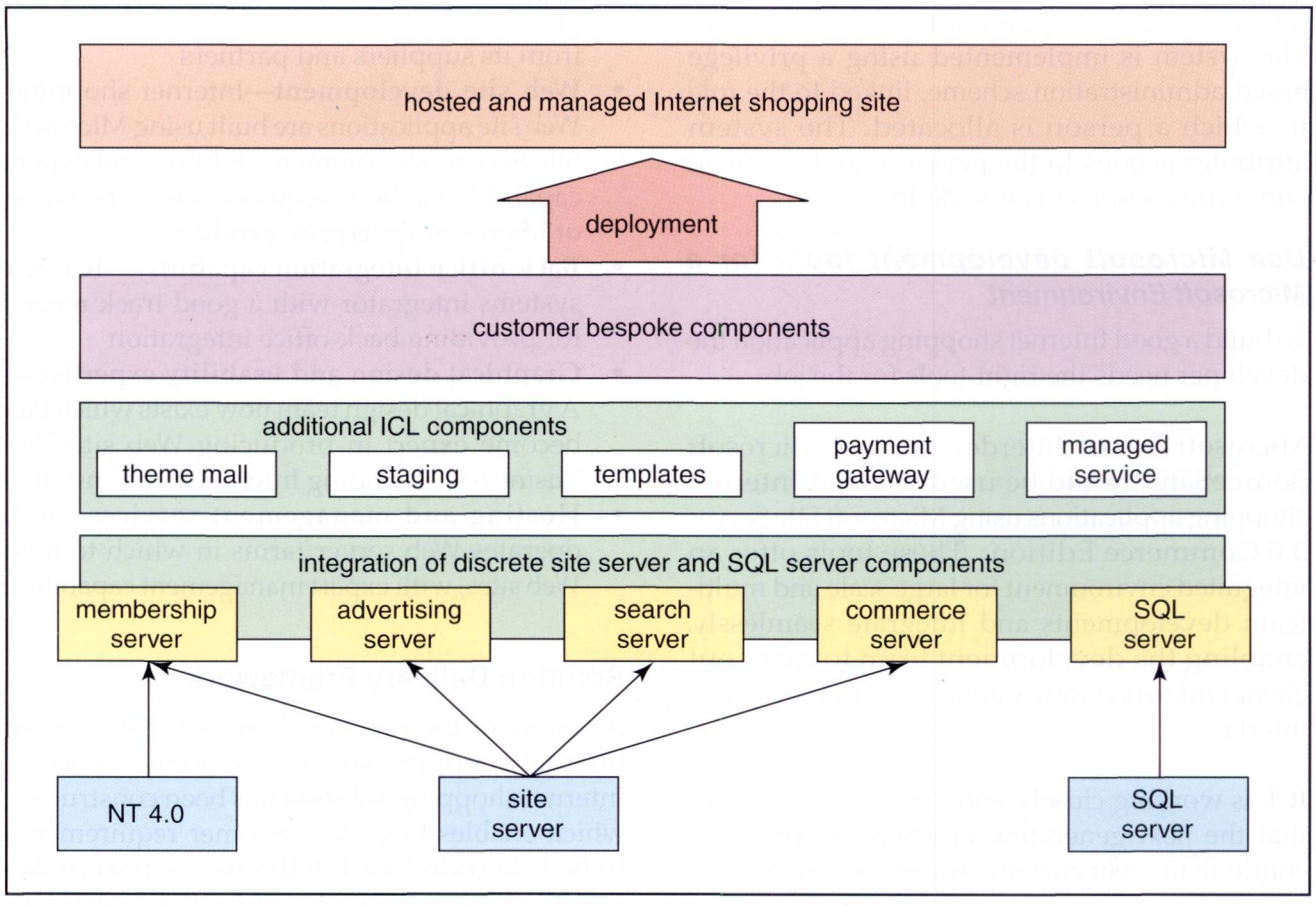


Figure 1: Solution delivery framework for an Internet shop

consultancy, hosting and ongoing management of the site. This enables the delivery of solutions with bespoke characteristics at a lower development cost and reduced time to market, which is advantageous to both the service provider and the customer alike. See Figure 1.

A strategic programme for Internet shopping

A strategic programme for developing a generic Internet shopping service is in operation, which ensures that new demands and changes in the marketplace can be met as they arise. For example, ICL is working in partnership with Microsoft to ensure that the Internet shopping service can support very large-scale operations running multiple front-end web servers. This work involves the migration of the service on to the Windows 2000 platform to ensure scalability on the next generation operating system.

The ICL and Microsoft partnership enables ICL's customers to benefit by taking advantage of a strategic programme of development which supports their own strategic planning regarding Internet based services.

Internet shopping service architecture

This section builds on earlier concepts and presents some of the more interesting technical aspects of the Internet shopping architecture, in particular where the facilities offered by Microsoft Site Server 3.0 have been significantly enhanced.

Physical Architecture

This section describes the hardware and software components of an Internet shopping system.

The current Internet Shopping architecture is based exclusively on Intel architecture Windows NT systems, using the following Microsoft products:

- Windows NT 4 Server
- SQL Server 6.5 or 7.0
- Internet Information Server (IIS) 4 web Server
- Microsoft Transaction Server (MTS)
- Site Server 3.0, Commerce Edition. In particular the following components are used by the Internet shopping application:
  - Commerce Server

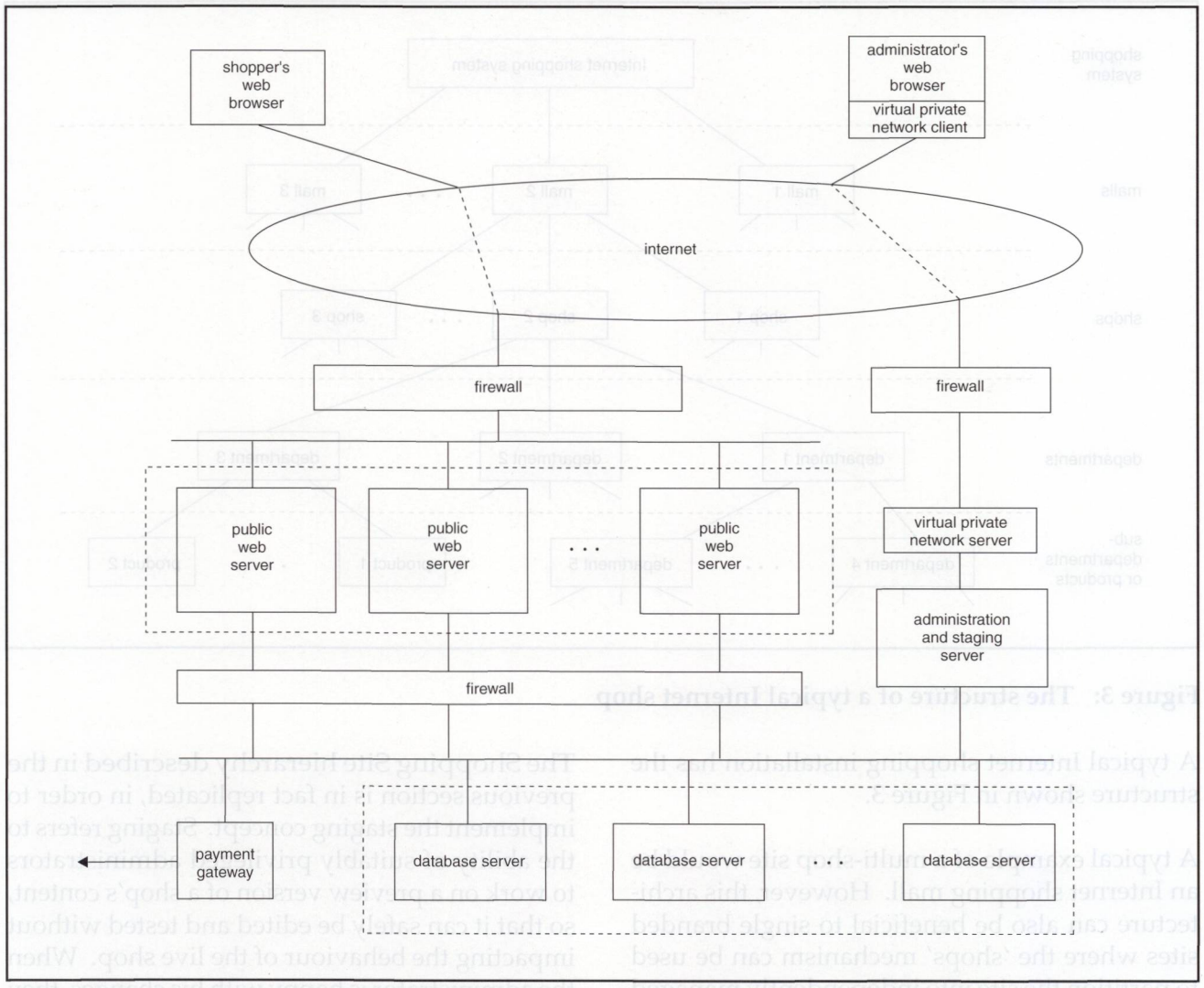


Figure 2: The physical architecture of a typical Internet shop

- Advertising Server
- Search Server (including Search Catalogue Build Server)
- Membership Server
- Analysis and Reporting Engine.

A typical hardware configuration for an Internet shopping application is shown in Figure 2. The precise components used in a particular implementation depend upon the business requirements, such as its performance, availability and security constraints. In particular, smaller implementations may combine a single database server with the administration and staging server where this offers sufficient performance.

Hierarchy of a shopping Website

The structural characteristics of the Internet shopping application were originally derived from a Commerce Server wizard generated site and, although considerably enhanced and improved, retain much of the basic structure of the Microsoft sample sites, for example, the Volcano Coffee starter shop provided with Microsoft Site Server 3.0 Commerce Edition.

One of the major enhancements made in the Internet shopping application was to extend the ubiquitous **Shop->Department->Product** hierarchy to allow related shops to be grouped into Internet shopping malls. Typically, the mall will have some kind of overall theme to draw shoppers to it. The shopper's experience when visiting such a themed mall is somewhat similar to that of a physical shopping mall, but with the advantage that a single shopping basket can be carried between all shops in the mall. Each shop within a mall can be independently managed, and can have its own look and feel if desired.

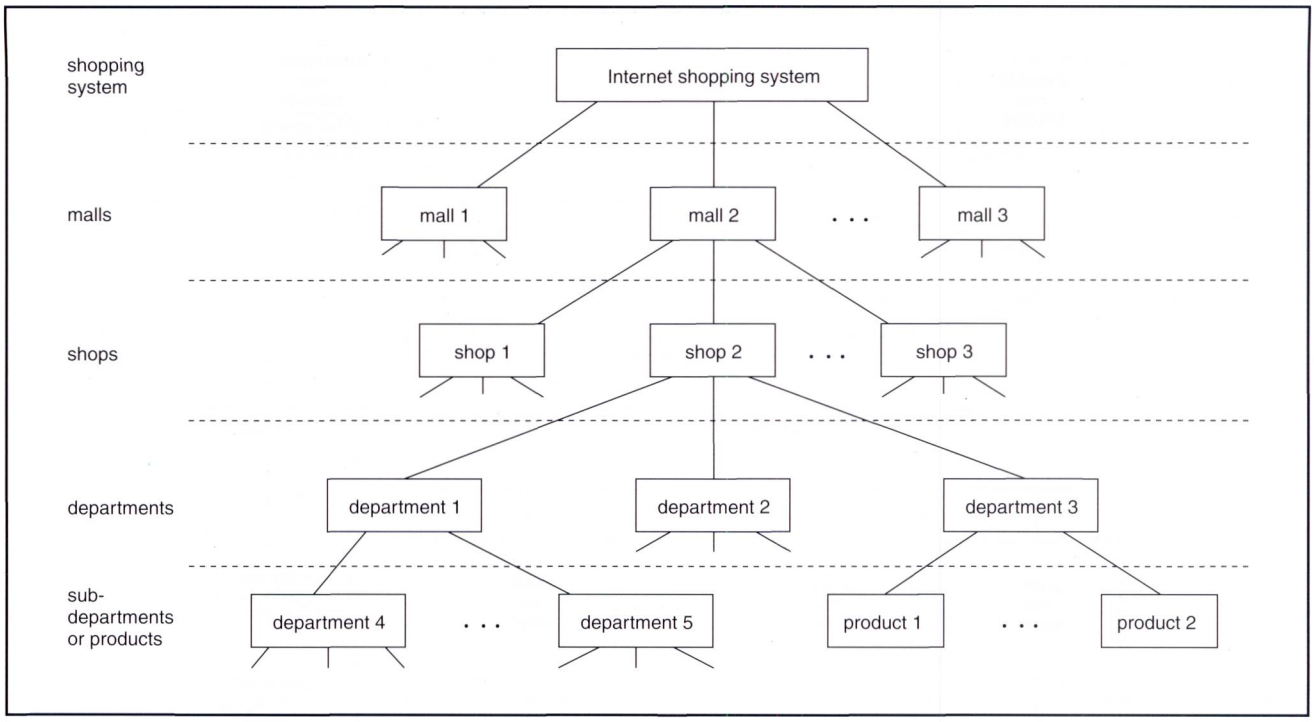


Figure 3: The structure of a typical Internet shop

A typical Internet shopping installation has the structure shown in Figure 3.

A typical example of a multi-shop site would be an Internet shopping mall. However, this architecture can also be beneficial to single branded sites where the 'shops' mechanism can be used to partition the site into independently managed areas, reflecting the operational structure of many large businesses.

Content Staging

This section describes the method, called content staging, used to enable site managers to administer their Web site without affecting the use of the site by the public.

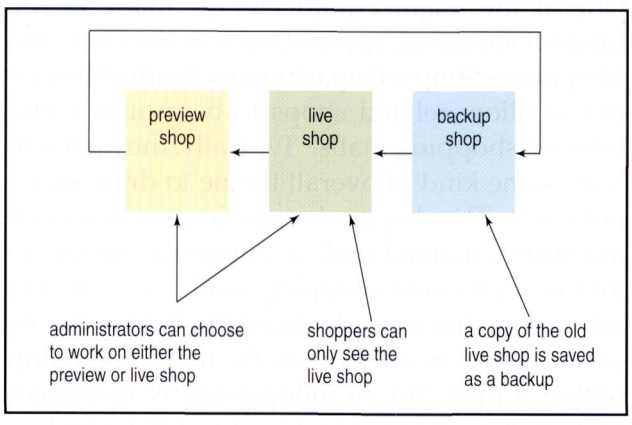


Figure 4: Staging

The Shopping Site hierarchy described in the previous section is in fact replicated, in order to implement the staging concept. Staging refers to the ability of suitably privileged administrators to work on a preview version of a shop's content, so that it can safely be edited and tested without impacting the behaviour of the live shop. When the administrator is happy with his changes, they are applied by making the old preview shop become the new live shop. This works cyclically, as shown in Figure 4.

Extending the use of Active Server Pages

The implementation of the separation of look and feel from business logic and business data is described in this section by comparing the approach taken by Microsoft's Volcano Coffee starter shop with ICL's application.

The Internet shopping application is based on the use of Active Server Pages (ASP), a Microsoft technology that allows the HTML Web pages displayed on a user's browser to be dynamically generated. As used in Microsoft Site Server Commerce Edition, ASP allows a complex shop to be constructed from a small set of files, each of which contains "hot spots" into which parameters (such as product description and price) are substituted at the time HTML is generated from the shop's relational database tables.

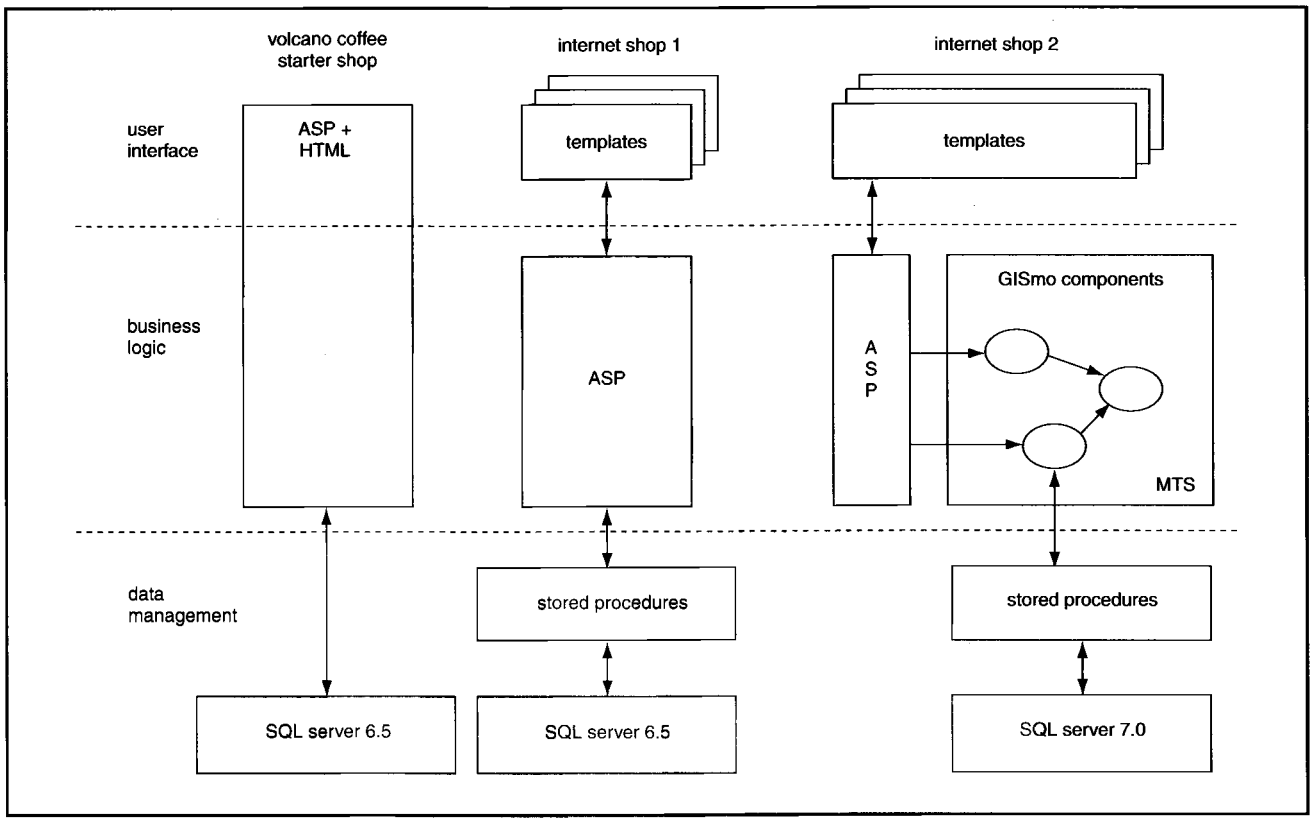


Figure 5: Two and three tier structures implemented with Microsoft Site Server 3.0 Commerce Edition

The Internet shopping application takes this parameterisation further than Microsoft's product does. In order to implement the themed mall concept, the identity of a shop is itself generated dynamically when each Web page is requested. This allows generic ASP pages to be written that can be used in many different shops across an entire shopping site.

Volcano Coffee: The entire functionality of the site is held in ASP files. A single ASP file contains the business logic for each Web page on the site, outputs its HTML and interacts directly with the SQL Server database.

Internet shop phase 1: ASP files contain only business logic. Look and feel aspects are separated out into a set of web templates, which allow that business logic to be presented in different ways depending on context. All database access is performed using SQL Server stored procedures.

Internet shop phase 2: The business logic is implemented in components, giving better performance, security and scope for reuse. ASP is simply used as the glue that holds components together.

Dynamic Templating

Dynamic Templating can be used to decouple program script from presentation data, allowing the same business logic to be presented in different ways according to brand image. With dynamic templating, each logical location on a Web site can have its own template file, specifying that particular page's look and feel. The ASP file no longer outputs HTML directly; instead, its function is to assemble a list of parameters that are inserted into the appropriate Web template file for the current location. Merging these parameters with the template produces the HTML seen on the shopper's browser.

On simple low-cost sites there need not be any shop-specific pages at all. In many cases, however, it is likely that shop operators would wish to customise their shop's appearance and, perhaps, to supply their own editorial content. To make this easier, the application adopts the dynamic templating mechanism.

For example, the ASP file that displayed a product page is common to all shops on a mall. It encapsulates the business logic that provides the details for a product. However, each shop may

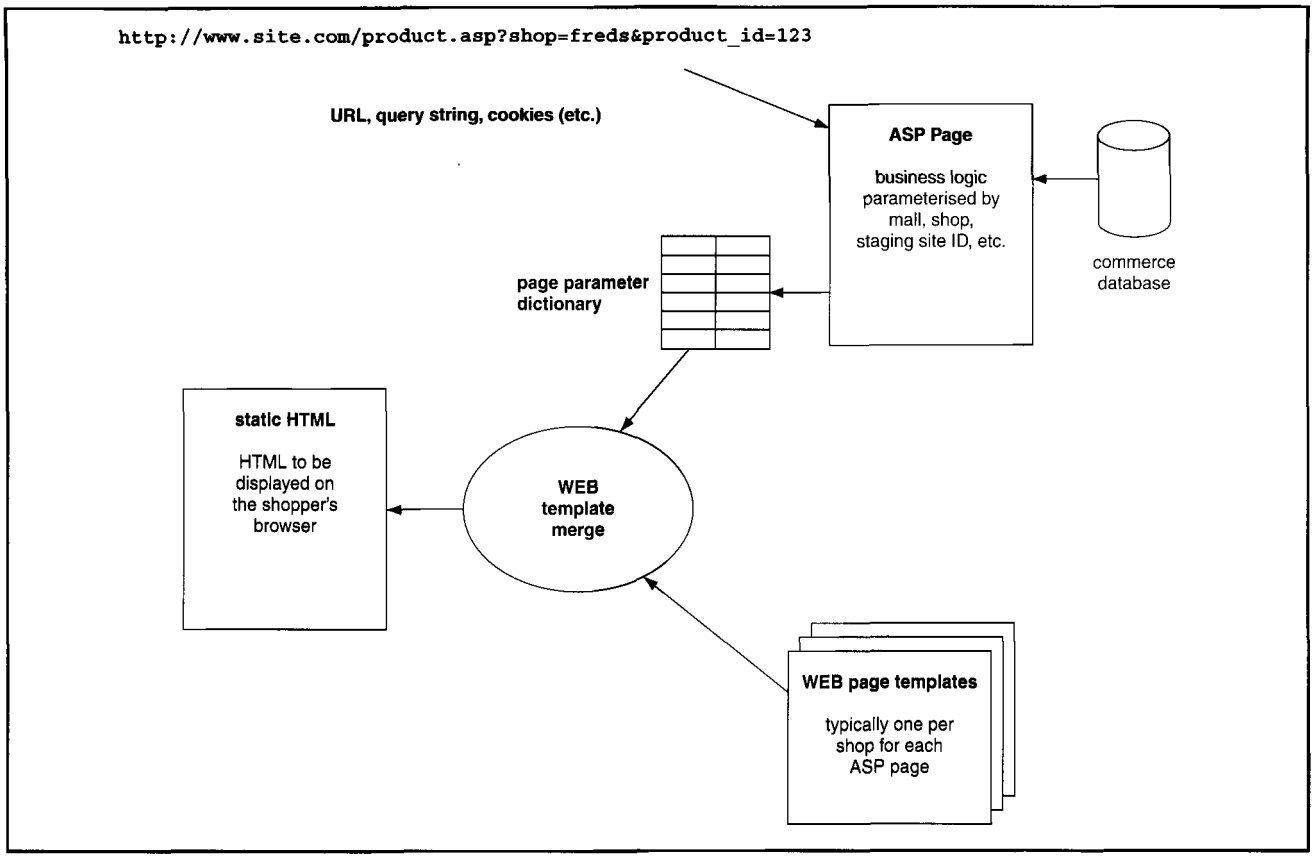


Figure 6: Dynamic templating example

require its own customised product display, incorporating the shop's particular branding and layout. The presentation is specified in a shop-specific template for that ASP page as shown in Figure 6.

Conclusions

This paper has presented some of the lessons that have been learned about how to provide good Internet shopping services to customers, and an overview of the approach that has been used to deliver successful sites, such as the recently relaunched BBC Shop and Buckingham Gate. Some of the more interesting aspects of the technical architecture that have been used have also been introduced.

Shopping on the Web will continue to grow. Businesses are only just realising the potential of this new medium. ICL continues to invest in the development of the Internet shopping service, working closely with partners such as Microsoft. As customers become more accustomed to what the medium has to offer and demand more from the service, so the service will be refined and grow to meet those needs.

Acknowledgements

Many colleagues in ICL have contributed important ideas to the subject of Internet shopping and electronic commerce. In particular thanks to Colin Rutland for much of the technical material in this paper, Jeff Parker, Paul Duxbury and Richard Lister for advice, guidance and assistance in the production of this paper. Additional thanks to the ICL electronic commerce Working group, led by Dick Emery and Dave McVitie.

Biography

Steve Picken joined ICL in 1992, and has spent time in a variety of technical roles. His interest in Internet shopping dates from 1996 while working as an engineer developing Internet shopping sites. Recently Steve has been involved in the design and development of ICL's Internet shopping service. Steve holds an honours degree in Computer Science from Leicester de Montfort University. He is a Chartered Engineer and a Member of the British Computer Society.

Online Loyalty in a U.S. Supermarket Environment

And that no man might buy or sell, save he that had the mark, or the name ... or the number of his name.
REVELATION 13:17

Doug Urquhart

ICL Retail Systems, Santa Clara, California, U.S.A.

Abstract

Consumer Relationship Marketing (Corema) is based on making efficient use of Marketing resources by targeting those resources to fit the shopping habits of individual customers. In order to do this, individual customers must be identified when they shop.

In some environments, such as Banking or Public Utilities, customer identification is the norm, but in Retail, particularly in environments like supermarkets, many transactions are paid in cash, and hence cannot be linked to a particular customer.

A *Loyalty Program* embodies the elements which not only identify the customer, but provide her with an incentive to identify herself.

Each customer is given a means of identification (perhaps a card) which must be presented at *Point of Service*. As an incentive for using the card, the customer may be offered various rewards—*discounts, points or continuity offers*—all available for redemption straight away. To support this functionality, the loyalty system must be online to the store.

This paper addresses the technical challenges which arise when integrating a loyalty system into the large-scale, high-performance environment of a major U.S. supermarket chain. To give an idea of the scale, such a chain would have upwards of 1,500 large stores, in six time zones, all of which operate 24 hours a day, seven days a week. The active customer base is larger than the population of Scotland.

Why Corema?

The North American retailing environment is highly competitive and unbelievably noisy.

It is not sufficient for a retailer to present a good range of high-quality products, at a fair price; he must also make strenuous efforts to attract customers, or he will lose them to his competitors.

At least, that is the theory.

For example, as this paper is being written, we are approaching Presidents' Day (February 15th). For reasons which are unclear, American car dealers chose this day as the beginning of their spring campaigns. The result is a deluge of unsolicited mail, TV advertisements of unspeakable brashness and unsolicited phone calls, usually around dinner time.

This effort is, to a large degree, counter productive. Consumers become adept at tuning out the low-level stuff, and anything which gets through the filter tends to annoy rather than attract.

Unfocused advertising is very expensive, as well.

The Corema approach, on the other hand, involves focusing your marketing efforts on an individual customer, based on what you know about him. A retailer who gets this process right will retain existing customers, attract new ones, and make much more effective use of his marketing budget (all other things being equal, of course).

The US Supermarket Environment

In this paper, we concentrate on the U.S. Supermarket environment. In addition to the attributes of competitiveness and noise mentioned above, we find the distinguishing features of:

- low margins but price sensitivity
- wide disparity in chain size
- low product differentiation—no customer loyalty

Low margins but price sensitivity

Since the products and the retailers are all much the same, the traditional way of influencing buying habits has been to drop the price.

If you sell a luxury item, you generally have a fair amount of margin to spend on marketing. If you are selling food, you may have to sell it below cost to persuade people to buy it (of which more, later).

Wide disparity in chain size

The biggest American supermarket chains are very large indeed—1,500 or more stores, scattered across the country. The regionals, on the other hand, could be as small as 100 stores. (There are chains smaller than this, but we'll leave them out of the discussion).

The forces of merger and acquisition are strong in this market segment, so it is not unknown for a chain to double in size 'overnight'.

Customer bases are typically very large—everyone has to buy food.

Low product differentiation—no customer loyalty

If you prefer Honda to Chrysler, you will tend to buy a car at a Honda dealer. A can of beans, on the other hand, is unlikely to influence your choice of grocer.

Let's face it—buying food at a supermarket isn't one of life's great pleasures. (At least it isn't for most people). The usual reason why we shop at a particular supermarket is that it happens to be the closest to our home, or office.

Traditional Marketing Approaches

Until the concept of Corema came on the scene, Supermarket campaigns were product driven, with goals such as maximizing the sales of Pepsi products. The methods (*offers*) used to influence customers to help achieve these goals generally involved price reductions of some sort. The money to cover lost margin often came from the vendor of the product.

Reduced Price

This is the simplest kind of offer—drop the price of something (a *loss leader* would be sold below cost), advertise the fact that you have done so, in signage, fliers, TV spots etc..

This method has some drawbacks:

- Everyone gets the offer, even people who were going to buy the product anyway
- It attracts *cherry pickers*
- Astute small shopkeepers can stock up at your expense.

Coupons

These offer the item at a lower price, but only if the customer has a paper coupon. You print the coupons as part of your advertising material.

- Low price is limited to those who care enough to bring in the coupon.
- You can limit the number of times the coupon is used (like once)
  but
- You still get *cherry pickers.*
- Customers who have to line up behind an enthusiastic coupon-collector can get sufficiently irritated to make them shop elsewhere.
- Store staff have to collect, and account for, fiddly little bits of paper.

Electronic Coupons

With the advent of intelligent POS systems, it became possible to construct quite elaborate couponing schemes, and to perform this process electronically, without the need for time-consuming paper coupons. The Point of Sale system tallies-up the things required to receive an award, and automatically gives the award when the required threshold is reached.

An example of an elaborate electronic coupon might be:

Spend more than $10.00 in the deli department and more than $15.00 in the meat department, and receive a voucher for 5% off your next transaction (offer valid between 8.00pm and 10.00pm on Thursdays), limit three per household.

Sounds good; the Marketing people can get more inventive, no more fiddly little bits of paper, and we can enforce limits on the number of times the coupon is used.

Well, actually, there are some serious drawbacks to electronic coupons:

- Everybody gets the offer, even those people who were going to buy the item anyway. (Yes, we've taken a step back to a simple reduced price)
- Generally speaking, customers don't understand complicated offers, and this tends to cause customer service problems
- The customer may not even know they have received the offer—a line item on the receipt may be the only notification they get
- An enterprising small shopkeeper can split his purchase into several transactions, to avoid electronically-enforced coupon limits.

Corema

The Corema approach is customer driven rather than product driven.

Based on what we know about the shopper—her buying habits, where she lives, etc.—we construct very specific offers, targeted at her, and not available to any other customer.

The objective of this kind of approach is a behaviour change, for example:

- buying more high-margin products
- shopping with us more often
- shopping at off-peak times, to allow for more efficient work scheduling
- deciding to shop with us rather than a competitor
- shopping with us again (repeat shopping).

Since we are monitoring her buying habits, we can tell whether the offer was successful, and bear that information in mind when constructing the next offer.

Market Segments

In practice, *targeting a promotion* at an individual customer is not yet achievable. With a customer base of many millions, current systems are not capable of managing the complexity of this.

To simplify matters, customers are grouped into Market Segments.

A Market Segment is a fairly arbitrary grouping, based on some kind of common criterion. The criterion may be supplied by an external agency, such as geographical or demographical groupings, or calculated by the retailer's Data Analysis process.

A shopper may belong to multiple segments.

Examples of market segments are:

- Shoppers who live in Power Cable, Nebraska
- Households with an income greater than $100,000 p.a.
- cat owners
- parents
- shoppers who spend more than $1000 per week in our stores
- shoppers who spend less than $10.00 per week in our stores.

Thus, an offer is targeted not at Mary Jones (who has a cat) but at all cat owners. This makes targeting easier to manage (but analysis of success more difficult).

Why have a Loyalty Program?

The key to Corema is identifying the customer when she makes her purchases. Without this basic information, we cannot begin to target offers. The obvious answer is to give the shopper some means of identification, and ask that she use it when shopping. A *loyalty program* gives the customer an incentive for doing so.

Loyalty Programs

Retailing , like the armed forces, has a jargon replete with euphemism. Customers don't steal things—*shrink* happens. Store detectives don't catch crooks—they indulge in *Loss Prevention.*

In like vein, a *Loyalty Program* is a means to entice shoppers to identify themselves every time they shop, by providing some kind of incentive. *Loyalty* means coming back to spend more money.

Batch Loyalty

The simplest form of *loyalty program* is one which performs its operations in batch mode. The airline *Frequent Flyer* programs are a good example. The customer is identified at the time of purchase and this identification is included in the sales data collected as part of the normal Sales Audit process. A batch program processes the sales data at regular intervals (say monthly), totting up the amount each customer has spent.

Based on the amount spent, the customer is awarded something—*miles*, gift certificates, tee-shirts, etc..

A regular mailing to the customer, rather akin to a bank statement, tells the customer what they have earned, and may include awards. The cost of mailing to a large group can be astronomical.

In an environment such as Air Travel, where the rate of shopping is low, customer bases are small, and margins are high enough to justify the cost of mailing, this approach can work very well.

In the more hectic, low-margin environment of the supermarket, its usefulness is more limited. Those pioneers who have adopted supermarket batch loyalty programs have tended to target them at small subsets of their customers, such as the top 10,000.

Online Loyalty

An Online Loyalty system uses the Point of Service system as the vehicle for communicating with the customer, thus avoiding the high cost of mailing.

The customer is identified at the time of purchase, and their account information is retrieved from a central database. At this time, their eligibility for offers is established, thus allowing the Point of Service system to give customer-targeted discounts, alternative prices in that transaction, personalized messages, balloons etc..

At the end of the transaction, transaction spend, coupon usage, points earned and the like are passed up to the central database, allowing it to be updated.

The POS receipt can be used to tell the customer what awards he has been given, and what progress he has made towards future rewards.

The simplest form of online loyalty uses the concept of points. For every dollar spent, the customer receives a number of points. These points can be accumulated and redeemed on demand, as tender, at the end of the transaction.

The loyalty mechanism also provides an infrastructure whereby electronic coupons can be given a continuity element; i.e. an offer which previously said spend $100.00 in this transaction and get a coconut might now say spend $5,000.00 this summer and get a ticket to Fiji.

Smart Cards

Smart Cards can provide similar functionality to an online system, and tend to be used in situations where the high cost of cards (and the necessary readers in every POS lane) can be justified. For example, Smart Cards might be used in countries with unreliable telecommunications, groups of retailers with no common network, retailers with no network at all etc..

For reasons outside the scope of this paper, Smart Cards, although very successful elsewhere, have not gained significant acceptance in the US, except in the field of EBT, and will not be discussed further in this document.

Identification

A loyalty program's main task is to identify the customer. This identification must take place at every transaction. It must be something that the customer will want to use, and which will not have an adverse impact on normal operations.

There are many ways of achieving this and some examples are now discussed.

Bar Coded Cards

Since all supermarkets have bar code scanners, capable of reading EAN and UPC encoded bar codes, a low cost way of identifying a customer is to issue her with a plastic card with a bar code containing her customer ID. This card is scanned at the beginning of a transaction.

(Sometimes the customer is given the choice of a bar-coded keychain ornament rather than a card, or perhaps a stick-on tab for attaching to an existing credit card, since most U.S. shoppers' wallets are groaning with cards already.)

This is, indeed, a low cost solution, and is the most widespread in the U.S., but it has a serious drawback. Since there is no centralized body to regulate the number ranges used, it is quite likely that two competing supermarkets will use the same numbers. So customer 123 of Acme Supermarkets might be J. Bruce Ismay, while customer 123 of Nadir Food Mart is Violet Jessop. If Ms Jessop uses her Nadir card at Acme, she could get all of Mr Ismay's awards.

The mechanism typically used to avoid this is to put the supermarket logo in large friendly letters on the face of the card, and hope the cashier notices when the wrong card is proffered. Sometimes it works.

Magnetic Cards

A much safer system is to issue each customer with a plastic card containing a magnetic stripe, just like a credit card. As with a credit card, the first six digits of the customer number are assigned by the ISO, and are unique to the retailer.

The downside to this approach is the slightly higher cost of the card, and the possible need to buy card readers. Not all U.S. supermarkets take credit cards, and the ones that do may not offer this service in all lanes.

Phone Numbers

Some retailers have been experimenting with the optional use of phone numbers, and have had very positive feedback from their customers.

Although intrinsically less safe than bar codes or magnetic stripes (nobody has thought of putting a check digit on phone numbers), phone numbers can work well enough, particularly if the feedback from the central database includes a customer name, and if customers are combined into households (i.e. more than one shopper can contribute to a single total).

Others

Other mechanisms have been tried, or suggested.

Drivers' licenses can work, to a limited degree, but the fact that they all differ in format, and are assigned on a state-by-state basis, can lead to operational problems.

Fingerprints, too, have been suggested. Although this may sound rather Orwellian, some states, notably Texas, collect fingerprint information with drivers' license records. The technology to use fingerprints for lookup (rather than confirmation) is not yet cost effective at POS, however.

The drawback to any radically new approach which requires additional equipment is the multiplier. Consider a chain with 1500 stores, each of which has 20 lanes. Take the cost of your new device and multiply by 30,000. Now, do the benefits outweigh the cost?

Incentives

Any identification method is worthless if the customer does not use it.

Although the retailer can train their cashiers to remind the shopper to use their ID, this is largely ineffective unless the shopper is convinced that there is 'something in it' for her. A good loyalty program will make the shopper insist on using her ID.

Here are some typical incentives associated with online loyalty:

Discounts

The act of using the card entitles the shopper to special prices throughout the store. Casual shoppers don't benefit. Since we have online access to the customer's database, special customers can have special discounts.

Continuity Offers

Since we have an infrastructure which allows customer information to be retrieved at the start of a transaction, and updated at the end, we have the ability to accumulate totals over several transactions. This has several advantages:

- we can award more substantial rewards, since they can be based on sustained spend over a period of time. Clearly an award based on spending $1,000.00 can be a bit better than one based on buying a can of beans
- There is an incentive for the shopper to keep coming back to your store—the more they accumulate, they more they have to lose by shopping elsewhere, particularly if you put a time limit on the offer.

Points

Taking the continuity concept a step further, we can allow the customer to accumulate points, each time she shops. Points are a notional currency, redeemable on demand by the shopper. Since points can reflect several months worth of purchases, they can represent a strong incentive for the shopper to continue to visit your store.

In multi-currency or multi-retailer environments, points can also provide a common ground.

Charities

It is fairly common for US supermarkets to make contributions to local institutions (such as schools), based on the amount spent by participating customers. Typically such schemes are manual and horribly labour-intensive, but are considered to be financially worthwhile, since they seem to generate customer loyalty, and a certain amount of good publicity at the grass-roots level. The fact that such contributions may be tax-deductible may also be relevant.

By extending the points concept slightly, and allowing a shopper to accumulate points on behalf of such an institution, the online system can support charities. Note that the customer may be able to accumulate points, but cannot redeem them—this is performed by a batch process.

System attributes

The U.S. Supermarket environment dictates that an online loyalty system should have certain attributes, notably resilience, scalability, flexibility, platform independence and a good underlying network infrastructure.

Resilience

U.S. supermarkets operate around the clock. Although there may be quiet periods during the night, these are eroded when the chain stretches over a multitude of time zones.

The retailer will expect any online system to operate during his trading period (i.e. all the time). The system must therefore be resilient to the failure of any component, and capable of performing housekeeping activities without relying on 'down-time'.

Scalability

Some U.S. Supermarket chains are very large indeed. To a degree, platform independence provides system scalability, but in addition the system architecture should allow for easy scaling.

Flexibility

The online loyalty system must be flexible enough to coexist with many flavours of Point of Sale or Point of Service systems. Some of the older POS systems tend to be expensive to modify, so the onus is placed on the Loyalty system to accommodate their shortcomings.

Platform Independence

In the U.S., large retail chains tend to have environments where adherence to a particular Operating System or Relational Database is a definite disadvantage.

UNIX servers are very common, with an assortment of flavours of UNIX. NT is still not perceived to be Enterprise-ready by many retailers.

The retailer will often have a corporate license for a major Relational Database, such as Oracle

or Informix, which makes an alternative suggestion commercially non-viable.

It is essential that any system being proposed in such an environment be designed around Open Standards, to allow Platform Independence.

Network Infrastructure

For an online loyalty system to operate, it requires an online link between its central database and the stores. Medium to large supermarket chains usually already have such a network in place, to handle collection of Sales Data, and servicing of Online Credit. (This is not always the case with large Specialty chains whose small-format, low volume stores can manage to get by with a combination of SNA or even Bisynch connections for data collection and dial-up asynch for credit.)

The ICL Corema solution

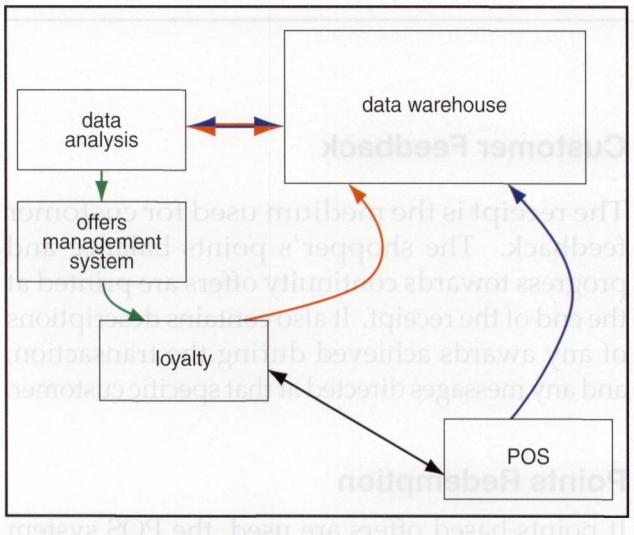


Figure 1: The *Corema* System

ICL's Consumer Relationship Marketing system for the U.S. Supermarket arena is called *Corema*®.

The overall shape of the *Corema* system is shown in Figure 1.

Sales Data flows from the POS system into the Data Warehouse, where it is analyzed by the Data Analysis system. The result is target groupings of customers (Market Segments) which are combined with offers generated on the Offers Management System and fed to the Loyalty subsystem.

POS interacts with loyalty, allocating offers to the appropriate customer, and updating loyalty with the results of those offers.

Loyalty, in turn, feeds the Data Warehouse with the loyalty transactions.

And so it goes on.

The loyalty system is shown in more detail in Figure 2.

The loyalty system is actually three processes. These may be on physically different machines, or may run on the same platform. The multiple machine approach is preferred, since it offers a higher degree of resilience.

POS interacts with the Front End, which feeds the CMC (Customer Management Center) with transaction data. The CMC can support POS interaction if the Front End should fail.

The CMC contains a customer database, containing Name and Address information, plus all the running totals associated with each customer. In a points based system, for example, these would be points totals.

User access to the Loyalty system is through a web server, via the retailer's corporate intranet.

POS

Point of Sale or Point of Service (e.g. kiosk). The prime interface with the shopper. For simplicity, we'll consider a standard in-lane Supermarket terminal.

In addition to the standard sales transaction, POS needs extra functionality to support Online Loyalty.

Customer Identification

At some time in the transaction (preferably the beginning), the terminal must capture the customer ID. This could be performed by scanning or swiping a customer card, or keying in a customer number or phone number. If there is any possibility of local validation, such as check-digiting, it is done at this time.

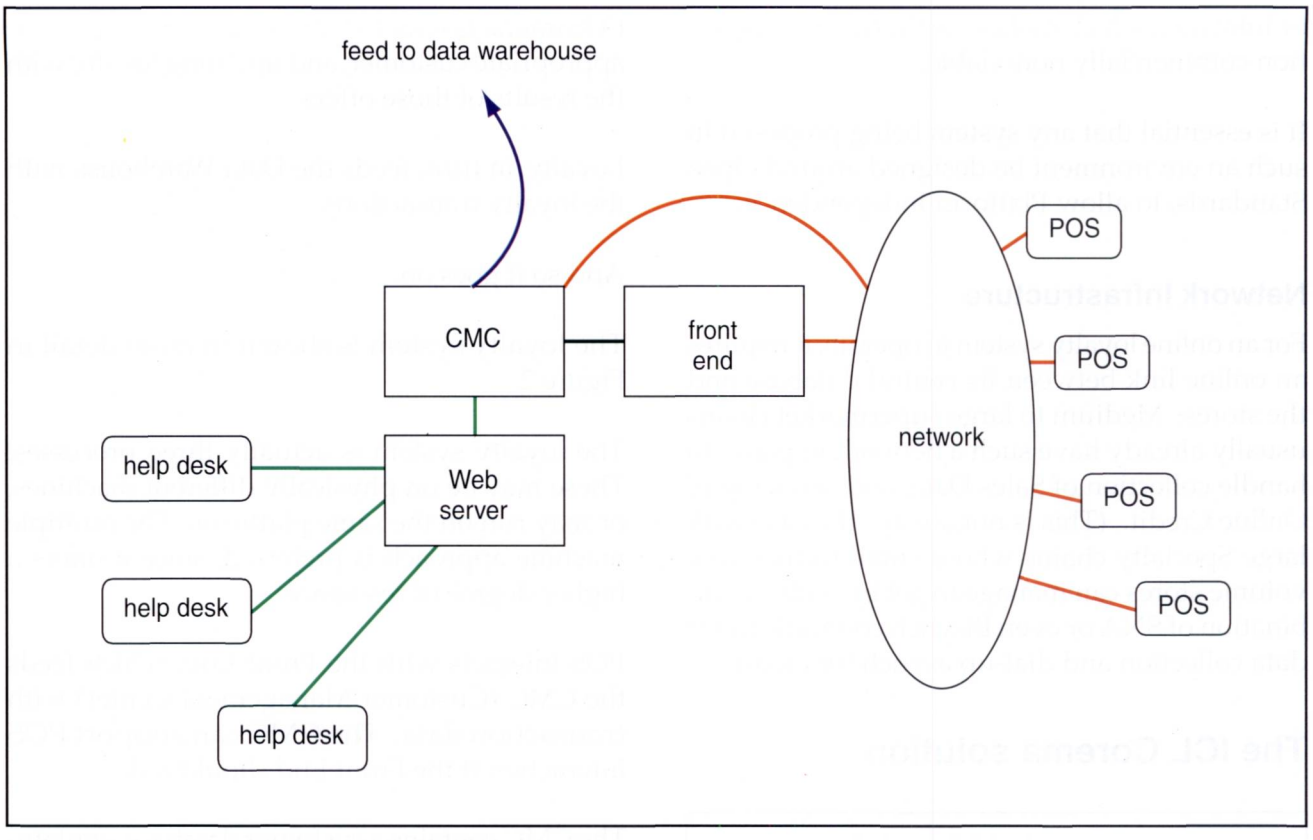


Figure 2: The Corema Loyalty System

Once the customer ID has been collected, it is sent in a message to the Corema front end. The transport mechanism used is TCP/IP.

Customer Eligibility

The Front End replies with a message listing the Market Segments the customer belongs to, and any previously accumulated totals from earlier transactions.

Offer Execution

Once POS receives eligibility information, it can execute the offers targeted at that customer.

It may choose to:

1. wait until the end of the transaction to do this (thus minimizing any reliance on the response time from its first message, and maximizing its own end-of-transaction time)

2. wait until the reply before proceeding (with opposite results).

Approach 1 is the norm.

Customer Feedback

The receipt is the medium used for customer feedback. The shopper's points balance and progress towards continuity offers are printed at the end of the receipt. It also contains descriptions of any awards achieved during the transaction, and any messages directed at that specific customer.

Points Redemption

If points-based offers are used, the POS system will allow some or all of these points to be redeemed as tender.

Update

Once the transaction is complete, a message is sent to the Corema Front End containing any modifications to the customer's totals which have taken place within the transaction. For performance reasons, this message is queued, and sent by an independent, asynchronous background process. This process waits for positive acknowledgement from Corema before sending the next message.

Fallback Mode

If the Corema Front End should fail, the CMC will take over processing the identification messages. This is transparent to the store system, as the switchover is handled by a DNS server. To reduce the load on the CMC while in this mode, update messages are not transmitted—they are stored, for forwarding to the Front End when it is back online.

Offline Operation

Another consequence of the resiliency requirement is the need for POS to provide some measure of functionality when the network is unavailable.

To this end, the POS system may cache customer IDs and their associated Market Segment Numbers. In the event that the store is offline, the POS system has the ability to identify the customer and establish eligibility.

The update messages generated at end of transactions are stored while the system is offline, and forwarded when the system is back online again.

The Corema Front End

This component is dedicated to servicing the online requests from POS, and passing transaction data to the CMC.

Internally, it consists of one or more pairs of software components:

- an online element, which performs communication and routing
- a database, which contains a subset of the main customer database (no address data, for example), denormalized for performance reasons. The database is a standard RDBMS, such as Informix or Oracle.

Scalability is achieved by increasing the number of component pairs. These pairs can exist on single or multiple hardware components. In a large chain, there would be several front end processes, each dealing with a subdivision of the chain. The online element is capable of detecting messages aimed at databases other than its 'own' and routine the messages accordingly. This is to cater for the case where a shopper is visiting a store in a division different from the one where she joined (Cross Shopping).

A message-oriented-middleware product—we use Talarian Smart Sockets—is used for message communication between the components. This is a guaranteed-delivery, publish-subscribe mechanism.

Identification

When the first message is received from POS, the Front End retrieves such information from its database as Customer Name, Market segments, previous offer activity. It sends a message containing these back to POS.

As an aside, if the customer is not found in the database, it could mean that this is the first time that the customer has used her card. In this instant-gratification environment, customers are unwilling to wait until an application form has been processed. If the Front End is 'sure' about the ID, i.e., it was input by mechanical means, it causes a new database entry to be created and, if there are any, replies with details of any market segments or offers specific to new customers.

The degree of work required by the Front End, to establish customer eligibility, is inversely proportional to the similar functionality performed by POS. In its simplest form, the front end returns merely a list of Market Segments to which the customer belongs (and POS interprets what this means in terms of offers). If POS can't do this, the Front End has to send a list of offers instead.

Update

The Front End is also responsible for processing update messages. These contain deltas to offer totals, on a transaction basis. Once it has updated its own totals, and committed the transaction, it sends a message back to POS, to acknowledge that this has happened. Note that POS cannot send another message until this handshake has occurred.

The CMC (see below) also maintains totals, for the whole chain, so that the Front End must forward update messages to the CMC on a regular basis. This is an asynchronous process.

The CMC (Customer Management Center)

There is only one of these. It, too, contains a standard RDBMS. The database on the CMC duplicates the totals held on the Front Ends, but for the whole chain. The various Front Ends trickle feed data to it which it uses to update its own totals, and then passes them on to the Data Warehouse.

The CMC database also contains customer Name and Address, and holds several months' trading information.

The User Interface to the CMC is over the Retailer's intranet, using standard web browser technology. A web server, either running on a separate machine or on the CMC itself, presents Active Server Pages which interface to the database.

The prime purpose of the CMC is to support Customer Service/Help Desk functions. Once a Customer Loyalty system is in place, shoppers become very keen to ensure that they get every point to which they are entitled. The result, apart from generally increased shopping, is an irate call to the Help Desk if they think they are losing out.

Help Desk

The Help Desk is designed to allow an operator to access customer information such as:

Customer IDs (card number, phone number etc.)
Name and Address
Household groupings
Progress against continuity offers
Points balances
Details of past transactions (say the last three months).

The operator also has some limited ability to make adjustments, to soothe the irate.

Although the most common user interface is a Help Desk on the company intranet, others are being experimented with. If the retailer's security standards will allow it, shoppers can access their own information, either via the Internet, or from their telephones, using an IVR unit.

An Example of Help Desk Functionality

To give an idea of how the Help Desk would operate, let us consider the example of Mr Homer Simpson, who made a purchase, and has phoned the Help Desk because he believes he should have received an award.

Of course, he does not have his card when he phones, so the Help Desk operator will perform a look-up using the screen in Figure 3, based on his phone number or, if this is not on file, his name.

Fortunately his name is fairly uncommon.

After verifying that she is, indeed, talking to the correct Homer Simpson, perhaps by verifying the address or phone number, the Help Desk Operator can look at the system to find the transaction in question. She does this by clicking on the Transactions button. See Figure 4.

Homer tells her that the transaction was in October or November, 1997. She sees a transaction which might be the one she wants, on October 24th. She also notes that the transaction had triggered activity on three offers. See Figure 5.

By drilling down on the Offers column, she can see further detail. See Figure 6.

The record shows that he has spent $100 towards a Wine Club offer which gives an award at $100.

He agrees with this, but maintains that the system didn't give him an award. This is borne out by the 'hits' column, which shows a zero—no award was triggered.

However, the Help Desk operator notices that the offer in question has a limit of 1 (shown in the Max hits column).

She returns to the main screen, and this time clicks on the Offers button.

By looking in the Maximum Hits column, she finds out that the offer was limited to one per household, and Homer had already had his one. See Figure 7.

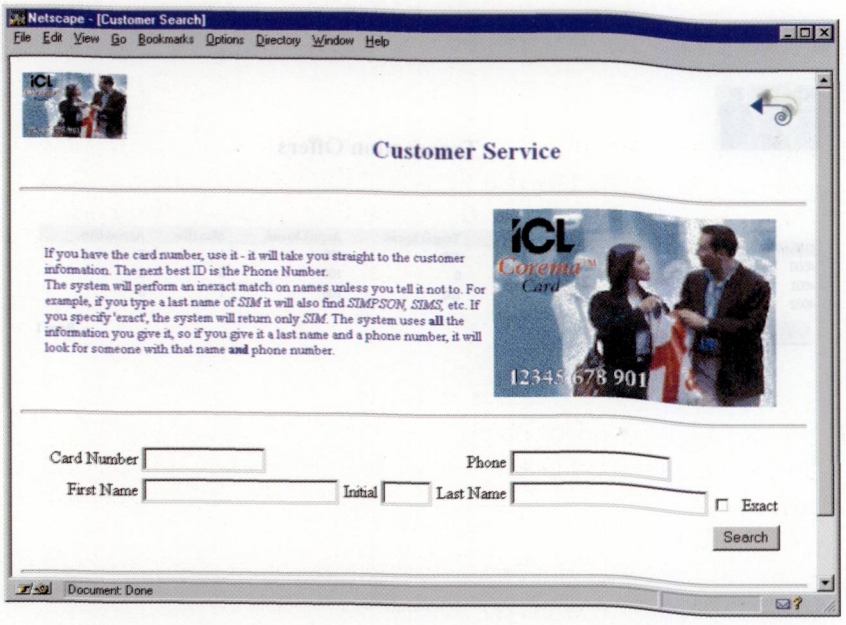


Figure 3: Customer Service

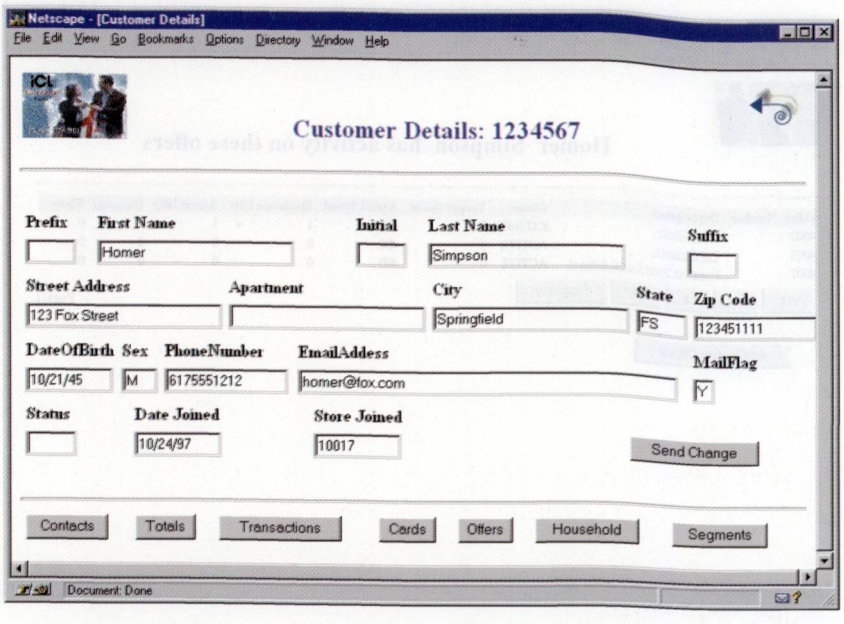


Figure 4: Customer Details

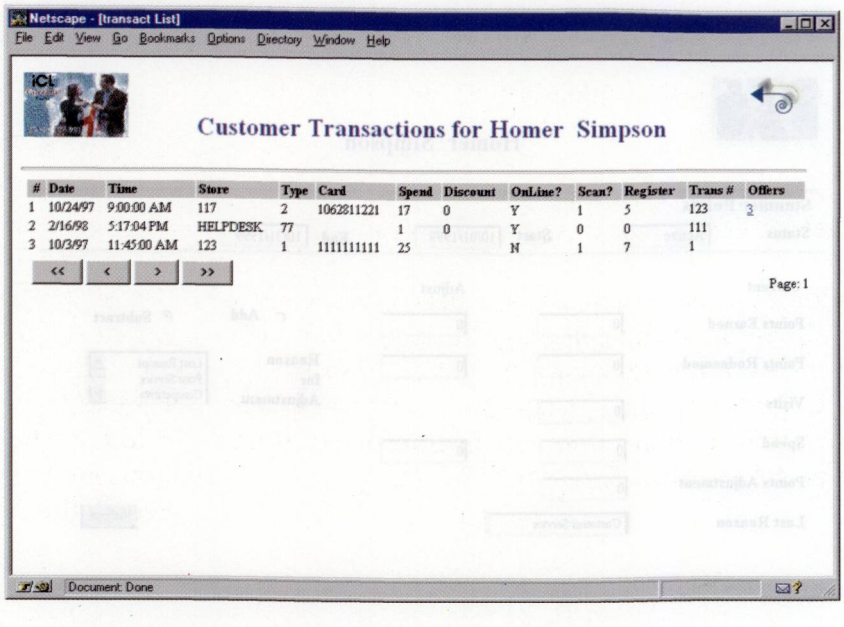


Figure 5: Customer Transactions

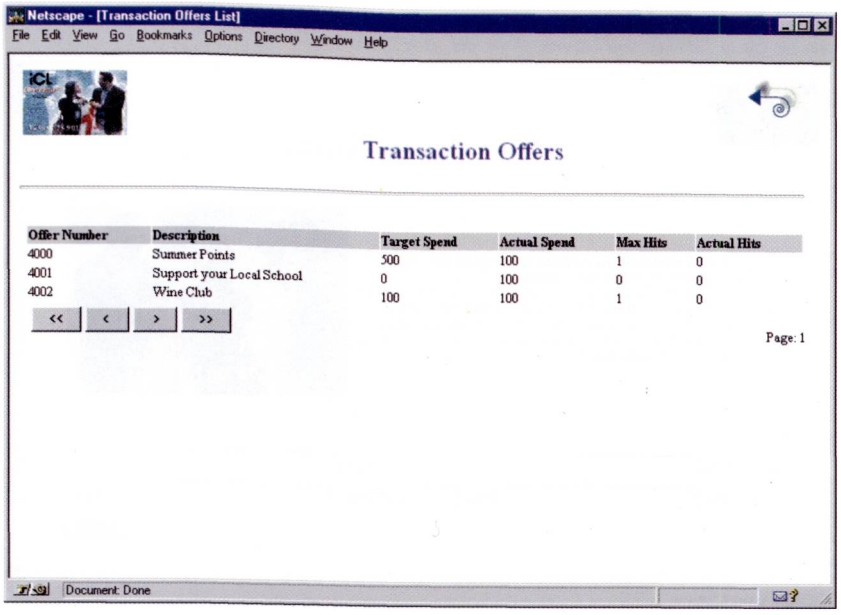


Figure 6: Transaction Offers

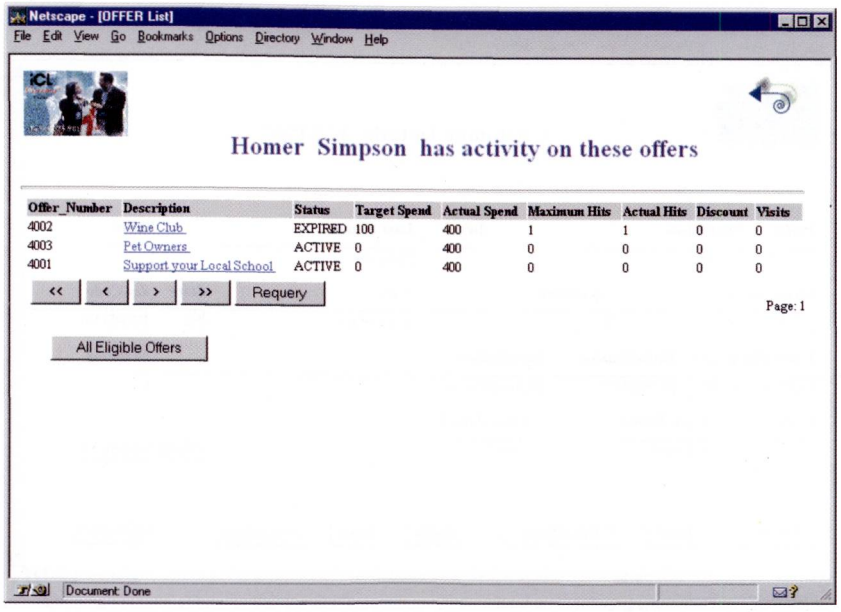


Figure 7: Offer Activity

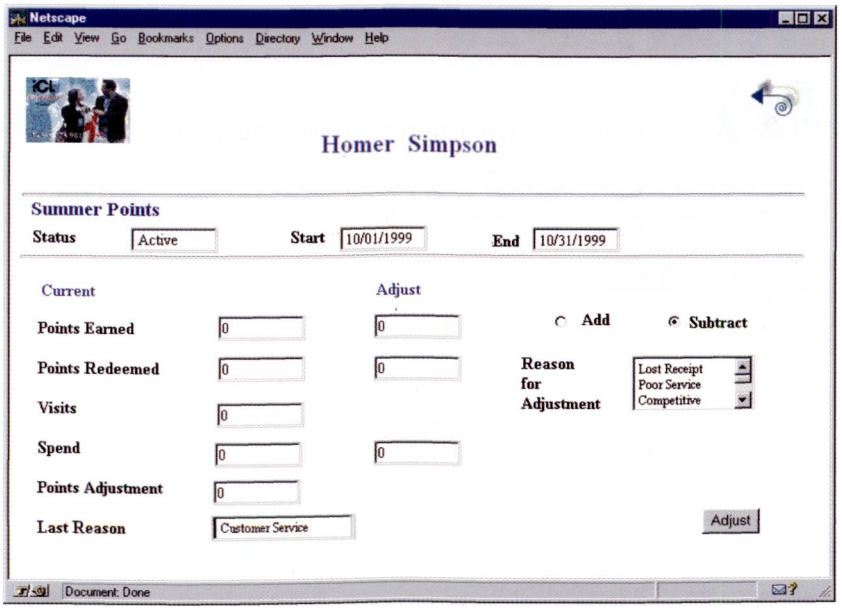


Figure 8: Points Program

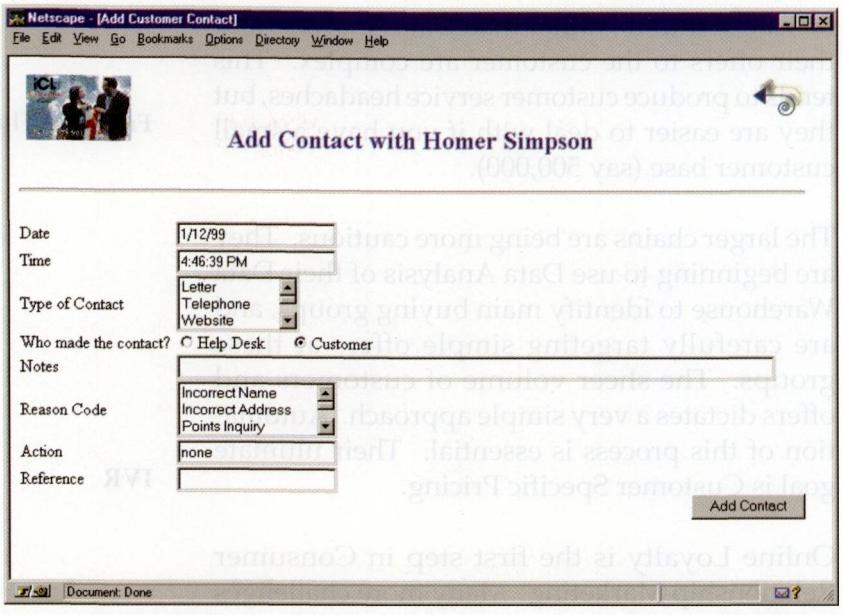


Figure 9: Contact Record

If this had been an active offer, and there had been a genuine mistake, the Help Desk operator could have amended the totals.

Since the original offer is long gone, and anyway the customer was in error rather than the system, the operator makes a discretionary decision to give Homer an extra 50 points, as a goodwill gesture.

She selects his Points Program, and makes the adjustment. See Figure 8.

Finally, she records the interaction with Homer by creating a Contact record. See Figure 9.

All of these actions are logged, and in addition are trickle-fed to the Data warehouse, where they can be analyzed.

Results:

- Homer is happy. He has got 50 points for nothing.
- The Customer Service Rep is happy—the whole interchange took a few minutes, and all the information was to hand.
- The Retailer is happy—no expensive mailings, upset customer, litigation etc..

The end result of all this unbridled happiness—the customer continues to use his customer ID, gets the occasional award, continues to shop with the Retailer, and provides regular helpings of Identified Sales Data.

And that, of course, was the reason for having a Loyalty Program.

The Next Step

Corema Online Loyalty is installed in several supermarket chains in the US.

Since the infrastructure of Online Loyalty provides a very low cost communication channel to the customer (much less expensive than mailing) our customers are finding other ways of using this channel—broadening the scope of Consumer Relationship Marketing.

We have seen:

- a message sent to the cashier when one of the top 10% shoppers makes a purchase, so that they can receive VIP treatment
- sweepstakes, where a random customer receives an in-lane prize [anything from a free turkey to a Jeep]
- rebate of overpaid tax
- 'stealth' offers, where the shopper doesn't know the rules—awards are related to spend, but the thresholds triggering them are not visible, or published.

However, the most important use of the Online Loyalty channel is the delivery mechanism for targeted marketing.

The smaller chains have dived into this subject with great gusto. Most of their targeting is based

on rule of thumb, rather than Data Analysis, and their offers to the customer are complex. This tends to produce customer service headaches, but they are easier to deal with if you have a small customer base (say 500,000).

The larger chains are being more cautious. They are beginning to use Data Analysis of their Data Warehouse to identify main buying groups, and are carefully targeting simple offers at these groups. The sheer volume of customers and offers dictates a very simple approach. Automation of this process is essential. Their ultimate goal is Customer Specific Pricing.

Online Loyalty is the first step in Consumer Relationship Marketing. Many more challenges lie ahead.

Glossary

Beano A product which claims to alleviate the socially unacceptable aspects of bean consumption. Not to be confused with the classic British comic.

Cherry Picker A shopper who goes to great lengths to select only those items with the lowest prices, often confining themselves to loss leaders (q.v.). Detested by all retailers.

Continuity Having existence outside an individual transaction. An offer which allows the shopper to accumulate a total over several transactions is a continuity offer.

Corema Consumer Relationship Marketing. A methodology for maximizing marketing resources by focusing on customers.

Cross Shopping Shopping at a store in a different division from your own. e.g. You live in Power Cable, Nebraska, but you use your card while on vacation in Secaucus, New Jersey.

Discount An item offered at a lower price than 'normal'.

EBT Electronic Benefits Transfer. An electronic mechanism for doling-out welfare payments. The recipient has a card which can be used for making payment. Smart cards have been used in some states.

Frequent Flyer A loyalty program run by an airline. In return for travelling long distances by plane, the member is given one point for every mile flown (with extra points for people who travel very long distances indeed). The points can be redeemed, so that the member can travel very long distances by plane.

IVR Integrated Voice Response. A unit attached to a phone line which will allow an end user to navigate a menu of pre-recorded messages by means of a touch-tone phone.

Loss Leader A product sold at a price below what the retailer paid for it. Loss leaders are used to attract customers to a store, in the hope that they will purchase other, higher-margin items while they are there.

Loss Prevention A euphemism for 'catching thieves'.

Loyalty Program A system (usually automated) where a customer's identified sales data is collected in return for some customer reward.

Market Segment A group of customers who share an attribute. For example:

- Parent
- cat owner
- cherry-picker.

Miles The airlines' jargon for points.

Offer An arrangement whereby a shopper receives something in exchange for particular shopping behaviour. For example:

- buy can of beans, receive free 'beano'
- spend $250 dollars in November, get a free turkey
- get points every time you shop.

Points A notional currency awarded to a customer based on the amount of money she has spent. Points have value, and may be redeemed on demand.

Promotion	A group of offers designed to achieve some goal. For example: • increase profits by 5% over last year • kick out the competition in Fairfield County • make the cherry pickers go elsewhere.
Sales Audit	Head Office application which analyses Sales Data for financial purposes.
Shrink	The difference between the amount you expect to get from selling items and the amount you actually get. Three main sources of shrink, not in any particular order: • Customer Theft • Employee Theft • Errors.
Targeting	Defining a Market Segment, perhaps as a result of Data Analysis, and then concentrating marketing resources on it.

Biography

Doug Urquhart is a Consultant, working for ICL Retail Systems in Santa Clara, based in Southport, Connecticut.

He is a graduate of the University of Glasgow and joined ICT in 1967, based in Reading. He spent his early years working on various design and development aspects of COBOL compilers on 1900, 2900 and later the System 25. In 1975, he joined the VME/K team, and spent some time providing on-site support for the European Space Agency, in Darmstadt, West Germany.

In 1981, he joined the Retail group, and was responsible or producing the world's first COBOL-based Point of Sale Terminal.

In 1984, he moved to New England, where he has been ever since, as part of ICL Retail Systems. His involvement with the Retail Group has been wide-ranging, from Development through Marketing and Consultancy. In 1994, he won a Gold Excellence award for his part in the rollout of a major Retail Chain.

He is currently part of the ICL Corema group, which is responsible for the largest online loyalty system in the world. Since the group is based in Santa Clara, California, he commutes by telephone!

Using scenarios in ICL to develop and implement strategy

Gill Ringland

ICL, Slough, UK

Abstract

This paper describes the use of scenarios as part of planning and management in ICL over the period 1993 to 1998. The introduction gives a short history of scenarios: what they are and who uses them. Section 2 is a short history of ICL, to set the context. Section 3 describes the work done as part of the Vision 2000 project in 1993. Following on from this, the relationship between scenarios and other forms of futures work was understood better, and this is described in section 4. Section 5 describes a scenario project in 1995, which has become the basis of major strategy development for us. The uses we have made of these scenarios are described in Sections 6 and 7. In Section 8, the paper discusses the scenarios developed in two multi-partner consortia to which ICL belongs, and describes their application to ICL's European Strategy as well as to its business offerings, customers and markets. Section 9 presents the conclusions, and in particular compares the underlying assumptions and concerns as they have evolved over the decade.

Introduction

This paper describes the use of scenarios as part of planning and management in ICL over the period 1993 to 1998.

Scenarios as models of the world

Scenario planning is a set of processes for creating a mental model, which improves the quality of educated guesses, and deciding what backup is prudent.

Models of the world are often used to anticipate "real life". For instance:

- wind tunnels are used to test car shapes for aerodynamic features—for example, does the car become unstable at high speeds, does it have higher or lower drag factors than other shapes
- the use of mathematical or computer models to schedule and allocate resources within sets of constraints. Linear programming techniques are used to solve problems such as forest management, agricultural production and production planning in factories.

Whether physical or computer modelling is used, the predictions for real life are only as good as the ability of the model to contain enough of the rules and constraints of real life. For instance, if a model was based on fixed proportions of income being available for discretionary spend, as a way of calculating the market for luxury goods, it would cease to be applicable if changes in lifestyle meant that increasing proportions were in fact being spent on food due to changes in diet.

Two aspects of a successful model are suggested:

- the ability to anticipate real world behaviour, which may be unexpected, through exploring the constraints; i.e., external environment or internal relationships
- the creation of a mental model which allows the user to look for early confirming or disconfirming evidence.

The history of scenario planning

Scenario thinking traces its history back to the Second World War. After the War, the RAND Corporation was set up to investigate new forms of weapons technology. RAND's Hermann Kahn pioneered the technique of "future-now" thinking, aiming, through the use of detailed analysis plus imagination, to be able to produce a report as it might be written by people living in the future.

The description "scenario" was given to these stories by the writer Leo Rosten, who suggested the name based on (obsolete) Hollywood terminology—he didn't think the then more current term Hollywood "screenplay" sounded dignified enough. Hermann Kahn adopted the term because he liked the emphasis it gave, not so much on forecasting, but on creating a story or myth.

When Kahn founded the Hudson Institute in the mid 1960s, he specialised in stories about the future aimed at helping people break through their mental blocks and consider "unthinkable" futures. He was best known for his idea that the best way to prevent nuclear war was to think through in detail what would happen if the war did occur. At that time, scenario planning was still fairly specialist. It was not until the Hudson Institute started to seek corporate sponsors that companies like Shell, Corning, and General Motors started to be exposed to this style of thinking.

Meanwhile, on the West Coast, Stanford University had already set up it own think-tank in 1947. Stanford Research Institute (now SRI) became the first university related think-tank to offer long-range planning for business, incorporating operations research, economics and political strategy alongside hard science and military consulting.

The late 1960s saw a shift in the work done by organisations like SRI, with increased interest in finding ways to look further into the future to help plan for changes in society, an interest underpinned by the upheavals resulting from the war. In early 1968 the SRI "futures group" began to use a variety of methods, from straight-line numeric forecasts to literature searches on utopias and dystopias from science fiction to create plausible scenarios for education in the US to the year 2000. The scenarios were based on two major questions:

- how good would society be at controlling its destiny?
- would society be flexible, open and tolerant or would it be authoritarian, violent and efficient?

In the 1960s, GE was the role model in corporate planning. It used scenarios primarily to think about the environmental factors affecting its businesses.

Shell's basic planning system was very similar to GE's. By 1965, Shell could plan for the whole chain of activity, from the oil in the ground through to its sale at petrol stations, for six years. But in the late 1960s they realised that for their business, because of the time scale on which oil companies worked, six years was too short a time for planning. So a study was set up to look at Shell's position to the year 2000. This showed that the predictable, surprise-free environment would not continue, and that a shift in power from the oil companies to the oil producers in the Middle East would create major discontinuities in the oil price.

The oil price had been based on seemingly predictable factors of demand and supply since the War. But behind the scenes in the oil producing countries, factors like politics were increasingly important. And Shell's scenarios were designed to help the Shell decision-makers to begin to question their inner model of reality, their "mental models" in the face of possible oil price shocks. Once the Yom Kippur war had broken out in the Middle East, the oil embargo by the producing countries did indeed lead to a sharp increase in the oil price with the inevitable depressing results on the world economy. Because Shell's executives had been prepared for this, they were able to react quickly and began to move from expanding primary capacity to upgrading the output of the refineries, well ahead of their competitors.

During the 1970s scenarios were widely used, often based on SRI methods. However, during the 1980s, the use of scenarios decreased from the peak of interest in the 1970s. The threat of the oil price shock had decreased, corporate staffs had been reduced, and an over-simplistic use of the technique, with a confusion between forecasts and scenarios, had given scenario planning a bad name.

In the early 1980s, Shell was getting more involved in oil and natural gas fields in the North Sea. It based its expansion strategy on the assumption that oil and gas prices would remain stable. In terms of oil, conventional wisdom was that the Arab-dominated OPEC consortium would continue to act in a unified manner towards production and pricing. An informal agreement in Europe not to open the market to more than 35% of cheaper natural gas from the Soviet Union would keep the price of natural gas high.

Scenario work showed, however, that both oil and gas prices could fall. With oil, OPEC's unified facade might crumble, made worse by a slowing demand for oil because of better energy conservation and efficiency. Even more strikingly, the continuation of the Soviet system was not assured, which might have implications for the natural gas market. If a virtually unknown man named Gorbachev came to power, one would see massive economic and political restructuring; an opening of the West; arms control; declining tensions in the West; and major shifts in international relationships. Gorbachev's arrival in power would increase the likelihood of the Soviet Union exporting large quantities of natural gas. This encouraged Shell to drive down costs at one particularly expensive gas field project in the North Sea. At the same time, it avoided investing in new oil fields or following the acquisition trail being trodden by it major competitors, who were engaged in an acquisition spree, buying other oil and gas companies at premium prices. Once the dust had settled following the price drop, Shell was able to pick up additional assets at bargain prices.

Shell is the best known example of a company using scenarios to prepare for major changes. This paper describes ICL's experience, as it wrestled with the volatile IT market of the 90s. The author's book [Ringland, 1997] gives a number of examples of the use of scenarios:

- Austrade
- British Airways
- Cable & Wireless
- DEC
- ECRC
- Electrolux
- Erste Allegemeine Verseichweung
- manufacturer KRONE
- Statoil of Norway
- United Distillers
- UK National Health Service.

The Structure of this paper

The paper starts with a short history of ICL, to set the context (Section 2). The reason for including this is to demystify scenarios—they are useful to companies smaller and leaner than Shell, as the examples above indicate! Section 3 describes the learning process through the work done as part of the Vision 2000 project in 1993. That project helped us to understand better the relationship between scenarios and other forms of futures work, and this is described in Section 4.

Following on from this learning experience, in 1995 we ran a classic "scenarios project", using the best advice we could find, and consulting with industry leaders. These scenarios for the Information Industries, Coral Reef and Deep Sea, have been the basis of major strategy development for us and the uses we have made of them are described in Sections 6 and 7.

Since 1996, we have increasingly worked with a number of external organisations to broaden our view of the forces upon us—taking into account global forces outside our industry. In Section 8, the paper discusses the scenarios developed in two multi-partner consortia to which ICL belongs, and describes their application to ICL's European Strategy and to ICL's business offerings, customers and markets. Section 9 presents the conclusions, and in particular compares the underlying assumptions and concerns as they have evolved over the decade.

Background—a history of ICL

ICL's roots as a "computer" company go back to the immediate postwar years, in particular, to the "Baby Machine" at Manchester University in 1948 [Burton, 1999], which led to the formation of the computer department at Ferranti Ltd., and to the EDSAC machine at Cambridge in 1949, which was developed into a commercial product by Lyons in 1951. The latter ran the first routine office job—Bakery Valuations. Lyons then formed LEO Computers Ltd. in 1954. This company merged with the computer department of English Electric in 1963. Another UK computer company, The British Tabulating Machine Company (BTM), had its roots in the punched card machines invented by Herman Hollerith in the 1880s. BTM was incorporated in 1907 and its history includes the production of components for the famous cryptography operation at Bletchley Park during the Second World War. The Powers-Samas company was set up in 1925 and, in an agreement with Remington Rand in 1936, "divided the world" with Powers taking the British Empire, Scandinavia and the Middle East. In 1959, Powers-Samas merged with BTM to form ICT, International Computers & Tabulators Ltd.. This company later acquired the

commercial computing interests of EMI and, in 1963, merged with the computer department of Ferranti Ltd..

The ICT strategy was [Campbell Kelly, 1989]:

- To become a large scale vertically integrated data-processing equipment manufacturer
- To supply products for the traditional punched card machine market
- To become a peripheral manufacturer and OEM supplier.

Creation of ICL

ICL was formed in 1968 as a government initiated merger of English Electric and ICT, as part of the then Labour government's "white heat of technology" theme, to create a strong British player in the growing computer business.

As the mainframe business began to mature in the 1970s, ICL bought the Singer Business Systems company, which had a base of small business systems and intelligent terminals in Europe and North America, and a headquarters and manufacturing facility in Utica, New York.

During the period, 1979-81, the company ran into trouble. Survival depended on more than cutting costs. Three factors were crucial: firstly, a new management team, including Robb Wilmot and Peter Bonfield from Texas Instruments; secondly, establishing a long term relationship with Fujitsu of Japan to develop the semiconductor circuits needed for its mainframe range and, thirdly, to reset the strategies.

These were stated as:

- deliver complete solutions not components
- create and work with a network of alliances
- lead the introduction of Open Systems (i.e. the break down of vertical integration)
- concentrate on a few markets, e.g. Retail, Financial Services and Government.

Since then it has become an often-quoted example of UK corporate renewal [Goffee, 1991]. As the company recovered, grew and became profitable, it was bought by STC (Standard Telecommunications & Cables of the UK) in 1984. During the mid-80s ICL was active in a number of European Community sponsored activities—such as setting up a joint R&D centre with Siemens of Germany and Bull of France in 1984. In fact a recent count suggested that to date ICL has been party to 43 joint ventures and 146 collaborations.

In 1988, ICL bought CCI, an office systems supplier with a North American base and, in 1989, Datachecker of the US was combined with the ICL Retail Business. These reflected the strategic view that the information industry was becoming global, with much of the initiative coming from the US.

In 1989, Northern Telecom bought STC and Fujitsu bought a major stake in ICL. Fujitsu also saw the information industry as becoming global, and already had stakes in US based companies, such as Amdahl. ICL and Fujitsu now have jointly owned companies in North America and in the Australia/Pacific region. ICL's relationship with Fujitsu over the years since the early 1980s has been discussed in several books on alliances—for instance [Faulkner, 1996] and [Prahalad, 1994].

In the 1990s, ICL expanded in Northern Europe, buying a number of software houses across Europe and merging with Nokia Telecom's systems business, which brought the company factories and R&D facilities in Sweden and Finland [Mayo, 1994].

ICL as a global player

In 1995, the then Chief Executive Officer (CEO), Peter Bonfield, moved to British Telecom and the Finance & Business Strategy Director Keith Todd took over as CEO. Keith Todd had joined ICL originally in 1985 as part of the STC team.

Fujitsu now owns 100% of ICL. The companies and Amdahl work together on a number of programmes but, in financial terms, the relationship is arms length, since the intention is to re-establish a public share listing of ICL in May 2000.

As the detailed discussion will highlight, through the use of planning linked to scenarios, ICL has changed during the 1990s from a computer conglomerate to, firstly, an IT services company, through managing the portfolio (Section 5) and, secondly, a company focused on a mission "to unlock the full potential of the knowledge society for our customers" (Section 6).

Vision 2000 – 1993

Like many other companies in the IT industry, ICL had been growing with double-digit growth most years for more than a decade. But the growth masked major changes in the industry. By 1993, looking back to the ten-year forecasts made in 1986, ICL could see that it had correctly foreseen that the industry was moving toward personal computers and systems integration. What it had not foreseen was the extent to which the industry would restructure, new players would dominate niches, and how far margins would decrease as the industry restructured.

So ICL decided that it needed to take a systematic look ahead to see what opportunities and potential dangers were ahead. A project called Vision 2000 was set up as a first step in investigating the possible future ICL faced, to concentrate on forces in the industry and the ways in which the challenges of the changing needs of customers could be met. The project began quite informally, reporting to Keith Todd, Director of Finance and Business Strategy, and mainly involving three people. The author had a HQ responsibility for software and services, Malcolm Austin headed corporate planning, and Tony Oppenheim was mainly concerned with mergers and acquisitions.

There were three strands to Vision 2000.

One was looking at the outside world, the conventional PEST (Politics, Economics, Society, Technical) factors. It was a question of concentrating on the megatrends, such as what was happening in the US, as opposed to Asia, and what was happening to demographics—to get a feeling for the wider environment.

The second strand was analysing the IT industry, looking for discontinuities in sectors like consumer electronics and telecommunications, examining the competition, and listening to who was saying what about future business directions. Because the project team sensed that what had been perceived as a monolithic computer business was splitting up, it turned to the PIMS (Profit Impact of Marketing Strategy) [Schwartz,1997] database to look for other industrial parallels for the different components of the IT industry. The team used the construction industry as a comparison for the business that dealt with large projects and looked at utilities for their resemblance to the services business, and at consumer electronics for the products/technology-based business. In each case we charted operating profits, the return on investment, investment required and the management profiles.

The third strand was to understand ICL's assets. After exploring several techniques, a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) of the ICL assets, business by business, was selected. At this level, the team was able to separate out ICL corporate assets from those unique to a particular business—for instance the infrastructure across Europe versus Retail applications solutions for instore systems.

The Scenario Planning Project

At the same time as Vision 2000, the team was experimenting with scenario planning to judge how effective an approach it could be in helping the company to make informed decisions about its direction. Particular interest was sparked by Matti Keiola, on secondment to corporate planning for a year from ICL Finland. He led a scenario planning exercise, involving the Vision 2000 group.

The team met about once a week over ten weeks. The meetings were held in the office and consisted mainly of brainstorming. In creating the scenarios common trends were found covering economics, technology and the information industry. The following trends formed part of the input to Vision 2000:

- Shifts in Economic Power from Europe to Asia and North America, and within companies away from IT departments
- Technology push and its effect on the prices and performance of mobiles, desktops and servers
- The IT industry continuing to grow but becoming fragmented, and expanding into telecoms, media, publishing, education and consumer electronics.

All the scenarios were subject to the following uncertainties:

- general geo-economic/political conditions
- world GDP growth, monetary systems/Europe
- price stability (especially energy)
- armed conflicts (defence spending)

- the harmonisation of telecommunications
- the integration, or otherwise, of Europe.

Three scenarios were developed, called Stagnation, Baseline and Technogrowth.

Communication of the Scenarios

At the end of the scenario building project, the team had a much clearer view of the world and the underlying trends affecting ICL. But we had great difficulty in communicating this clearly. The problem was that initially the scenarios were presented in tabular form, spread over three pages, in very small type. These took several hours to talk through in detail and there was no useful summary.

Vision 2000 work had been discussed with Stanford Research Institute (SRI) in order to tap into their scenario expertise. SRI had found that a useful analytic framework for scenarios in technology based industries was to use as axes:

- consumer needs/demands
- business needs/demands
- the macro-environment
- delivery structures.

Each of the three scenarios was rated along these four axes. In the Technogrowth scenario, the macro-environment was strong, as was business demand, customer markets and innovation in delivery structures and channels. In the Stagnation scenario, business needs/demands would be very high, but it would all be about reducing costs, with particular demand for outsourcing. Consumer demand would be very flat, while the macro-environment was low growth and delivery structures were relatively stable. The Baseline scenario represented "conventional wisdom" with customer markets surging to be more important than business markets, medium growth of the economy, and little change in delivery structures. The resultant "spider diagram" is shown in Figure 1. This approach helped to begin to tease out the differences among the scenarios.

Using these scenarios the team were able to discuss a number of aspects of the industry which had not explored before from this angle. For example, outsourcing could be seen to be viable when the economy is booming, because people have discretion to rethink their business, and it could be viable when the economy is difficult, because people are trying to obtain efficiencies. So outsourcing would be driven by different forces in different scenarios. That was not a devastating conclusion, but on the other hand it was a useful one at the time.

Lessons Learnt

With hindsight we saw a number of reasons why we had been less successful than we would have liked.

- Because the sessions were held in the office, the team probably did not think as laterally as it should have.

- The team had not started with the questions it was trying to answer. A question to answer is needed in order to put the scenario into context.

- Presentation was initially very unhelpful. The scenarios were spread over three pages in very small type, with far too much detail but lacking any explanation of the essence of the scenario.

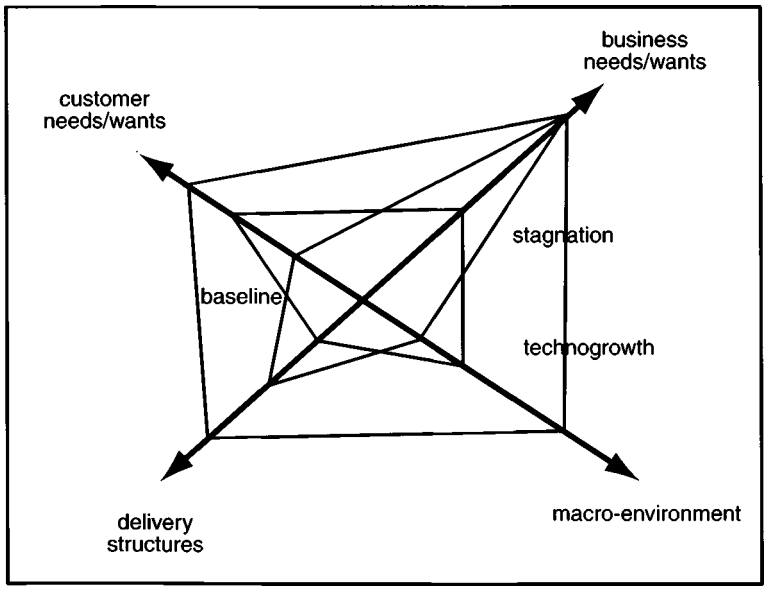


Figure 1: Displaying Scenarios

- The scenario representation using the axes suggested by SRI (consumer needs/demands, business needs/demands, delivery structures and macro-environment) made us realise the utility of using graphical techniques. It allowed homing in on the delivery structures as an area of major change, and to see that the demands on IT would be severe in both the Technogrowth and Stagnation scenarios. But we found that a very detailed commentary was required as well as a clear reconciliation of the axes. We still did not have a framework in which to place the results and did not have an immediate "picture" through which anyone who had not been through the exercise could get a feeling for what had been understood.

- Having three scenarios proved to be dangerous, particularly in an industry where people are reasonably numerate and literate. They will usually make what seems to be the sensible assumption that the middle one is the forecast, which limits the value of the scenario thinking.

Outcome of Vision 2000

Overall Vision 2000 proved its worth in helping ICL to draw up a coherent picture of where the industry was moving. In January 1994 the company was reorganised into three main business streams: systems integration for project-based business; a services-based business specialising in outsourcing; and the technology/product divisions. It was tools like competitive benchmarking and the SWOT analysis that lay the foundation for the structural change.

The trend analysis suggesting a move away from powerful IT departments helped the team to think about ICL's customer relationship strategy. The trends towards growth and fragmentation of the industry also informed our technology watch, as new industries and competitors sprang up.

But perhaps the dominant trend that was seen was the effect of technology push on the prices and performance of mobiles, desktops and servers. The team saw ICL's business needing to operate with margins very much lower than in the mainframe era, and needing to increase volumes significantly to maintain constant revenue from hardware. This led to the tackling of some of the issues described later in Sections 6 and 7.

Framework for scenarios

By the time work started on the scenario project in 1995, an approach had been developed, which combined work with a number of external organisations on research into the future, with its exploitation in house.

We worked with Global Business Network on major trends, and were a member of the Business in the Third Millennium consortium, looking at the changes as they affect individuals in Asia, US and Europe, and the effect that this has on government and business. Since then we have worked with a number of other external organisations such as:

- Promethee, the French thinktank
- The Copenhagen Institute for Future Studies
- The Futures Council of The Conference Board
- The EC ++++
- The Open University Environmental Scan project.

A number of individuals from the ICL line operations can and do participate in these research projects, as well as external consortia, to extend the vision of the future; see Figure 2. But the nature of the output is such that a key role is that of distilling the many confusing and conflicting signals and information into a form that can be used for planning. Scenarios are one way of doing this, of applying the vision, so that the organisation can implement.

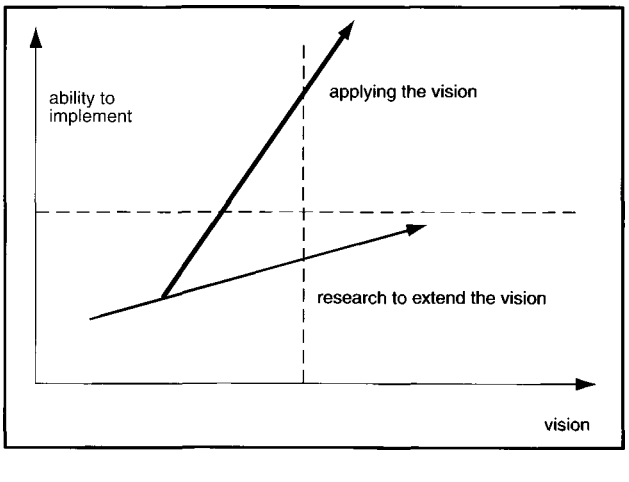


Figure 2: Futures Framework

Scenarios for Information Markets in 2005

Late in 1994 we talked to Laura Raymond of Shell about our views of the IT industry. In return the scenarios team under Ged Davies briefed us on how they do scenario planning and use the results. Much of this has now been published [Kees van der Heijdn, 1996]. In particular, their emphasis on the importance of credible scenarios which could plausibly evolve from "where we are now" was very helpful.

The organisations from which we found we got most at this stage in terms of methodology were:

- Global Business Network, which we found out about through the Internet. They run training courses, provide a flow of ideas and lateral thinking, and hold workshops several times a year, and Peter Schwartz's "The Art of the Long View" is still the classic on scenarios
- The Batelle Institute, which focuses on the health sciences/chemical industry axis, but has a general-purpose methodology.
- SRI, who also have a well-developed methodology
- IDON, who have a methodology orientated towards interactive workshops.

storylines, and the naming of our "Worlds in 2005", for which we chose Coral Reef for one, and Deep Sea for the other. The reasons for this will be discussed below.

The final stage before taking the scenarios into the company for dissemination was to validate them against the ICL portfolio—were the business units differently susceptible to the two worlds in 2005? We found that the scenarios did indeed help us to analyse the portfolio, and so we decided to disseminate them into the organisation. This section describes each of these stages.

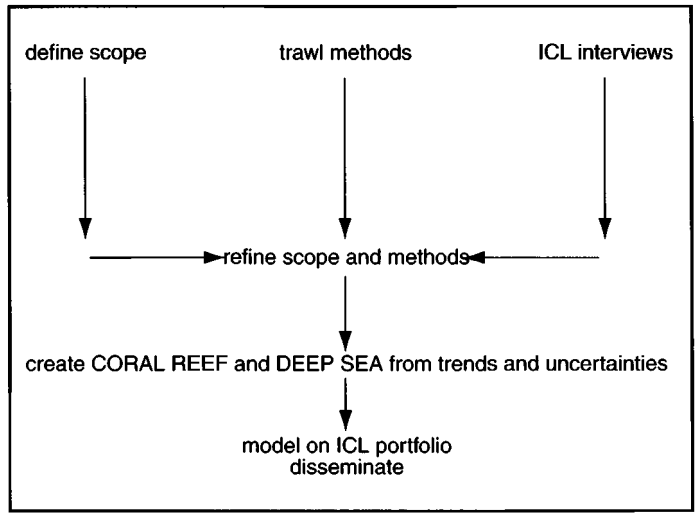


Figure 3: Scenario Process

Focal Issue or Decision

The purpose of the project was to provide a framework for analysing the ICL portfolio.

The first crucial step was to decide just what were the questions to answer. The planned process, in which five main activities were recognised, is shown in Figure 3. During the first stage we used interviews and external analysis to define the scope of the project, and at the same time investigated available methods. At the second stage, we had to decide what method to use, and refine the scope, in this case to "information Industries in 2005", with particular emphasis on Europe. It was relatively easy to assemble vast numbers of factors affecting us, and to sort them into trends and uncertainties. Much more taxing was the synthesis of these into scenarios by painstakingly grouping them & agreeing linkages and dependencies. Most taxing of all was the creation of

In some environments it might be more suitable for a senior management team, or the management team of a business unit, to engage in the process. What we decided would best meet our aims was to pull together a group of staff from headquarters who could work with the businesses to exploit the scenarios once they had been created, could if necessary modify or extend them, and then explore the implications.

The group consisted of:

- Paul Clayton: a business analyst who had done research into outsourcing and the changes in the computer services industry
- Laurent Douillet: a stockbroker and financial analyst from France who was on a mid-term break from an MBA at the Wharton School
- Jane Dowsett: an economist by background with specialist experience in market research

- Steve Parker: a business analyst with specialist expertise in the computer and telecommunications industries
- Gill Ringland: Group Executive with specialist expertise in software and the services industry.

We wanted the group to encompass diversity but also to be reasonably empathetic to the concerns of the company, so that the company's assumptions and concerns were a central part of the scenario-building exercise. We decided to take the risk that having mainly ICL people might mean that important issues remained unidentified because of cultural blindness because we wanted to explore the process, the methods, and crucially, how to use the scenarios.

The main project ran over three months, with the group meeting twice a week for half a day, as well as carrying out research between meetings. No outside consultants were used, although we did discuss some of the outcomes during the project with Professor Gareth Price of St Andrew's University and Dr Oliver Sparrow of Chatham House Forum.

The scale of the project seems typical of those creating broad-spectrum scenarios, and is qualitatively different from that needed for the workshops used to create common vision and vocabulary. And since we were concerned that there was a link to corporate planning we focused on two "near in" scenarios, rather than more, diverse scenarios.

Defining the question

Jane Dowsett carried out a round of half-hour interviews with some 50 senior staff, both old timers and those new to ICL. The interviews had two objectives: to get management to "buy-into" the project and to understand what was the burning question. In other words, sitting in 2005, what would we have liked to have known about in terms of political, economic, societal, environmental, technological and lifestyle trends ten years earlier. To unlock the answers, a set of seven questions was used. The questioning technique we used is referred to as "The Oracle" and it works on the basis that people know a great deal, but do not always know what they know.

The Vital Issues (the Oracle):

- Would you identify what you see as the critical issues for the future?—When the conversation slows, continue with the comment—suppose I had full foreknowledge of the outcome as a genuine clairvoyant, what else would you wish to know?
- What would be a favourable outcome, if things went well? Being optimistic but realistic, talk about what you would see as a desirable outcome?
- What would be an unfavourable outcome? As the converse, if things went wrong, what factors would you worry about?
- Where culture will need to change? Looking at internal systems, how might these need to be changed to help bring about the desired outcome?
- Lessons from past successes and failures. Looking back, what would you identify as the significant events which have produced the current situation?
- Decisions which have to be faced. Looking forward, what would you see as the priority actions, which should be carried out soon?
- If you were responsible, if all constraints were removed and you could direct what is done, what more would you wish to include (the 'Epitaph' question)?

We reviewed the interviews and analysed them by the factors, which were mentioned under each of the headings. What we found was that one-third of the interviewees explicitly raised, as a critical concern, what will be ICL's source of added value? The question was implicit in the answers of another third.

We thus decided that in terms of the scenarios, we should focus on the question:

> *What added value will we provide to our customers in 2005?*

We added an extra hypothesis, that the scope of added value could be bounded by, "innovate to improve their business," or alternatively by, "to minimise their risk"?

Sources of external data

In building the scenarios we found that the problem was distinguishing the relevant data. In the

Vision 2000 and other projects we had identified good sources for a wide range of IT industry data, telecoms research and for technology in general. For economic data, the Economist Intelligence Unit and the OECD provided good background.

The areas we found difficult were those relating to the wider information industry and to new developments like digital cash. For these consumer related sectors we found that the financial analysts were often the best source of both data and opinion.

Developing the List of Relevant Factors

From the interviews we identified a list of about 40 topics that people wanted to know about in the future. We added a number more from our researches, and we categorised the topics as either trends or uncertainties.

The trends reflected "best educated guesses" about directions. These would be common to all the scenarios. A judgement was taken on which trends were already incorporated in ICL's thinking and processes. For instance, the dominance of the US in IT innovation, the decreasing role of national governments in Europe and the increasing reliability of hardware were all significant, but perhaps no longer needed spelling out.

Even with a well-expected trend, there could always be surprises and questions about the pace at which the trends would develop. So that forecasting, even without uncertainties, is a fraught exercise.

Trends are mostly seen in two areas: those related to demographics and those related to technology. Surprises in demographics occur only rarely. For instance, between censuses in the US, the size and pattern of Catholic immigrant families changed the predicted demographic balance significantly. Surprises are more common in the political or social arenas. In addition to the trends we identified in 1993, we added:

- The effect of pervasive information, increasing the power of the customer and the pace of competition
- A young population in the developing world and an ageing population in the developed world
- Environmental concerns affecting more businesses
- Bandwidth explosion, exceeding the effect of Moore's law in its rate of change
- Digital information changing the sources of value.

Grouping the Critical Uncertainties

Some of the factors were inherently unknowable and not causally linked to any of the others—for instance, the occurrence of a major earthquake in the USA was not caused by any of the other factors, although it could contribute to a factor like "more terrorist actions". The factors which were not linked were called "Wild Cards", and we found that the best way of treating these was to identify where the policy for dealing with these should rest, and getting them set, rather than building them into the scenarios.

Some of the factors were seen to be important to us, but with an unpredictable outcome in ten years time. The intellectual activity to correlate these was one of the hardest of the project. For each factor, we determined whether it was positively correlated against every other factor (on a scale 1 to 3), negatively, or not at all (0). Then the factors were sorted, giving the list in Figure 4. For each of these uncertain factors we constructed a correlation matrix relating to each of the others. For example, did one increase, decrease or remain unaffected by another?

- Shift in technology innovation to Asia
- Loss of government control on information flows
- Greater economies of scale in technology
- Consumer marketing dominates
- Major breakthrough in technology
- High adoption of innovation; e.g. multimedia
- Major IT disasters; e.g. from Y2K
- High economic growth
- Fragmented IT industry

Figure 4: Scenario Factors

We then saw a pattern, in which four themes emerged.

The degree of influence/power exerted by governments, e.g. regulation/deregulation:

- What will be the balance between government-imposed regulation and self-regulation?
- Will governments regulate to protect national cultures?
- Will governments be able to control crossborder information flows and electronic commerce?

- What will be the impact on the West of economic growth in Asia?

These groupings helped us to begin to build up the story line of how the world might look in two different scenarios. We linked open cultures and trading, deregulation/less government, individual values, and innovation, because of their correlations, and built up scenarios of two different worlds, shown as inner and outer circles in Figure 5.

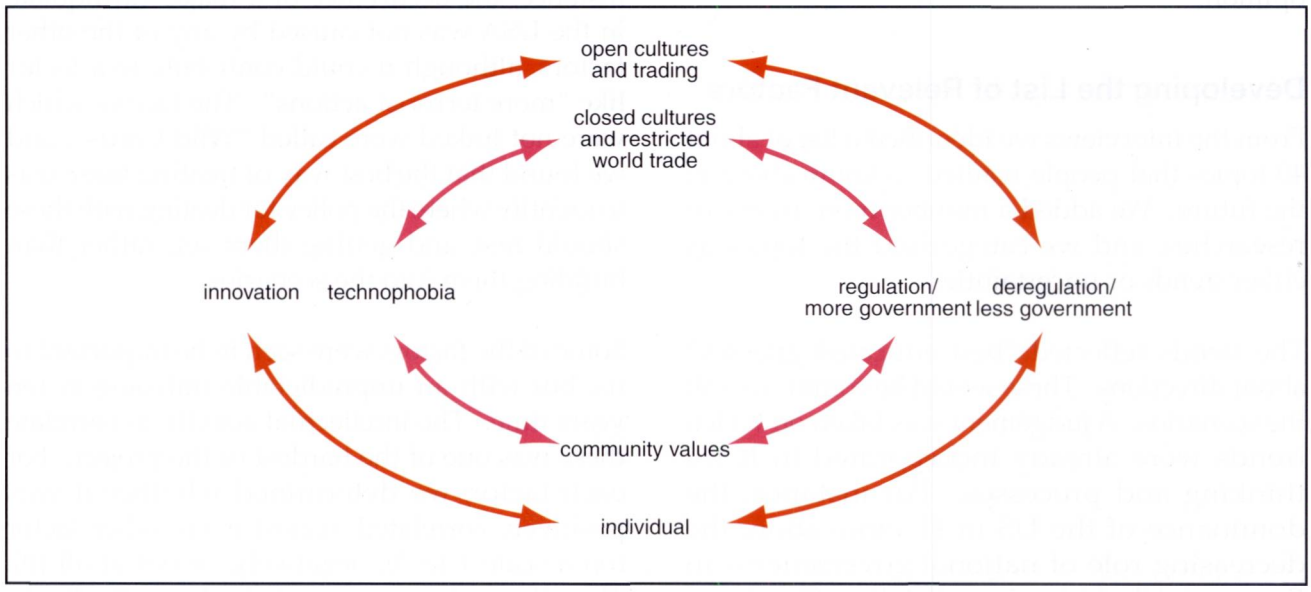


Figure 5: ICL's Uncertainties

Social values, community versus individual:

- How important will be environmental concerns?
- Will individual or community values predominate?
- What will be the level and forms of security threats?

Consumer behaviour, innovation versus technophobia:

- What will be the buying points in large organisations?
- Will consumers become tired of constant change?
- How IT literate will they be or need to be?

The shape and degree of global trade—cultures and trading:

- What will be the degree of intra-block trade and of inter-block trade?

All in the Name

We believe that one of the reasons this exercise worked so much better than our 1993 project, was that we came up with names for the scenarios which described the essence of what they were about. The names had to act as metaphors so that when we were talking about a scenario we could use the name as an evocative short-cut and give people an instant picture of each scenario and be more receptive to the detail.

At first we toyed with names like Chinese meals versus Hungarian Hot Pot. The Hungarian Hot Pot was stodgy and could describe the scenario without much innovation, while the Chinese meal could apply to the more short-lived scenario. However, these were not really satisfactory and after much discussion and brainstorming the names, Coral Reef and Deep Sea, emerged.

They seemed to fit well because of the intuitive behaviour which each describes. The Coral Reef

world is very diverse, with much visible activity and complex food chains. There are many small fish. The Deep Sea world is less diverse, with fewer species of mainly larger fish. It is a simpler world in many ways. And for people who have reflexes formed on land, both the Coral Reef and Deep Sea can be dangerous places unless the reflexes are retrained. For instance, the natural response to danger under the water is to hold one's breath and come to the surface (if you have been breathing from a scuba tank, this is a way to die painfully). The moral is—train for the new environment—and this is what scenarios are about—trying things out in a possible future environment via scenarios; see Figure 6.

In Coral Reef, a multiplicity of devices would be available to connect to a number of competing services, with price wars and confusion.

We thought that early indicators of a world behaving like the Coral Reef could include:

- Bill deregulating US markets passed in 1995, European countries meeting their deadlines, and Japan deregulating in 1999
- AT&T sells NCR or Siemens sells SNI or Olivetti sells the PC business
- Spin-offs increase relative to mergers in the media business
- Digital sells its semiconductor business to TI, and TI sells its software business to CA

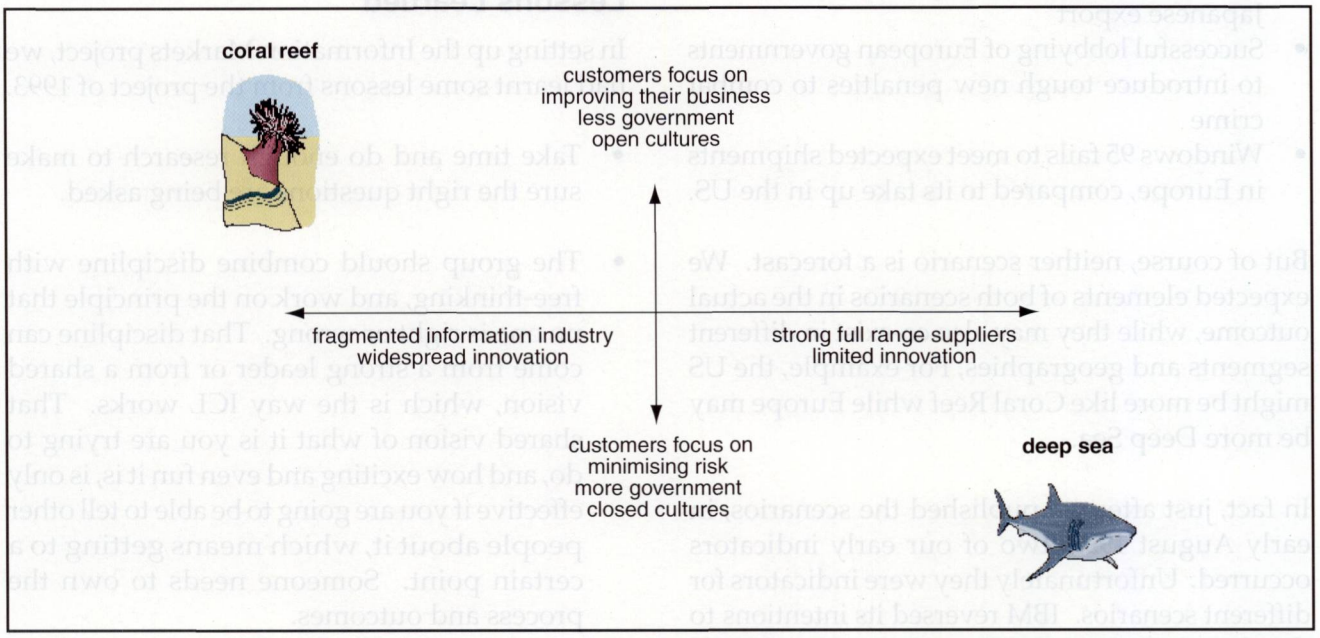


Figure 6: ICL's Scenarios

Information Markets in 2005

Coral Reef

Under the Coral Reef scenario the demanding and sophisticated customer outsources or purchases systems integration because of the potential for IT to change his business and is interested in new technology. Coral Reef is largely deregulated or self-regulated, while Deep Sea is regulated. Coral Reef exploits energy and innovation, with growth from Asia and new businesses in new areas. The competition to ICL is small, fast start-up companies.

An example of the differences is how information would be supplied to customers over networks.

- Fabrication capacity in Asia (excluding Japan) exceeds that in Japan, Europe and North America combined by 2000
- US & China establish a trade treaty in 1996
- Information Society takes off in Europe
- Microsoft and Intel constrained by anti-trust legislation.

Deep Sea

In this scenario, we see Europe and the US reacting somewhat negatively to changes in world balance. Under Deep Sea, the demanding and sophisticated customer outsources or purchases systems integration because it is not his core business. He is interested in a full range supplier taking the risk and reducing cost. The

existing (US) full range suppliers are ICL's competition.

The consumer would see a smaller range of offerings, with a lower bandwidth offering as the norm—so less possibility of movies on line. The devices would be packaged with the network, and only work with one service provider.

Early indicators of Deep Sea could include:

- The early indicators of Coral Reef are not seen; e.g. the number of mergers increases relative to the number of spin-offs
- UK, Spain & Denmark isolated at the 1996 Maastricht conference
- US imposes punitive tax on a very visible Japanese export
- Successful lobbying of European governments to introduce tough new penalties to combat crime
- Windows 95 fails to meet expected shipments in Europe, compared to its take up in the US.

But of course, neither scenario is a forecast. We expected elements of both scenarios in the actual outcome, while they may also co-exist in different segments and geographies, For example, the US might be more like Coral Reef while Europe may be more Deep Sea.

In fact, just after we published the scenarios, in early August 1995, two of our early indicators occurred. Unfortunately they were indicators for different scenarios. IBM reversed its intentions to divide into separate businesses—a sign of Deep Sea. AT&T sold NCR—a sign of a Coral Reef world.

But the analysis that we had done meant that we could make sense of this. It told us that the projected Information, Communication and "Edutainment" industry was not yet one, that telecoms was about to enter a phase of innovation and startups, while the computer industry was starting to coalesce into fewer, bigger players.

Communicating the Scenarios

In communicating the scenarios, we have already discussed the importance of a name that conveys the right picture and associations for the scenarios.

Additionally, we prepared and circulated back to the group of managers that we had interviewed:

- a summary, with an offer to come and talk through the scenarios
- a slide set
- a glossy booklet, using images to convey the excitement and dangers of taking a view of the future.

We gave briefings to the Policy & Strategy Network of corporate planners, and briefed the Client Managers and Board. We included questions related to dealing with uncertainty in the planning guidelines.

Lessons Learned

In setting up the Information Markets project, we had learnt some lessons from the project of 1993.

- Take time and do enough research to make sure the right questions are being asked.

- The group should combine discipline with free-thinking, and work on the principle that no one is right or wrong. That discipline can come from a strong leader or from a shared vision, which is the way ICL works. That shared vision of what it is you are trying to do, and how exciting and even fun it is, is only effective if you are going to be able to tell other people about it, which means getting to a certain point. Someone needs to own the process and outcomes.

- The group needs to be made up of a fairly disparate set of interests, backgrounds and approaches. Not only do you need experts on your industry, but it is also helpful to have different nationalities. We were too UK-oriented: the ideal balance for us would have been 50% UK, with representatives from the US, perhaps Japan and the Nordic countries.

- The scenarios must be relevant to the business in order to convince what can be sceptical line managers about their usefulness in creating fresher mental models.

But we found that none of the communication mechanisms worked without personal input from

a member of the team and, with that ingredient, we were able to tackle a range of applications, as discussed in the next section.

Applying Scenarios

When we developed the scenarios in 1995, we were facing a number of immediate business decisions. I have chosen to describe two of these in more detail.

Applying Scenarios to a Business Plan

We had decided to concentrate on services, moving our volume product business into Fujitsu to achieve the economies of scale. But what should we do about our contract manufacturing business, D2D? It was a services business and by then achieving significant external revenues from e.g. SUN, Camelot.

We had a business plan for D2D written by the management team as we were developing the scenarios. We needed to check that it was viable under both scenarios. D2D had particular strengths—for instance winning the EFQM Quality Prize the previous year—did this have different value in the two scenarios?

We applied the Coral Reef and Deep Sea scenarios to the business plan written by the D2D management team. In Coral Reef, the consumer electronics market booms, and networking demand explodes. Both provide ideal opportunities for contract manufacturing, but for very short runs and low cost items.

In a Deep Sea world, the demand is mostly for PCs, terminals, servers, and higher end networking in the business world. The sources of competitive advantage in this world are a low rate of returned items and the ability to change specifications on the fly.

When the D2D management team looked at the plan with the scenarios in mind, they realised that they had a default mindset very close to the Coral Reef scenario. (This is true of many people in our industry). The initial business plan was orientated towards tooling up for the Coral Reef markets, which would have involved a major investment. But when the team reconsidered their assumptions, they saw their business being dominated by Deep Sea characteristics.

With a revised business plan, we sold the business in 1996.

Investing in a portfolio

Another application of the scenarios was to a portfolio of seven potential new ventures. We used the scenarios to help us decide on the two to invest in.

To do this, we used a simple method, but one I have not seen described elsewhere. A team consisting of the scenario group plus a protagonist for each project evaluated the seven businesses and set them on a MA/C (Market Attractiveness/Capability) matrix. This is a simplified version of that used by GE and Shell, as discussed by Kotler [Kotler, 1996]. The Market Attractiveness of a business or product line is assessed using Porter's criteria; e.g. size of market, barriers to entry/exit etc.. The Capability is assessed on business specific criteria; e.g. engineering strength, geographical coverage, relationship management. The businesses are then ranked and displayed on the matrix below. Each square suggests a different management strategy.

	Capabilities		
Market Attractiveness	***Weak***	***Medium***	***Strong***
High	Double or quits	Try harder	Leader
Medium	Phased withdrawal	Proceed with care	Growth
Low	Withdraw	Phased withdrawal	Cash generation

For each of the factors in the Market Attractiveness list, the scenarios were "applied" in the sense that we asked the question:

- would the trends have a favourable/zero/unfavourable effect in 10 years time?
- what would be the effect of the uncertainty factors for each of the scenarios?

This was a tedious process but provoked useful discussion.

Since a common theme of the portfolio was taking advantage of multi-media technology, the effect of the technology trends was mostly

positive. But the effect on each of the businesses was different. The impact was to move many of the businesses to a different MA/C box "in ten years time under Coral Reef" or "under Deep Sea". See Figure 7.

So for instance, the market for Business 1 stayed attractive, but our position in Europe meant that in a Deep Sea scenario it was difficult to see how to compete.

The market for Business 2 grew in both scenarios and we could see how it played to our capabilities.

Business 3's markets became less attractive under Deep Sea but our capability was higher—were there attractive sectors of this market?

So we were able to explore the risks and make the choices.

One of the two businesses later won the IT Prize of the Year in 1996, and the other led to our work on BBC Online—so, even if the other could also have been successful, these certainly are.

Early Indicators leading to strategy development

Coming back to the early indicator "The Information Society happens in Europe"—this referred to the EC Information Society initiative with which our then CEO, Peter Bonfield, had been closely associated. Tracking this started a train of events in ICL.

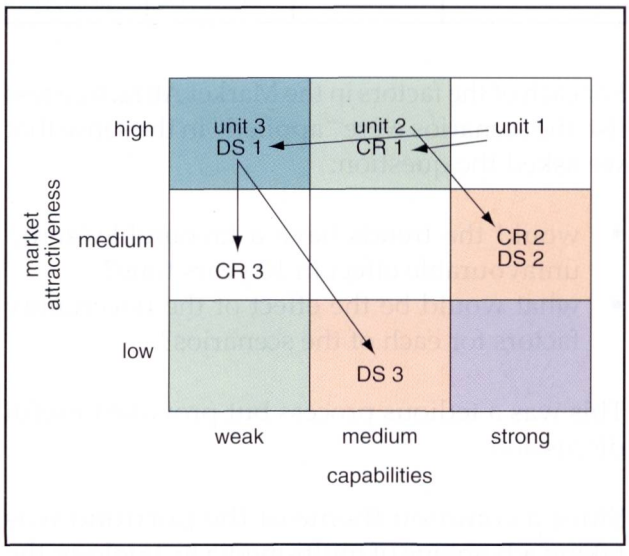


Figure 7: An Evolving Portfolio

Our new CEO, Keith Todd, was due to visit the EC in Brussels, just after the European Parliament had rejected funding for the Information Society initiative. He offered to help move public opinion forward through a set of CEO's thinktanks held with the Commissioner and his staff, and high level attendees from across Europe. This also helped us, through clarifying our ideas on how the Information Society differs from the industrial society.

The first insight that we reached was that the levers for change are different in the information and industrial societies. In the industrial society, change was often initiated by central government and large companies, whereas in the information society, local government, small companies and individuals take the lead.

And in terms of lifestyle, we see extensive changes in work, leisure, learning, trade and governance.

Following this up with research in a number of forums, we have found that there are two effects going on. One is that everybody has started to expect to "be treated like the Queen Mum". The second is that the attitudes of people under the age of 30 seem to be radically different from those held by many of the people making decisions who are over 30. The increased importance of a sense of community—no longer a geographically based concept—and the new sources of status based not on price but on scarcity, factors that are important to our customers as they meet the wishes of their customers.

So one of my current projects is to experiment with ways of sharing this understanding with our customers, helping to work through the effect on their business. We are exploring the use of virtual reality to imagine, devise and evaluate propositions. We call this the envisioning process, and it is central to our mission "to unlock the potential of the knowledge society for our customers".

The reason that I have treated this example in detail is to emphasise that a set of scenarios is a starting point, not an answer. They can provide a framework for deciding to look for certain events or trends, the analysis can prompt more questions, but nothing can substitute for thought followed by action.

Scenarios developed with external partners

Business in the Third Millennium

The Business in the Third Millennium programme was set up as a result of discussions between ICL and Stanford Research Institute (SRI), and was later run by FirstMatter, the futures off-shoot of SRI and Meta Group. It ran from 1995 to 1998.

The hypothesis behind the Programme was that digital information would have a major impact on the way business evolves—or changes discontinuously—in the next decades. We also expected that the impact might be different in Asia, the US and Europe, and so have partners in all three geographies and cultures. The Programme started in 1996 with the construction of some scenarios to describe possible ways that digital information could have an impact on society, and the hurdles to this. By the end of 1997 we had examined the roles of business, government and individuals and identified that a major new entity needed to be considered—that of community.

There are several changes from the conventional relationship that we previously assumed between business, government and customers. The first stage of the research identified the importance of the individual, who may take many roles; e.g. consumer, citizen, leader, manager. One of the characteristics of the new environment is the "unboxing" of individuals and our decreasing ability to predict individual and community behaviour.

It also identified the decreasing capability of governments to control their geography, the increasing role of NGOs and regulation, and the importance of the infrastructure—covering social aspects like education, law and order and health, as well as the media and technological infrastructure. What we carried forward to the next stage of the programme was the importance of the community concept, driving many of the changes in individual behaviour.

Our four scenarios were developed over three days of interactive work after earlier interviews and discussions with participants. The discriminator axes were business agility and community, where community was assumed to be strongly linked to individual and lifestyle choices. Business agility was assumed to drag with it government agility and agility in the social infrastructure.

In Liberation Works, a sense of community develops to provide a rich and well balanced social environment, and business practices remain those of the 1990s. This is a "Greenpeace" world, which reconciles global and ecological concerns with business and individual interests.

In Gung Ho, there is also a sense of community but, since the business and social infrastructures have adapted, these may be less dependent on geography. Business has found ways of meeting their customers' new wants but, because of the lags in the adaptability of government, have had to take over some of the regulatory roles of government.

MegaCorp is a world in which the sense of community and governments has not been able to evolve, and so the large companies have established new rules. This is the nearest to Aldous Huxley's "Brave New World".

Finally, Organisation Rules. This scenario was originally named "France", but equally applies to Japan. Here the sense of community comes from family or work, both of which are hierarchical in their nature.

We used the BIT3M scenarios to develop our research portfolio, but include them here to indicate the change in the nature of concerns, which we have seen over the decade. In 1993 we had macro-economic and purely IT related concerns, in 1995 the concern was the shape of information markets and in 1998 the adaptability of organisations.

Chatham House Forum

The Chatham House scenarios, which are nominally for 2020, are built around the drivers of the industrialised world. While this may seem to ignore the potential of Russia, China, and the tiger economies of Latin America and Asia, it reflects the reality that collectively the industrialised world represents over 80% of the world's GDP and this is not expected to change significantly under any scenario. The four key geographical/cultural groupings are taken as the US and its

clients, the European group and their dependent sphere, the industrialising nations (or perhaps what is now typified as "emergent markets") and the populous poor nations.

The scenarios are based around two unknowns which will set the agenda if there is not a complete breakdown. The first is the ability of commercial organisations to change: will they be able to evolve, or to break up and start up, to meet the environment of continual change? The second is the ability of individuals and society to learn to manage complexity and the new rules, often without role models to help.

We have chosen to build the ICL scenarios on the basis of the three Chatham House Forum scenarios. This is for three reasons:

- they are well researched and soundly constructed, providing a macro-economic framework
- they have evolved over three years and many iterations
- they are providing a strong input to the UK Government's Foresight Programme.

One limitation is that they do not cover two extreme scenarios that are potentially of interest:

- explosive and sustained growth fuelled by new economic rules, as espoused by many US commentators. However we do include a scenario, Markets, which has strong growth in the short (five year) term.
- melt-down of the financial system, with world-wide stock markets retreating to less than half of their current value; i.e. the world suffering in much the same way as Japan has done over the past decade. However we do include a scenario, Storms, in which in the short term, markets are in a crisis of confidence and governments are unable to see how best to react.

In our ICL Storms scenario (based on Atlantic Storms/Rough Neighbours) the answer to both questions is broadly no and the resulting world retreats into nationalism. In the early years of the 21st century, there are large sums of money washing across national boundaries, put and withdrawn at a hectic pace. This causes a crisis of confidence and governments in Europe suffer as funds are withdrawn as the competitive position worsens. Unemployment among the unskilled is high, particularly in Europe. Europe and Japan suffer economically and governments "return to their roots", defending national trade barriers. The US fares better economically and by 2010 is pursuing a free market model excluding Europe and Japan.

In the ICL Markets scenario (based on Market Quickstep/Faster-faster) the answer to both is broadly yes, for at least enough organisations and individuals to fuel initial explosive growth. This scenario is not inclusive and provides rapidly fluctuating fortunes for organisations and people. The nodes of commerce are "city states" held together by a trading network. Capital is invested for short-term returns and new advances constantly erode value. This scenario is expected to contain periods of growth and correction, with governments providing a reactive environment.

In the ICL Third Way scenario (Wise Counsels/Post-Industrial Revolution) the answer to both is broadly yes, but with a difference. This is that business and subsequently government attain mechanisms that provide resilient structures based on harnessing knowledge. Governments provide a stable regulatory regime, with agencies providing public services (health, roads, etc.). Rapid economic growth follows, with high returns attracting investment. The industrialised nations will jointly implement a regime, through their economic dominance, which aims to change or eliminate "unsatisfactory" governments.

Using these in ICL

We have used these in two significant applications, one to advise our European Strategy Board on the range of potential developments to consider, and secondly to direct our business offerings strategy.

For our European Strategy Board, which consists of some of the ICL Executive Management Committee and a number of non-Executives, we used a standard briefing paper approach. The Board was not interested in the provenance of the scenarios but was very interested in the short term implications; for example, which scenario would anticipate the appreciation of the Euro, and which its depreciation. We are currently

developing the scenarios to help think about the relative strengths of national versus regional groupings, and to provide early indicators.

The engineers and marketeers who used the scenarios to think about business offerings became interested in the scenarios themselves. The format we used was:

Day 1

- Brief on trends
- Divide into groups to brainstorm the effect of trends on business offerings
- Brief on scenarios
- Divide into groups to internalise scenarios; e.g. write newspaper front pages

Day 2

- Divide into groups to brainstorm the effects on business offerings for each scenario
- Develop a timeline for significant new offerings each year
- Develop a timeline for significant threats to existing or new business offerings each year
- Draft the recommendations and plan how to follow up.

The jury is still out on how successful this has been in anticipating threats and opportunities.

Conclusions

Our first scenario project in 1993 was useful primarily because it alerted us to the magnitude of the trends, which were about to overwhelm us, rather than the uncertainties we explored in the scenarios. Our industry at that time was dominated by the changes in price/performance and the implication of that on the potential applications, the penetration of IT, and the cost structure of the supply industry.

Our second major project, in 1995, concentrated on our added value to our customers, and the shape of Information Markets in 2005. This project provided the framework to help us revamp our portfolio through disposals, and to invest in new businesses. It has been used widely to help management teams think about their business, and I was delighted that a Client Director explained the scenarios cogently to one of our customers as a basis for a strategic discussion with him.

The third "wave" of scenario is based on the reactions of organisations and individuals to change, whether represented by organisational adaptability, national structures, or individuals' capability and/or desire for change. Our current view is that individuals are proving more adaptable to the knowledge society than many European organisations and this has become one of our key strategic issues.

In this paper I have concentrated on our use of scenarios in association with planning. What I have not had space to consider is an equally important use of scenario building exercises, as a team activity. In this mode, a well-ordered process and timescale of the discussion serve to de-fuse current management tensions, and allow for creative and non-defensive thinking.

Bibliography

BURTON, CHRIS, " 'Baby's' Legacy—The Early Manchester Mainframes," ICL Systems Journal, Vol 13, Issue 2, Spring, 1999.

BUZZELL, Robert D. & GALE, Bradley T., "The PIMS Principle", Free Press, 1987.

CAMPBELL-KELLY, Martin, "ICL, a business & technical history", OUP, 1989.

CHATHAM HOUSE FORUM, "Open Horizons", The Royal Institute of International Affairs, 10 St James Square, London, SW1Y 4LE.

GOFFEE, Rob et al, "Case Study—International Computers Ltd", London Business School, 1991.

FAULKNER, David "International Strategic Alliances", McGraw Hill, 1996.

KOTLER, Philip et al, "Principles of Marketing", Prentice Hall, 1996.

MAYO, Andrew & LANK, Elizabeth, "The Power of Learning, a Guide to Competitive Advantage", IPD, 1994.

PRALAHAD, C.K., and HAMEL, Gary, "Competing for the Future", Harvard Business Press, 1994.

SCHWARTZ, Peter, "The Art of the Long View", John Wiley, 1997.

RINGLAND, Gill, "Scenario Planning", John Wiley, 1997

RINGLAND, Gill, "Why we get forecasts wrong", Entretiens Science & Defence, November 1998.

van der HEIJDN, Kees, "Scenarios, the Art of Strategic Conversation", John Wiley, 1996.

Biography

Gill Ringland's role at ICL Headquarters has covered the period since 1990, as ICL has been taken from a computer conglomerate to an online services provider in the Knowledge Society. During this time her strategic roles have covered software, services, and the current role as futurist and scenario planner. Her current interests include communicating the challenges of the Knowledge Society, through visual as well as verbal means, using virtual reality and software to help the creative process.

She graduated as a physicist from Bristol University and worked on the theory of liquids using early computers. After academic appointments at Berkeley and Oxford Universities, she joined CAP (now Sema), afterwards Inmos, Modcomp, and then ICL. At ICL she has had at various times responsibility for database, network management office systems products and internal IT before her current role.

She is a Liveryman of the City of London and the Worshipful Company of Information Technologists, an alumnus of Stanford Business School, a Chartered Engineer, a B.Sc and M.Sc, an FBCS and a MIEEE, and serves as a Council member of the ESRC. She has published articles and conference contributions on a number of technical and business topics. Her book "Scenario Planning: Managing for the Future" is a "why to", "when to" and "how to " for managers, and was published in 1997.

The ICL Policy Scheduler for Windows

Ben Thornton

ICL, Manchester, UK

Abstract

There is an increasing trend towards centralising control back to the data centre, including the consolidation of many small systems on to fewer larger servers with enterprise qualities—*high availability, predictable performance and dependable security.*

As enterprises need to run an ever increasing number of Windows applications in their data centres, ICL has brought its patented policy scheduler technology to the Windows environment. This article describes:

- the problem being addressed
- how standard Windows scheduling works
- how the ICL Policy Scheduler can help
- results from the scheduler in action.

Introduction

Downsizing to smaller systems distributed through the organisation promised user empowerment, inexpensive, flexible applications and rapid deployment. For many, the reality has been somewhat different, with high maintenance costs, lack of standardisation and loss of control.

A counter-trend is increasing in significance, that of centralising control back to the data centre, consolidating many small systems onto fewer larger servers with enterprise qualities—*high availability, predictable performance, dependable security.*

Many enterprises need to run an ever increasing number of Windows applications in their data centres, using a limited amount of floor space and a limited number of staff. Building on many years' experience of providing solutions for the data centre, ICL has now brought its patented policy scheduler technology to the Windows environment to help them achieve this.

Why Consolidate?

There are two main benefits of consolidation—*standardisation* across the enterprise and *sharing* of resources. Consolidation is not a requirement for standardisation, but can be its most valuable side-effect—there are many benefits, for example, in implementing a standard, centrally managed email system across an organisation, such as the ability to share documents or to consult each other's diaries. Consolidation of a distributed, heterogeneous system into a standardised, central system can be attractive for this reason. More obviously, consolidation provides benefits in sharing resources—system resources, peripherals, back-up and support staff. These shared resources can have better enterprise qualities than is economical for a large number of smaller servers.

These benefits are particularly clear in the data centre, where applications are by their very nature *data centred*—consolidating multiple applications on the same system can maximise the efficiency of accessing that data, while making it easier to manage.

So, why not consolidate?

Barriers to Consolidation

There are three main barriers to consolidation.

- Application availability:
  Applications which do not run on the same version of the operating system cannot be consolidated (e.g. one requires NT4 and another requires Windows 2000).
- Application compatibility:
  Applications which require mutually exclusive system configurations, or different revisions

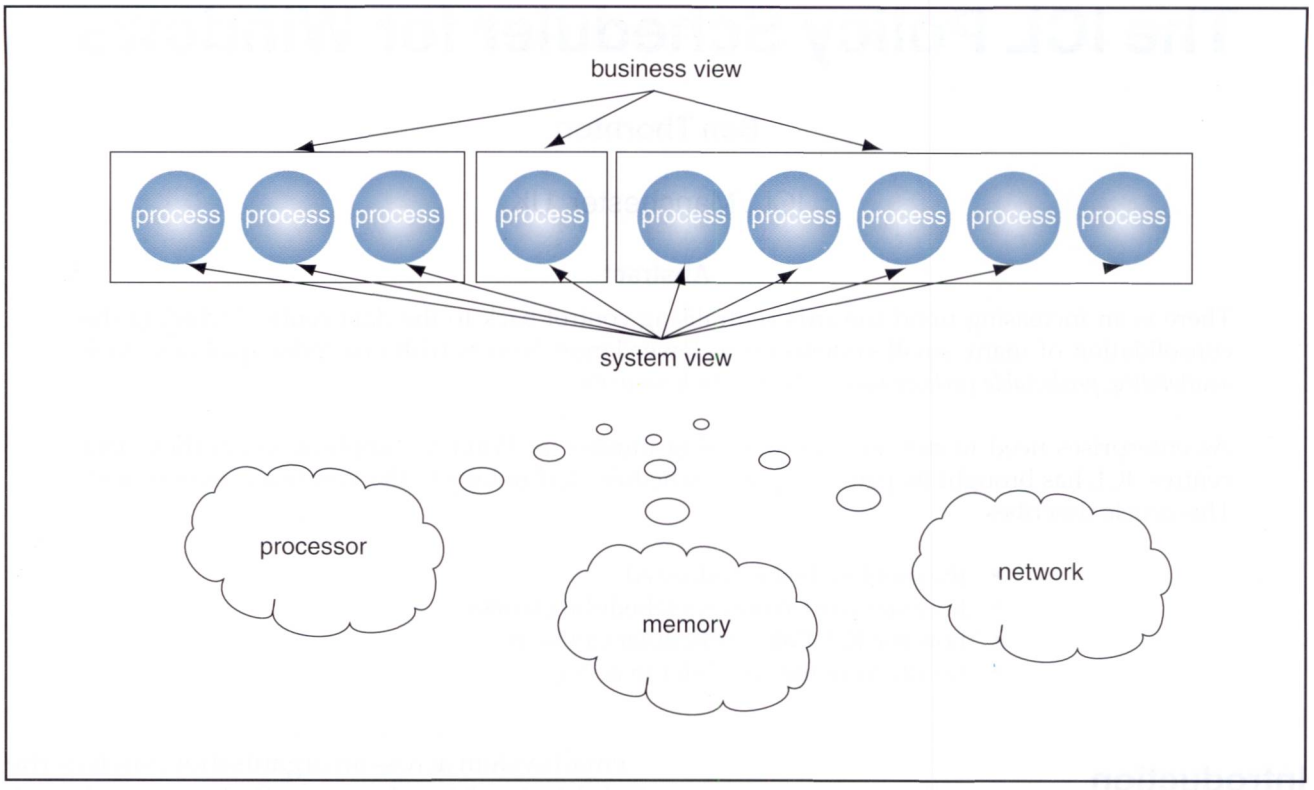


Figure 1: Business and System View

of software cannot be run together (e.g. one application may be written to use SQL Server 6 specific interfaces, while another might require SQL Server 7).

- Resource management:
  If applications are available and function together, they still may not deliver the required levels of service when running concurrently, due to the way in which shared resources are managed. Many applications in the Windows environment are designed to run on their own separate platforms. This presents a real problem for organisations that wish to reduce the number of systems that they need to manage.

For applications which will function together on the same platform, the ICL Policy Scheduler can address some of these resource management issues.

The Resource Management Challenge

The resources available on enterprise servers—time running on the processors, memory, access to peripherals such as disks, space in file systems, etc.—are distributed among the various applications running on the system, and their users.

The System View

The operating system has the task of distributing these resources among the application components running on the system from one millisecond to the next. This could be done on the basis of "first come first served", or on some concept of "fairness", or even by specifying explicitly which application components can use which resources. In Windows, an application with ten processes will receive ten times the resources of one with a single similar process. See Figure 1.

The Business View

The business requires resources to be allocated in order to meet business priorities such as one application being more business critical than another, or one group of users having priority over others during a critical period (the finance department at year end, for example). An application with one process may be as important from this point of view as one with ten.

The challenge is to make efficient use of system resources while meeting the priorities of the business.

Fair Scheduling

Windows attempts to schedule resources fairly among the threads running on the system. Processor time is scheduled fairly between threads of equal priority while higher priority threads are given preference over lower priority ones. As most threads on the system start with "normal" priority, in practice Windows implements a "fair" scheduling algorithm.

Thread scheduling

Threads are scheduled on the basis of their current priority—Windows attempts to schedule the highest priority threads to run on the available processors at any one time. Threads are selected from a queue of runnable threads for each priority level—those in a queue at a lower priority will only run when there are no runnable threads in the higher priority queues. See Figure 3.

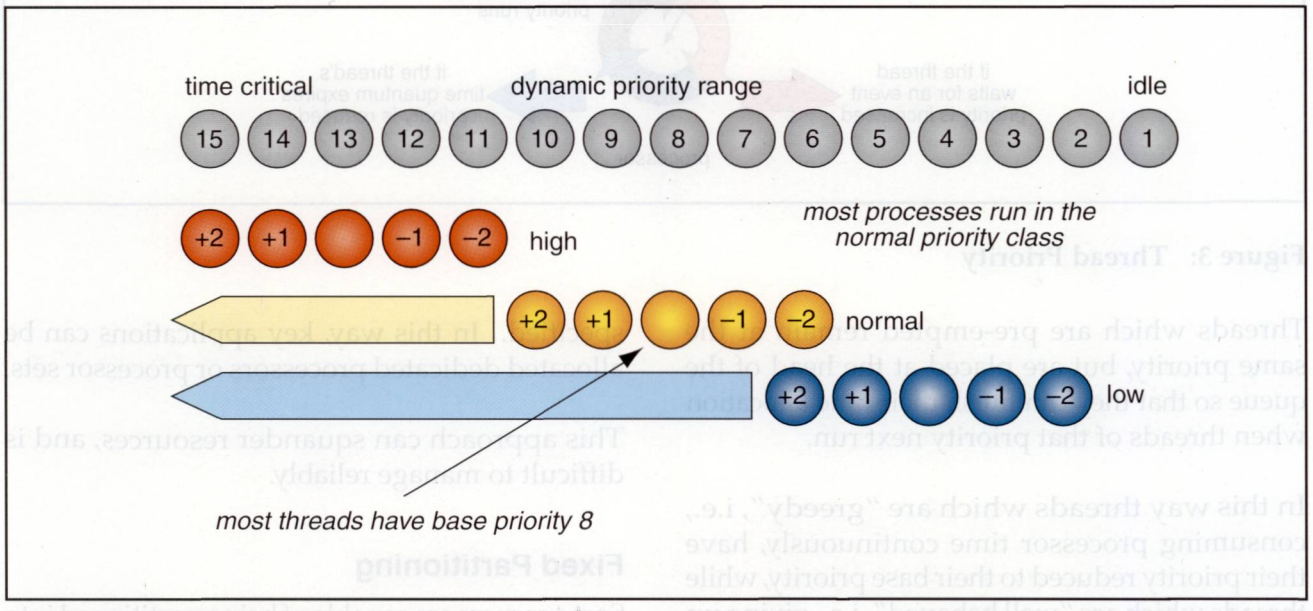


Figure 2: Process Priority

Processes, threads and priorities

Applications under Windows run within a number of processes (very often just one), each containing one or more threads (typically several, and often dozens). Processes provide an environment within which threads can run, while threads actually execute code. See Figure 2.

Each process has a base priority, set by its priority class. This is set to "Normal" (by default) for almost all processes. Each thread also has a base priority, set by the base priority of the process within which it runs, combined with a relative priority from -2 to +2 (but almost always set to the default of zero).

The priority of a thread can be increased above its base priority, after which it will fall back over time. The value at any one time is called its current priority.

Threads give up the processor for one of three main reasons—waiting on an event (giving up the processor voluntarily to wait for something, e.g. user input), completing their allotted time quantum (using up their "turn") or being pre-empted by a higher priority thread.

Threads which wait on an event have their priority increased, so that they will be better able to deal with the event when it arrives. The increase in priority depends on the nature of the event—longer waits for slower devices tend to have larger increases in priority.

Threads which complete their time quantum have their priority decreased, down to a minimum of their base priority, and are placed at the back of the queue (of runnable threads) at that priority level.

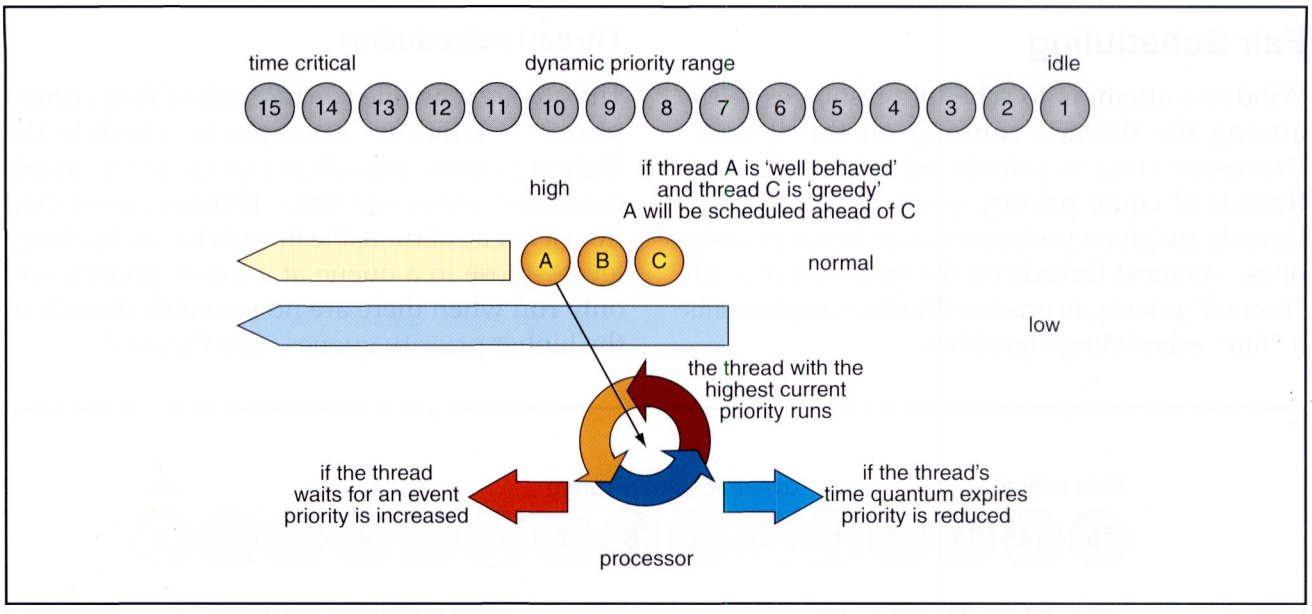


Figure 3: Thread Priority

Threads which are pre-empted remain at the same priority, but are placed at the head of the queue so that they can finish their time allocation when threads of that priority next run.

In this way threads which are "greedy", i.e., consuming processor time continuously, have their priority reduced to their base priority, while threads which are "well behaved", i.e., giving up the processor to wait for events, have their priority increased.

Fairness

This scheme is fair, but not necessarily what is required to reflect business priorities—often a greedy application (e.g. a database server) is the most important for the business. One option is to run such applications on stand alone servers, so that they do not have to compete with any other applications. Unfortunately, this can result in a proliferation of servers, leading to higher management costs.

What alternatives are there to fair scheduling?

Fixed Scheduling

Processor Binding

Threads can be bound to (sets of) processors by setting their affinity mask, causing Windows to restrict them to running only on the processors specified. In this way, key applications can be allocated dedicated processors or processor sets.

This approach can squander resources, and is difficult to manage reliably.

Fixed Partitioning

Some servers are capable of being partitioned into several distinct systems, each running a separate image of the Operating System. This approach can be valuable in reducing the number of physical systems that need to be managed, and in allowing some sharing of peripherals.

At present, such facilities are not available on Windows servers, although some systems of this type are likely to be introduced over the next few years.

Percentage Systems

Some Operating Systems use a system whereby a minimum allocation of CPU time can be set for different classes of work—this kind of partitioning will be possible in Windows 2000 Data Centre Edition. This can be useful in some situations, but raises the question of how these allocations are set; analysing the resource requirements of applications is a very complex and costly business. As workloads change, the allocation appropriate to each also changes (the changes that occur when a clustered system fails are a good example of this).

For many similar workloads this can work well, but the "fair" approach is satisfactory for most of these cases.

All the above schemes are ways of partitioning the system in a fixed way—fixed scheduling. They do not respond dynamically to changes in resource requirements and are difficult to manage reliably. Is there another alternative to fair scheduling?

Policy Scheduling

Policies

On OpenVME, ICL's mainframe Operating System, the word policy is used to describe types of workload, for example, on-line transaction processing, development users and batch processing. For each policy, priority ranges, memory quotas and other resource boundaries are defined to allow the use of the system by a large number of different applications to be managed. This protects the level of service provided for business critical work.

The ICL Policy Scheduler introduces this concept to Windows by defining policies, and using them to control the allocation of processor time.

Policy—a New Concept

Windows has no concept of collections of processes to define an application or an application type. Such a concept is needed in order to capture business priorities such that the system can implement them.

Processes Map to Policies

The ICL Policy Scheduler introduces the concept of policies to address this. A database contains the allocation of processes to policies such as "on-line", "batch", "back-up" or whatever is required, on the basis of process name.

Policies Map into Priority Classes

The same database maps these policies onto the priority classes, defining how each of them is to be allocated processor resource. This "double mapping" (from processes to policies, then policies to priority classes) allows application types to be captured and then left unchanged, while the mapping of these applications onto the available resource can be managed independently.

Dynamically Updated

The ICL Policy Scheduler service uses this database to adjust dynamically the classes to which processes are allocated, as the system runs.

Unfair Scheduling!

Each application is therefore mapped to the appropriate priority class, giving it preference over applications running in the lower classes. Those within the same class are still scheduled fairly. The processor is allocated to the thread with the highest priority across all classes.

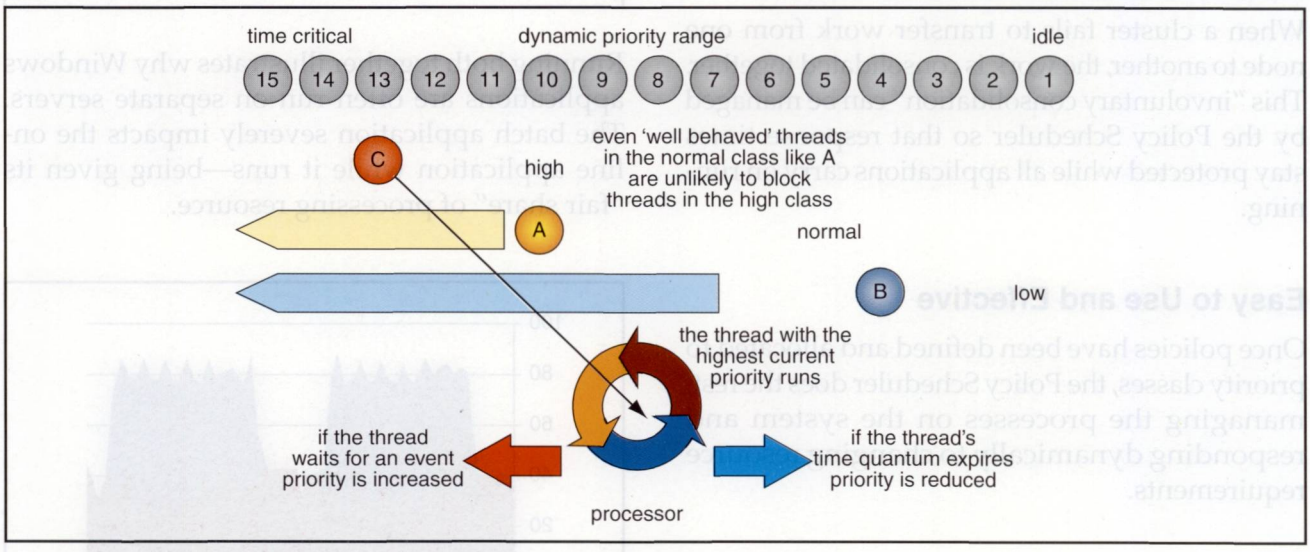


Figure 4: Allocation of Resources under Policy Scheduling

Benefits

Connects the Business to the System View

Policies describe a business view of the different applications on the system, while Priority Classes are a system construct, used to allocate processor time. The Policy Scheduler connects these together, allowing the system to apply business priorities.

Makes Full Use of the IT Investment

Instead of allocating separate servers to every application, the Policy Scheduler allows spare processor time to be used safely by lower priority applications.

Protects Response Times

The scheduler protects the response times of key applications by ensuring that applications running in selected policies are given priority for managed resources.

Enables Consolidation

By providing a tool for managing the processing resources allocated to multiple applications, some applications which would not work well together under a "fair" scheduling scheme can be consolidated together. This reduces the costs of managing separate servers while maximising the return on IT investment.

Simplifies Management of Clusters

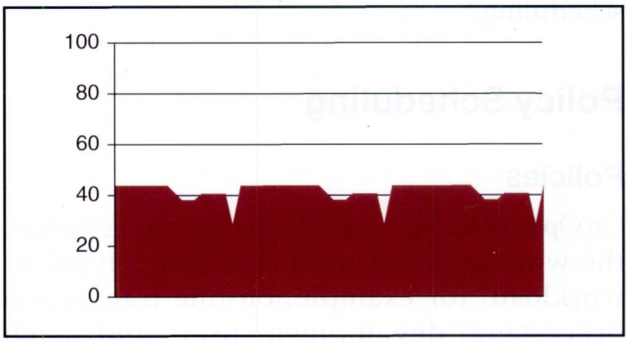
When a cluster fails to transfer work from one node to another, the work is consolidated together. This "involuntary consolidation" can be managed by the Policy Scheduler so that response times stay protected while all applications carry on running.

Easy to Use and Effective

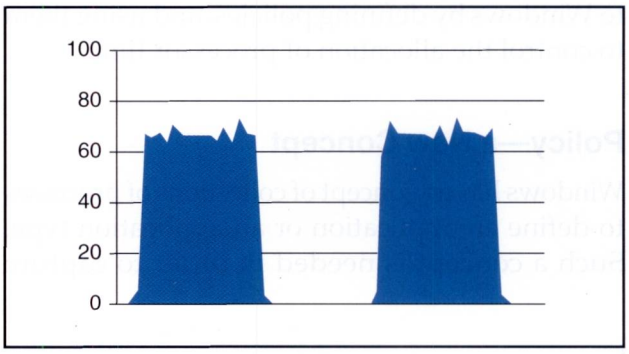
Once policies have been defined and allocated to priority classes, the Policy Scheduler does the rest, managing the processes on the system and responding dynamically to changing resource requirements.

Makes Full Use of the IT Investment

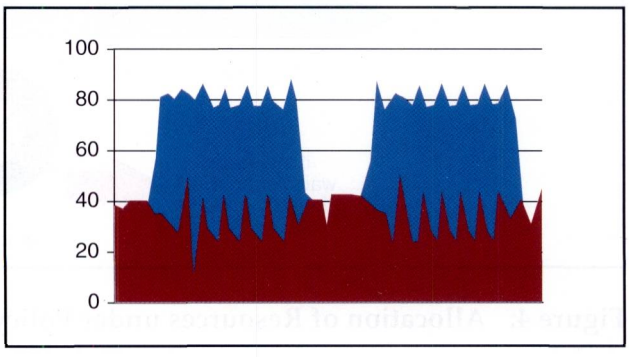
The graph below shows an application running on a Windows system, using about 40% of the available processor resource. This represents an on-line application and, for such business critical applications, it is imperative that this level of resource continues to be available.



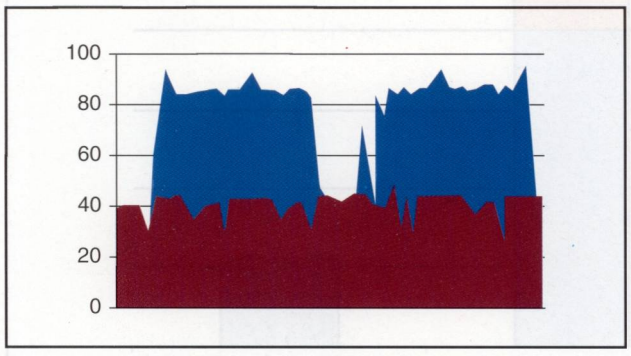
Running a different application (twice in succession) provides the following graph of processor utilisation. This represents a batch job—perhaps the provision of a report, which has no on-line users waiting for its output. This job uses almost 70% of processor resource when running.



Running both together illustrates why Windows applications are often run on separate servers. The batch application severely impacts the on-line application while it runs—being given its "fair share" of processing resource.



Using the Policy Scheduler allows the on-line application to receive the resources that it needs, while the batch application uses up the remainder. The business critical application is protected and both applications can run concurrently on the same server.



Protects Response Times

To prove the effect of the Policy Scheduler on response times for a realistic workload, an experiment was set up using a Microsoft Exchange mail system. In a production environment, there is a commitment to Service Level Agreements (SLAs). An SLA would typically specify a response time target in which 95% of requests should be met—this is called a 95th percentile response time. For this system, a mean of one second and a 95th percentile of four seconds might be set.

Running on its own, response times of less than one second were achieved for 96% of transactions, and over 99% achieved less than four seconds. This meets the SLA without problems. Typically, other work would not be run on such systems for fear of reducing the response times of on-line users, and breaking the SLA. What happens if you do?

Fair Scheduling Allows Batch Work to Interfere

The runs were repeated with another workload—a heavy batch application—running alongside the Exchange system, without the use of the Policy Scheduler.

Response times increased significantly, with 7% of transactions taking over four seconds (and some taking very much longer). Note that this is not impacting performance for 7% of the user population, but rather all users for 7% of the time; in other words, delivering unpredictable performance. This lack of predictable performance is often far worse than poor absolute performance. Compare this to a train service—would you prefer a reliable service or an unreliable one which was 20% faster?

The traditional way to deal with such interference would be to separate the applications, running them on different servers.

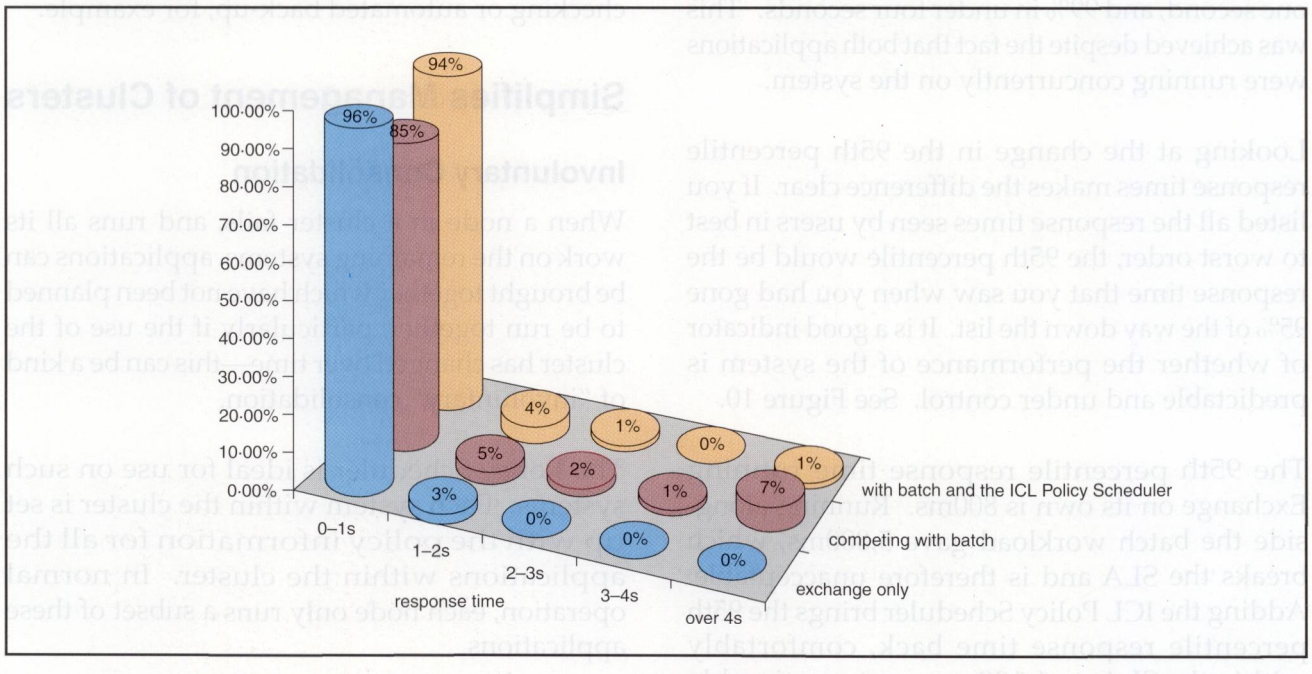


Figure 9: How the Policy Scheduler protects response times

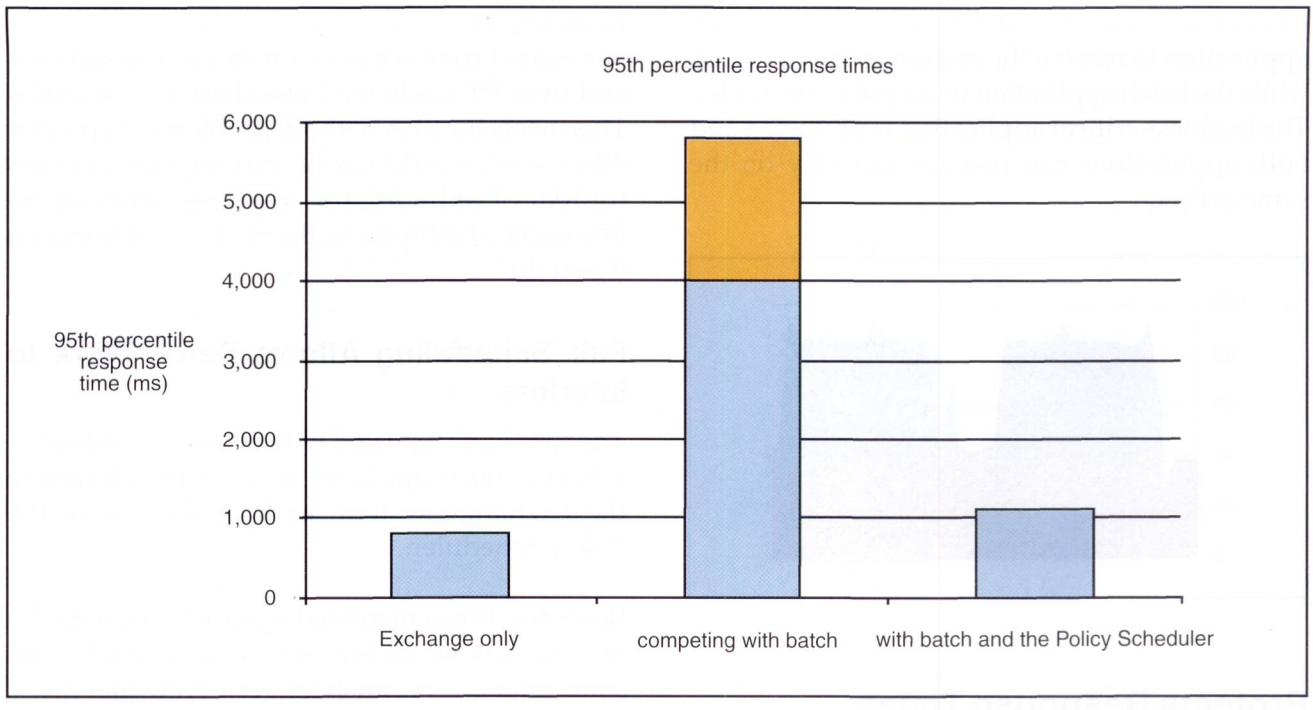


Figure 10: Response Times

Can a reliable service be provided on a single server?

A Reliable Service

Using the Policy Scheduler, the batch processes were placed in a lower policy than the Exchange work. When the runs were repeated again, the response times for Exchange were recovered—with 94% of transactions completing in less than one second, and 99% in under four seconds. This was achieved despite the fact that both applications were running concurrently on the system.

Looking at the change in the 95th percentile response times makes the difference clear. If you listed all the response times seen by users in best to worst order, the 95th percentile would be the response time that you saw when you had gone 95% of the way down the list. It is a good indicator of whether the performance of the system is predictable and under control. See Figure 10.

The 95th percentile response time running Exchange on its own is 800ms. Running alongside the batch workload gave 5,800ms, which breaks the SLA and is therefore unacceptable. Adding the ICL Policy Scheduler brings the 95th percentile response time back, comfortably within the SLA, to 1,100ms—not a noticeable difference for the users.

This work demonstrates that the ICL Policy Scheduler can protect the response times of on-line applications against being extended by the resource demands of less business-critical workloads.

These other workloads may be associated with the same business applications—it might be that on-line work needs to be protected from the resource demands of report production, virus checking or automated back-up, for example.

Simplifies Management of Clusters

Involuntary Consolidation

When a node in a cluster fails, and runs all its work on the remaining systems, applications can be brought together which have not been planned to be run together, particularly if the use of the cluster has changed over time—this can be a kind of "involuntary" consolidation.

The Policy Scheduler is ideal for use on such systems. Each system within the cluster is set up with the policy information for all the applications within the cluster. In normal operation, each node only runs a subset of these applications.

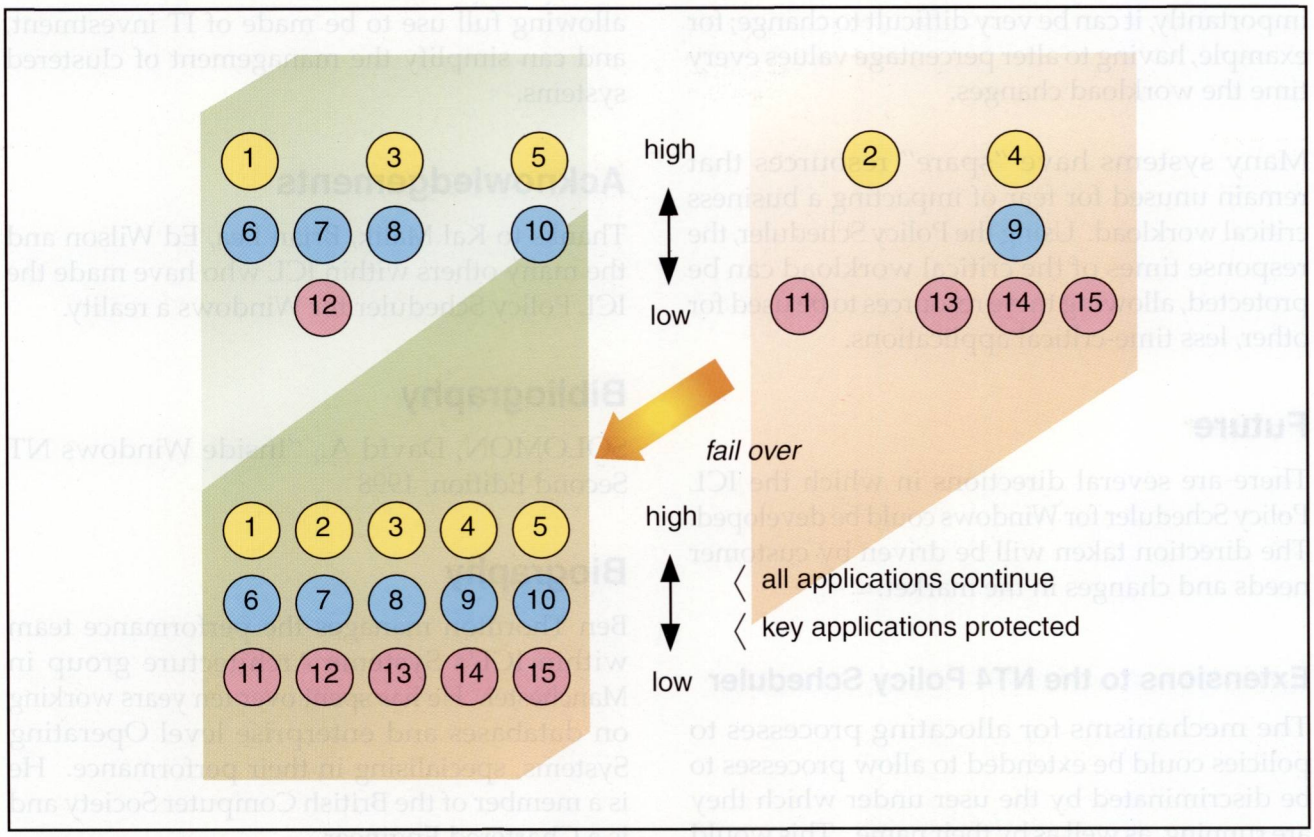


Figure 11: Protection of key applications

If a node of the cluster should fail, and its work fail over to another node (using standard clustering mechanisms), the scheduler will ensure that the extra work is given the appropriate level of priority.

All Applications Keep Running—While Key Applications are Protected

Where there is any spare resource in the failed over case, applications in the lower priority policies can take advantage of this, continuing to run, albeit with a reduced level of service. Response times for key applications in the higher priority policies remain protected by the scheduler.

In many clustered configurations, the less business critical applications are abandoned completely in the failed over case. This is done to ensure that the most important applications can be guaranteed to receive adequate resource. With the Policy Scheduler, this is no longer necessary—giving less priority to the policies in which less business-critical applications run will ensure that they cannot lock out key applications from access to managed resources.

Easy to Use and Effective

Policy scheduling allows business priorities to be used in ranking the processes on the system. This eases consolidation, where applications which would not run together on Windows, when "taken out of the box" (perhaps because one of them blocks out the other), may run together under the scheduler.

On-line response times can be protected from the interference of other applications by placing them in appropriate policies. This gives more predictable performance levels for those applications most important to the business.

On clustered systems, the policy scheduler can be used to ensure that the system keeps going in the event of node failure, with all applications giving some level of service, while response times are protected for key applications—making clusters safer and easier to manage.

Unlike the fixed scheduling approach, there is no fixed partitioning of resources (in hardware or software) which can lead to wasted processor time and can be complex to set up. More

importantly, it can be very difficult to change; for example, having to alter percentage values every time the workload changes.

Many systems have "spare" resources that remain unused for fear of impacting a business critical workload. Using the Policy Scheduler, the response times of the critical workload can be protected, allowing these resources to be used for other, less time-critical applications.

Future

There are several directions in which the ICL Policy Scheduler for Windows could be developed. The direction taken will be driven by customer needs and changes in the market.

Extensions to the NT4 Policy Scheduler

The mechanisms for allocating processes to policies could be extended to allow processes to be discriminated by the user under which they are running, as well as by their name. This would enable administrators, for example, to be given priority over other users running the same code.

Integration With Windows 2000

The scheduler could be enhanced to build on the new features within Windows 2000, in particular the Data Centre Edition, providing more advanced capabilities and better integration with the Operating System.

Management of Further Resources

Resources other than processor time could be managed by the scheduler, such as memory usage. This would allow even further control over system resources.

These changes will take place within an integrated programme providing enterprise capabilities on top of industry standard components.

Conclusions

The ICL Policy Scheduler for Windows provides a mechanism to connect a business view of priorities to the system view, allowing response times to be protected for key business applications. This protection is an enabler for consolidation, allowing full use to be made of IT investment, and can simplify the management of clustered systems.

Acknowledgements

Thanks to Kal Malik, Brian Lea, Ed Wilson and the many others within ICL who have made the ICL Policy Scheduler for Windows a reality.

Bibliography

SOLOMON, David A., "Inside Windows NT Second Edition, 1998

Biography

Ben Thornton manages the performance team within ICL's Systems Architecture group in Manchester. He has spent over ten years working on databases and enterprise level Operating Systems, specialising in their performance. He is a member of the British Computer Society and is a Chartered Engineer.

Performance—an Engineer's Guide

Stuart Forbes and Ben Thornton

ICL, Manchester, UK

Abstract

A common misconception is that performance problems rarely happen nowadays, because modern hardware is so fast and so cheap. Unfortunately, this is very far from the truth and an alarmingly high proportion of IT projects are badly affected by performance issues. Another misconception is that performance work requires a deep understanding of the internals of hardware, operating systems and applications. In fact, this is not generally necessary and it is more important to understand the business drivers than the details of the technology. This paper describes how to minimise the risks of bad performance, and how to resolve any issues which arise. It also aims to show that performance is not an esoteric skill, but is really all about common sense.

Introduction

This article describes the art of the performance practitioner. Performance is an overloaded word—some people say "the system does not perform" when they mean that it does not meet its functional requirements or that it is often not available. This article discusses performance in terms of responsiveness and throughput. It encompasses such diverse subjects as running benchmarks, building models, tuning software and analysing user requirements. Hopefully, it will give the reader a better understanding of the broad range of performance considerations and a first idea of how to resolve the problems that can be encountered.

The article is split into the following sections:

- What is Performance?
- Performance Engineering
- Performance Modelling
- Resolving Performance Problems.

The first section gives an introduction to what is meant by performance. The next two sections describe an approach to getting performance right—through performance engineering and performance modelling.

Performance Engineering describes the systematic process by which a correctly performing solution can be produced, and shows how this process fits into the development lifecycle.

Performance Modelling describes the cornerstone of any performance engineering activity—building effective performance models.

The last section on resolving performance problems looks at what to do when something has gone wrong; i.e., how to deal with performance problems. It describes a structured approach to this which allows simple problems to be resolved quickly and complex problems to be approached in a systematic way.

The order of the topics starts with high-level abstract concepts and ends with low-level concrete advice. This is intentional—it is important to consider performance at every stage, and to carry out low-level activities with a clear understanding of the higher-level rationale. These are themes which will be repeated throughout this article.

What is Performance?

Performance Perspectives

Different categories of people have different perspectives on performance.

The **builders** of a solution are the people that create its various hardware and software components. They are concerned with meeting the performance requirements placed upon them. For this, they need the tools to assess performance, and to fix any problems.

The **users** of a solution are the people who actually make use of it. They are rarely interested in how it performs, as long as it performs well. Users want systems to be responsive and consistently so.

Business **managers** are concerned that the needs of the business are met. They want business tasks to be completed in an effective and profitable way. Managers want low-risk and value for money.

The solution is given by **service providers** who are concerned with ensuring that system performance requirements are met. They want to be able to size a system with the necessary capacity, see clearly how it is performing, and tune it to improve performance where necessary. Service providers need to be able to size systems and monitor and control performance.

Performance Attributes

Computer systems and the components that make them up have a number of performance attributes. The following table gives some examples of these performance attributes alongside the authors' views of the perspective from which these attributes are most important.

Attribute	Perspective
Fast response time	User
Hardware buffer size	Builder
Number of daily bills	Service Provider
Value for money	Manager
Sizing information	Service Provider
System monitoring facilities	Service Provider
Lines of code executed	Builder
Bills reach customers on time	Manager

Fast response time most directly affects the Users of the system, although Service Providers may have SLAs (service level agreements) which require them to provide a guaranteed response time to the Users.

Only those building the system are likely to be concerned with low level concepts like hardware buffer size or lines of code executed.

Value for money and bills reaching the customer on time are business concerns and therefore most relevant to the Manager.

The number of daily bills is likely to be of concern to Service Providers in monitoring the system and capacity planning.

Service Providers needing to provide the right configuration and run it effectively will use sizing information and system monitoring facilities. Builders will be involved in providing this information.

When deciding upon a course of action, the performance practitioner must consider both the different performance attributes and the different perspectives.

Good Performance

What is good performance? Good performance is not just about fast response times or high throughput. Other qualities must also be considered.

Absolute: How rapidly does the system work? How quickly does it respond? How long do batch jobs take to run? How many transactions a second can the system handle?

Predictable: How well is the performance of the system understood? Can customer solutions be sized? What happens if a component of the system changes or more users are added to the system?

Manageable: What choice is there about the allocation of resources? Are there tuneable parameters available to improve performance? Can the performance on live systems be monitored and understood? If a customer perceives performance problems can anything be done about it?

Scalable: How does the resource usage of the system increase with increasing load? Are there any resources (execution threads, data locks) which limit the maximum level of concurrency? Does the amount of CPU power required increase linearly or exponentially?

The different performance qualities can be illustrated with a travel analogy; see Figure 1.

For a road network **absolute** performance would correspond to the time taken to travel from A to B,

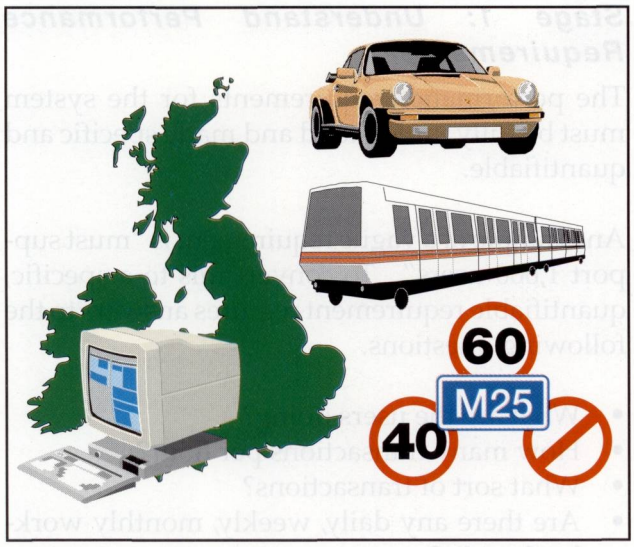


Figure 1: Travel analogy

or the number of cars which pass a certain point every day.

However, these are not the only measures of performance. It is important to be able to predict the amount of time that a journey will take. By having a *model* of the road network, such as is used in route planning software, journey time becomes much more **predictable**. For consistent performance, the train might be preferred over a motorway journey—although potentially longer, the improvement in predictability may be worth the loss in absolute performance.

Advantages can be obtained from the road network by making it more **manageable**. An example of this is the variable speed limit on the M25. At peak times the speed limit is reduced, improving the throughput of the motorway. Off-peak the speed limit is increased reducing journey times.

All transport systems can be extended to cope with increased demand, but the cost increases. If there are too many cars then new roads must be built. Similarly, new trains and railway tracks, new airports and air traffic control systems may be required. How **scalable** are these different transport systems? How do the costs of providing a road network increase with the number of cars on the road?

Performance Engineering

The Perception

A common misconception is that performance problems are things of the past—from the dark ages of computing—and that nowadays they never or rarely happen.

The main reason for this belief is that hardware is fast and cheap. However, this is no guarantee that the performance of a solution will meet its requirements. Poor designs can eat up large amounts of CPU, respond slowly on the fastest hardware or not scale beyond a few users because of some limit or bottleneck.

Another reason for the belief is that in the "unlikely event of problems" the software can always be tuned. However, the reality is that it can be very expensive to tune code; major performance improvements are likely to require major re-designs and repeated testing.

Some argue against spending effort on performance, saying that it is too expensive and time consuming to produce responsive systems. However, as systems that do not meet their requirements are worthless, producing unresponsive code is even more expensive.

Another argument is that responsive code is "tricky", difficult to understand and to maintain. This is not true, code which is designed to meet performance requirements will not be "tricky", whereas code which is "improved" later is far more likely to cause problems.

The Reality

The reality is that all too many projects suffer from performance problems. Long delays are introduced as performance improvement exercises attempt to retrieve some form of acceptable performance. Work may stop on the functionality of the product whilst the performance is improved, resulting in systems which may not have all their intended features.

Often some features perform so badly that they are withdrawn, or have other usage restrictions imposed, such as only ten users logging in at once, or reports being generated at night.

The result of all this is likely to be that the budget is exceeded, sometimes to the extent that the project is cancelled. Surveys estimate that only about 16% of projects fully succeed, a further 53% are over budget, late or have reduced functionality and 31% get cancelled. Problems with performance are often the reason.

The Answer

The solution to all these problems is performance engineering: a systematic step-wise approach to ensure that a solution meets its performance requirements.

Performance must be considered at every stage of the project from feasibility and requirements analysis all the way through to operation and evolution. This does not mean that great effort should be expended at every stage, but that sufficient work should be done to manage performance risks in a cost-effective manner.

Central to effective performance engineering is building performance models. There is a common misconception that models are complex, difficult to produce, and wildly inaccurate. In fact, a model should be the simplest possible encapsulation of the knowledge held about the performance of the system. The model should only be complex if that is genuinely required in order to manage the performance risks. Complex models are rarely necessary. Modelling is described more fully in the next section of this article.

This section describes a 6-stage approach to performance engineering:

1. Understand performance requirements
2. Set performance targets
3. Implement against targets
4. Measure against targets
5. Validate against requirements
6. Manage performance of live system.

The six-stage process proceeds through the product life-cycle from requirements to the running of the live system. At every stage the process is backed up by performance modelling and at every stage the performance qualities of absolute, manageable, scalable and predictable performance must be considered. The six stages are described in detail below.

Stage 1: Understand Performance Requirements

The performance requirements for the system must be fully understood and made specific and quantifiable.

An example of a vague requirement is "must support 1,000 users". To convert this to a specific, quantifiable requirement requires answers to the following questions.

- What are the users doing?
- How many transactions per hour?
- What sort of transactions?
- Are there any daily, weekly, monthly workload peaks?
- What response times are acceptable for the different transactions?

If the requirement is not properly understood and agreed between the customer and the project then developing the system is an open-ended commitment. Furthermore, there is no way of knowing what constitutes a system with acceptable performance, so there is no way one can be developed, except by chance.

Stage 2: Set Performance Targets

The overall performance requirements must be devolved into component targets. These targets must be realistic and achievable.

This is necessary as overall requirements may be clear, but to individual developers or teams it may not be at all clear what they, personally, have to do in order for the system as a whole to meet its requirements.

Similarly, if part of the solution is being bought in or sub-contracted then the performance requirements for that part of the solution must be identified. If requirements are not placed on the sub-contractor then the project is taking on risks over which it has no control.

This devolution of targets can be an iterative process. Component targets can be broken down into sub-targets for sub-components.

When setting targets it is important that they be realistic, achievable and have the buy-in of the developers. Targets can be produced in a variety of ways, for example, comparison with similar

products or components, prototyping, by breakdown into "known" chunks, using information from suppliers or benchmark results (e.g. TPC-C, AIM, SPEC), by guessing or by considering best and worse cases. Performance modelling is important here to demonstrate that the component targets combine to produce a solution that meets its performance requirements.

Stage 3: Implement Against Targets

The next step is obviously to try to construct components that meet their targets.

Having targets clearly defined helps the developer to produce code with the right performance characteristics. The developer knows components must be slick or must scale well, and which are not performance critical.

The developers can also flag performance problems at an early stage if they can see it is impossible to meet the target. The earlier a potential problem is found the more that can be done about it. The options are trade-offs between complex functionality and performance, improvement of other components, re-design or a renegotiation of the requirements with the customer.

Again, performance modelling should be used to show what the effect of the component missing its targets has on the overall system performance. This time, targets are replaced by the new performance estimates.

Stage 4: Measure Against Targets

Once the components have been developed they must be measured to see whether they have met the targets. Any shortfalls must be investigated. This is analogous to functional unit testing.

The performance model can now be populated with actual performance measures and the overall system performance predicted. Any possible system performance problems can then be investigated.

The sorts of things measured include response time, resource usage, throughput and queue lengths. There are a number of different ways of collecting the information including counters, profiles and traces. Care should be taken to use the appropriate techniques for each case.

Measurement can be compared with traffic monitoring; see Figure 2 below. The measurements with the smallest overhead are counters; for example, a strip on the road that records cars going over it, or a census taker counting traffic. These do not interrupt the flow significantly.

To get an idea of who uses a road and why, every Nth car is stopped, and the driver asked a number of questions. This is a form of profiling, which provides more information, without having too much impact on the throughput of the road.

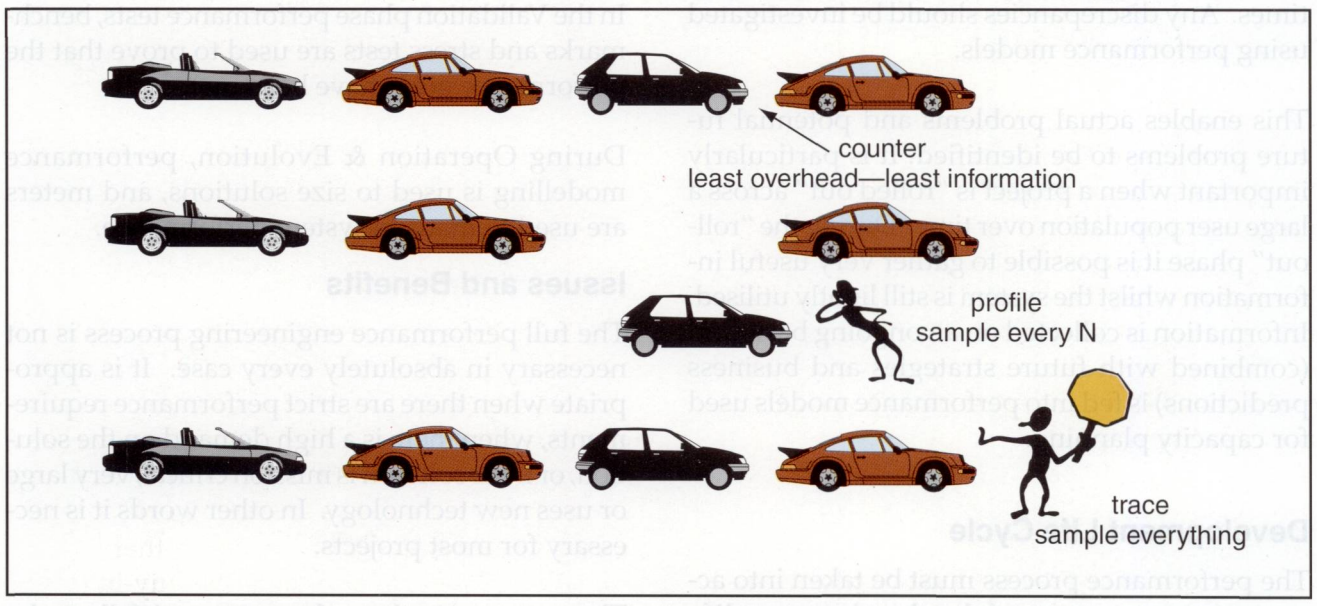


Figure 2: Examples of measurement

Where a full trace is required—for example when there is a criminal suspected of being on the road—a roadblock may be set up, and every single car stopped. This is similar to tracing, and it causes a very heavy performance overhead.

Stage 5: Validate Against Requirements

When the product is complete the performance must be measured against the requirements and against the model predictions. This is analogous to functional validation.

Any performance shortfalls must be investigated; performance models will be very useful in helping to identify which parts of the system are not behaving as expected and in identifying possible improvements and the impact that each will have.

Stage 6: Manage Performance of Live System

Once the system goes live it is important to manage its performance to ensure that it performs well. It may turn out that the system is not used as was expected and this may result in performance problems. It is important to understand the cause of any problems and if possible to notice the potential for future problems so that action can be taken before the users notice.

The performance of the system should be monitored to check that performance requirements are being met and that behaviour is as expected. This includes the workload, resource usage (CPU, IOs, network, memory), response times and elapsed times. Any discrepancies should be investigated using performance models.

This enables actual problems and potential future problems to be identified. It is particularly important when a project is "rolled out" across a large user population over time. During the "roll-out" phase it is possible to gather very useful information whilst the system is still lightly utilised. Information is collected on an ongoing basis and (combined with future strategies and business predictions) is fed into performance models used for capacity planning.

Development Life Cycle

The performance process must be taken into account at every stage of the development life cycle. A typical 'V'—diagram representing the development life cycle is shown in Figure 3.

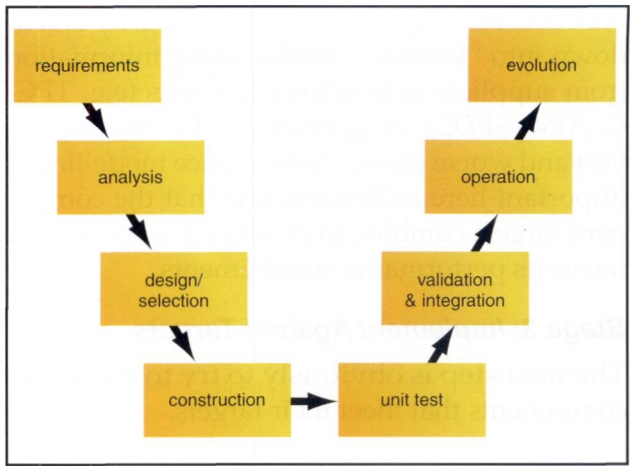


Figure 3: Development Life-cycle

During the Requirements phase the performance goals are identified. In the Analysis phase the performance of alternative architectures is estimated to see whether they meet these goals.

The Design phase causes the solution to be decomposed into sub-components—the performance goals must similarly be decomposed into performance targets for each of the sub-components. Also, the meters necessary to understand and manage the performance are identified.

During Construction sub-components are implemented or bought-in to meet their performance targets and in the Unit Test phase the performance of the components is measured against those targets.

In the Validation phase performance tests, benchmarks and stress tests are used to prove that the performance goals have been met.

During Operation & Evolution, performance modelling is used to size solutions, and meters are used to manage system performance.

Issues and Benefits

The full performance engineering process is not necessary in absolutely every case. It is appropriate when there are strict performance requirements, when there is a high demand on the solution, or if the solution is mission critical, very large or uses new technology. In other words it is necessary for most projects.

There are a number of common pitfalls to be avoided when using the performance engineering process. Firstly, if the performance requirements

are insufficiently clear then this may lead to arguments between the customer and the supplier about whether the performance really meets its requirements.

Secondly, performance models are often produced at the start of the project and then never looked at again. In this case the accuracy of the model is unlikely to be very high, and the chance of the performance requirements being met is very low. Responsibility should be allocated to someone to maintain performance models throughout the project.

Another pitfall is when requirements are generated and targets set, the developers simply ignore the targets. This may be because the developers do not agree with, or are not aware of the targets. It means that performance problems will not be identified until system validation, at which stage it is far more expensive and time-consuming to do anything about them.

The final common pitfall is that the solution is not scalable. Single user performance is fantastic and simple performance models show that many users can easily be supported. However, because of some bottleneck which has not been modelled, the system can only support a few users. This can be a very difficult problem to solve late in the project—the scalability of the solution needs to be carefully analysed during the analysis and design phases of the development life-cycle. **Load testing beyond the maximum anticipated load should be factored into the project plan to give confidence in the scalability of the solution.**

There are many benefits to performance engineering. It is a controlled way of system development which allows the project to meet its budget, timescales and performance requirements. Without performance engineering the costs incurred in getting the performance right tend to be wildly variable, and to be incurred at the back end of the project. Often a compromise solution is produced which does not quite meet its requirements and which is unresponsive. As a general rule fire prevention is twenty times cheaper than fire-fighting.

A further benefit is that the system is more likely to work, as performance engineering practitioners tend to have a cross-functional and cross-organisational rule which allows them to spot functional problems more easily than someone working on a single component of the solution. Often the performance people are the only ones who gain this helicopter view.

Some of the issues raised are illustrated in Figure 4.

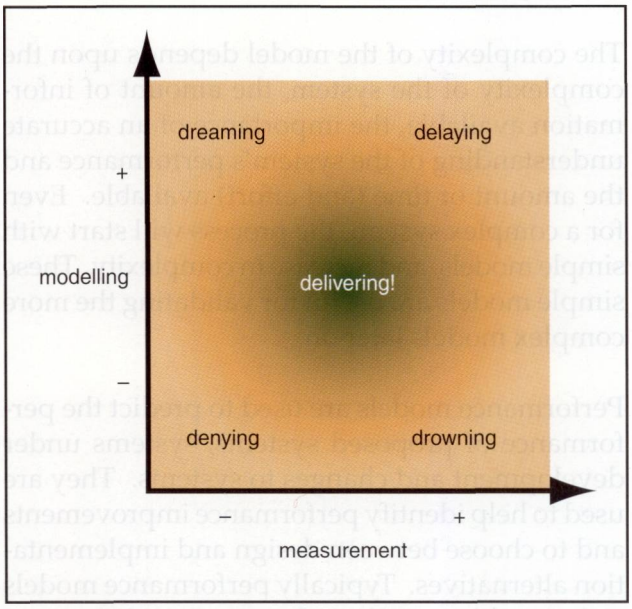


Figure 4: Stuben Performance Window

Too little modelling and measurement—you are **denying** the issue. Low measurement, too much modelling—you are just **dreaming**. Low modelling, lots of measurement—you are **drowning** in data. Too much of everything and you are **delaying** the project. Get the balance right and you will be **delivering**.

Summary

Performance problems are more common than you think. But do not despair, because performance engineering is the answer. Remember the performance qualities, *absolute*, *manageable*, *predictable* and *scalable*, and follow the six-stage process.

Performance Modelling

What is a performance model and why are they used?

Model: A simplified description of a system etc. to assist calculations and predictions.

A performance model is any simple description or representation of a system that is used to

increase understanding about the performance of the system. It need only describe those aspects of the system that influence its performance (for example, rare transaction types might be ignored). It can be anything from a few calculations on the back of an envelope to a spreadsheet to a complex computer program that simulates the behaviour of the system.

The complexity of the model depends upon the complexity of the system, the amount of information available, the importance of an accurate understanding of the system's performance and the amount of time (and effort) available. Even for a complex system, the process will start with simple models, and increase in complexity. These simple models are useful for validating the more complex models later on.

Performance models are used to predict the performance of proposed systems, systems under development and changes to systems. They are used to help identify performance improvements and to choose between design and implementation alternatives. Typically performance models estimate things such as the response time a system will deliver or the maximum workload it can support.

Without building models, the only options are to guess or to build the system and measure its performance, with the inevitable expense and delay that this will create.

Performance Indicators and Queuing Theory

The key indicators of performance are:

Throughput: Mean jobs completed per unit time

Service Time: Mean time taken to handle a job (excluding queuing)

Waiting Time: Mean time taken for a job to complete including queuing (this is also known as Response Time, Elapsed Time or Existence Time)

Utilisation: Percentage of time a resource is used; i.e., ratio of throughput to capacity

Population: Mean number of jobs in the system (this is also known as Concurrency).

Maximum throughput is also known as capacity.

To illustrate what these indicators mean consider a bank with a number of counters, each with a clerk. The meanings of the performance indicators are:

Throughput: Customers processed per hour

Service time: Mean time a clerk takes to handle a customer

Waiting time: Mean time spent by a customer in the bank

Utilisation: Mean proportion of time when clerks are serving customers

Population: Mean number of customers in the bank.

These indicators can then be used to describe the performance of the bank. Given that there are 4 counters (each with a clerk), that 180 customers are handled in a typical hour and that a clerk takes a mean time of 1 minute to handle a customer then:

Throughput = 180 customers/hour
Service time = 1 minute per customer.

The Utilisation can also be worked out. If each clerk takes a mean time of 1 minute to handle a customer, then on average they can each handle 60 customers in an hour. They are handling 180 customers per hour, which is 45 customers per hour for each clerk. The clerks are, therefore, each handling 45 customers per hour when they could handle 60 customers. This means they are utilised 75% (45/60) of the time.

Insufficient information has been provided to work out either the Population or the Waiting Time.

For example, if all 180 customers entered the bank at the start of each hour then it would take 0.75 hours to handle them all (180 / (60 x 4)). During these 45 minutes there would be an average number of customers in the shop of about 90 (180 at the start, 1 at the end). For the other 15 minutes of the hour there will be no customers in the shop. This gives an average Population of about 67 (=(45 x 90 + 15 x 0)/60).

Alternatively, if 3 customers came into the bank every minute (making 180 in total in an hour) and each was handled in exactly 1 minute then there would always be exactly 3 customers in the bank. This gives an average Population of 3.

In order to work out the Population and Waiting Time more information is required such as arrival distribution (e.g. Markovian), service distribution (e.g. constant, Normal), queuing discipline (e.g. first come first served) and number of queues (e.g. one queue for each counter, or one queue for all the counters).

However, if the mean number of customers in the bank (Population) is known, then the Waiting Time can be calculated using Little's Law, which says that:

Waiting time = Population/Throughput.

Little's Law can be intuitively explained by saying that the Population is the number of jobs that arrive whilst a job is waiting in the queue. A more rigorous explanation is as follows. Allocate each job a point for each second it spends in the system. In time T seconds the mean points scored by each job is Waiting Time. The number of jobs is T x Throughput. The total points is T x Throughput x Waiting Time. But the total points is also the Population x T. So Throughput x Waiting Time = Population.

There are far more complex formulae for analysing systems—this is a whole branch of mathematics. Models which look at a system using mathematics in this way, are called "analytical" models. However, mathematics can only go so far. For a precise analysis of a complex system a computer simulation may be necessary.

To show the relevance of all this to IT the indicators can be mapped onto some computer system concepts. For example:

Waiting time: TP response time, batch elapsed time
Utilisation: total CPU used, network bandwidth consumed
Throughput: transactions per second, IO/s
Population: buffer sizes, number of users.

Approach to Modelling

There are many different types of performance model for solving different types of problem. For example, a simulation of a processor pipeline may be produced in order to predict the performance of the CPU before it has been built. This section describes the production of a performance model for a complete customer solution (rather than for a component). This is the sort of model referred to in the section on Performance Engineering.

The steps in building a model are described below. It is important to note that it may not be necessary to go through all the steps depending on the level of accuracy required. A complex queuing model is only necessary either where great accuracy is required or where there is great contention for resources in the system and therefore much queuing. See Figure 5.

The whole system need not be described in the same model. For example, if only one part of the system has substantial queuing then a separate complex queuing model can be built just for that.

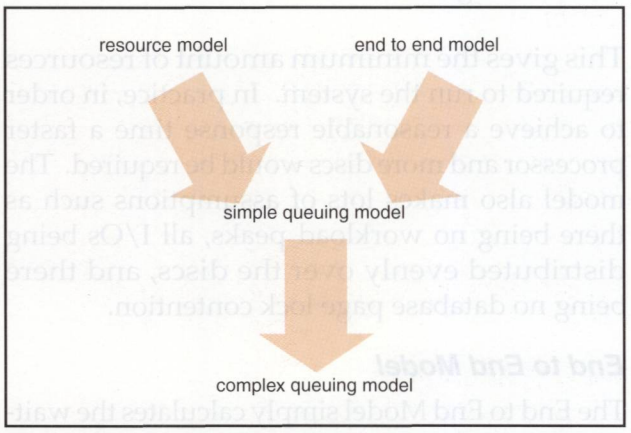


Figure 5: Queuing models

Resource Model

A resource model calculates the utilisation of all key resources in the system. It does this be using the formula:

utilisation = throughput/maximum throughput.

A resource model has the following elements:

Workload: volume of services e.g., how many TP enquiries per hour?

Design: what operations are required to do each service?
what sub-operations are required to do each operation?

Implementation: what resources are used by each operation?

Technology: the capability of each component of the system to deliver resources; e.g., MIPS, IO/sec, network bandwidth.

A resource model will usually be a spreadsheet which contains numbers for all the above elements. By multiplying up the numbers the total utilisation of all the system resources can be identified. Alternatively, the technology required (e.g., speed of CPU, number of disks), so that the utilisation of all resources is not too high, can be calculated.

An example of a resource model is shown in Figure 6 for a simple library computing system which allows books to be registered online as either loaned or returned.

The usage of resources can then be calculated as shown in Figure 7.

This gives the minimum amount of resources required to run the system. In practice, in order to achieve a reasonable response time a faster processor and more discs would be required. The model also makes lots of assumptions such as there being no workload peaks, all I/Os being distributed evenly over the discs, and there being no database page lock contention.

End to End Model

The End to End Model simply calculates the waiting times for all the key processes by ignoring queuing. It does this by adding together all the service times.

In the library example it would be used to give a simple first estimate of the response time to the user of the system.

The End to End model works out what the waiting time would be if there were only a single job in the system at any time; e.g., a single user system. This waiting time can often be verified by single user testing.

The model produces a minimum possible value for the waiting time. If that minimum value is higher than the requirement then a performance problem has already been identified.

Simple Queuing Model

The simple queuing model recalculates the waiting times for all the key processes by adding some simple queuing for all the important resources. It does this by using the formula W=S/(1-U) to find the waiting time for all the resources. These can then be added together to give the total waiting time. The utilisation U of each resource is taken from the Resource Model.

In general the formula W=S/(1-U) is just an approximation, although it is exact under certain mathematical constraints. The waiting times calculated by the simple queuing model are still not precise, but give a much better estimate than the End to End Model.

Complex Queuing Model

On occasions when an accurate answer is required for the key process waiting times, a proper queuing model must be produced. This can sometimes be done mathematically or a simulation model can be produced. The complex model should be validated using the simple queuing model and the resource model.

Some Modelling Tips

- Start broad & shallow; go deeper only where necessary.
- If the worst case is all right use it. If the best case is not all right, then there is a problem.
- Use as little queuing theory as possible, it is difficult and error-prone.
- Only simulate if necessary; it is very complex and time consuming.
- Document the model and its assumptions and get them reviewed.
- Remove model bugs by looking at trends and extreme cases and by comparison with expectations and simpler models.

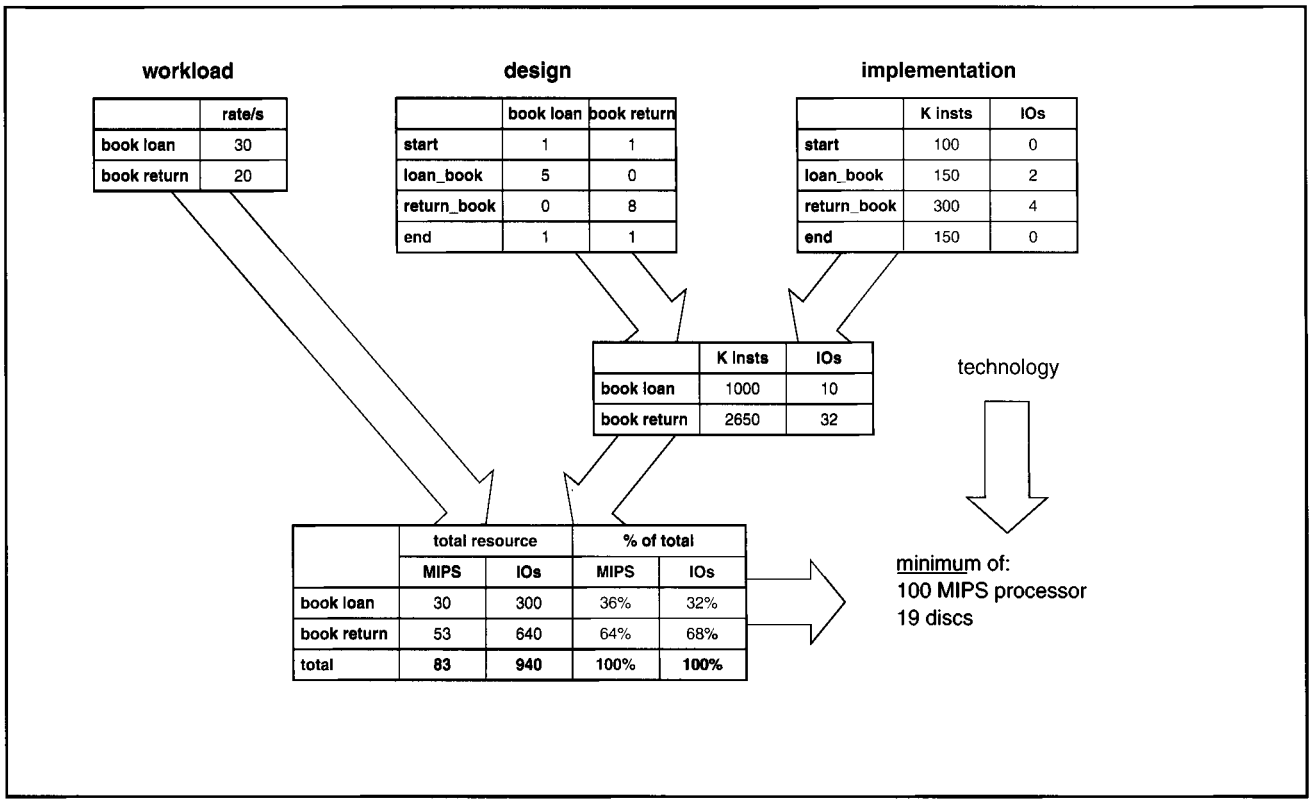
❑ **Workload**
- 30 book loans per second (5 books per loan on average)
- 20 book returns per second (8 books per loan on average)

❑ **Design**
- book loan
  — start
  — loan_book x no. of books
  — end
- book return
  — start
  — return_book x no. of books
  — end

❑ **Implementation**
- start: 100K insts, 0 IOs
- loan_book: 150K insts, 2 IOs
- return_book: 300K insts, 4 IOs
- end: 150K insts, 0 IOs

❑ **Technology**
- servers available at 50 MIPS, 100 MIPS & 200 MIPS
- disks support 50 IOs

Figure 6: Example of a resource model



	rate/s
book loan	30
book return	20

	book loan	book return
start	1	1
loan_book	5	0
return_book	0	8
end	1	1

	K insts	IOs
start	100	0
loan_book	150	2
return_book	300	4
end	150	0

	K insts	IOs
book loan	1000	10
book return	2650	32

	total resource		% of total	
	MIPS	IOs	MIPS	IOs
book loan	30	300	36%	32%
book return	53	640	64%	68%
total	**83**	**940**	**100%**	**100%**

Figure 7: Resource model calculation

- Verify the model inputs by unit testing and validate the model outputs by load testing and measurements from live systems.

- Do not overestimate the accuracy of the model. The accuracy of the model depends on the accuracy of its inputs and measurements are much better than estimates. It is also depends on the detail of the model and the amount of resource contention—queuing for resource is difficult to model and tends to make the model outputs more inaccurate.

Resolving Performance Problems

The previous two sections describe how to get performance right. This section looks at what to do when performance goes wrong, in other words, how to deal with performance problems.

What Is a Performance Problem?

A performance problem is any situation where a system does not conform to its performance requirements. This applies not only for absolute performance (i.e. poor response times or the inability to scale beyond a certain number

of users), but also for unpredictable or unmanageable performance.

A Structured Approach

When dealing with performance problems it can be very tempting to leap straight in to taking measurements and running tests—often, there is a great pressure to get results quickly (a customer may be on "red alert" for example). As with the development of a solution, however, performance problems are solved more reliably and, overall more quickly, by following a structured approach.

Such an approach, in three steps, is now described.

1. **Define the problem:**
   It is surprising how often the root problem is not a problem with the computer system. Identifying the real problems (i.e., if these issues were resolved, would you be happy?) and getting all interested parties to agree to them at the beginning, is a very good investment of time.

2. **Fix the immediate problem:**
   The main focus of this section is identifying a fix for the immediate problem.

3. **Solve the long term problem:**
   As important as the immediate fix is the long term solution, often omitted until the same problem recurs. Solutions to performance problems are typically built on improved processes rather than technology changes, such as capacity planning to anticipate resource shortfalls before they occur.

Step 1: Define The Problem

This is a vital step before attempting to fix what you perceive to be the problem. Unless you have an agreement about what the problem is, you are making an open-ended commitment of your time—not to be recommended. These problems may not be problems with the computer system at all, for example:

> One customer was unhappy because the throughput of their UNIX system was higher than their more expensive mainframe system for the same test programs. It was pointed out that the mainframe had many other benefits— security, reliability, etc.—which justified its higher cost. There was no performance problem with the mainframe as such (the response times were fast enough). To solve this problem did not need any IT skills at all.
>
> Another customer was unhappy with the performance of their new system. It had been expected that they wanted their new system to be faster, but in fact they had bought the new system to attach more disks—so they would be happy if performance remained the same while the file system was much larger. A simple configuration error was found to be the problem—performance in fact improved somewhat when their previous configuration was reinstated on the new system—and the customer was happy. Much time could have been wasted trying to improve performance relative to their old system if their requirements had not been agreed at the beginning.
>
> Many projects which are close to delivery still have unclear requirements. In one project, for example, very different response time targets had been agreed for "client" operations and "processing" operations—once the system was ready to be deployed, the debate opened about which operations were "client" and which were "processing". In these situations, it is vital that the requirements are agreed before you attempt to tune the system to meet them.

Many "performance problems" can be solved at this stage, typically with a non-technical solution such as changing requirements or scheduling work at other times. For most projects, finding out the real requirements is an eye-opening process.

Step 2: Fix The Immediate Problem

Listed below are common approaches to fixing performance problems.

- Repeatedly doing "good" performance things until the problem is fixed:
  This approach involves looking for good advice which seems to relate to the current system and applying it; e.g., enlarging a database cache, applying a service pack, configuring more threads or moving files around to reduce contention.

- Changing the application until the performance is acceptable:
  This approach involves modifying the application wherever performance can (allegedly) be improved, often including cutting back functionality or placing arbitrary restrictions and, as a consequence, destabilising the application.

- Generating a large amount of "performance data" to find out where the problem might be:
  This approach involves turning on every form of available metering and generating many gigabytes of output.

These approaches are all extremely inefficient, although one may eventually stumble across a solution to a problem. The key to dealing with problems effectively is to understand them, allowing both the short term fix and a long term solution to be put in place. As understanding of the problem grows, whether a "good" performance change is appropriate can be evaluated, application changes can be avoided unless vital to delivering against performance requirements. Performance data can then be limited to that required to confirm or enhance understanding.

In order to understand the performance of a system, and thus be able to address its problems, a scientific approach should be adopted; i.e. developing an hypothesis about how a system behaves, testing this hypothesis through experiment and revising the hypothesis based on available evidence.

The following process is proposed to master performance.

- **Measure**: obtain data about current system behaviour.

- **Analyse**: extract evidence from this data.

- **Structure**: use performance models to build understanding—a theory.

- **Tune**: use this understanding to tune the system.

- **Examine**: does the system meet its performance requirements? Can it ever do so?

- **Repeat**: until the system meets requirements, or this is known to be impractical.

The following sections look at each of these steps in more detail. Note that, in practice, these steps inevitably blur into one another—they are not a sequence to follow slavishly in all cases.

Measure

Measurement should be about gathering data through experiment. The approach should be to undertake test runs to confirm a theory, or to provide base data in order to form a theory. In short you should have an expectation before you take measurements, based on your current level of understanding. It can be all too easy to fall into the trap of running test after test to little effect, generating huge amounts of data with little idea of how to go about analysing it.

Whatever measurements are taken must use **repeatable** testing in order to understand the effects of changes to the system in a controlled environment. The most practical way to ensure this is to **automate** testing as much as is possible, using load generators to simulate user populations. In order to understand measurements, it is often helpful to start with simple tests, using, for example, only a single user and then build up to more complex ones.

Results should be managed in a structured way (unique run identifiers, organised storage and archiving of results etc.). This will allow you to go back and answer questions about past runs when you gain new insights later on. A short summary of the run (a text file with the purpose of the run, number of users, any software changes for example) can be invaluable. All meters that can be switched on without a significant impact on system performance (say, over 5% of processing), should initially be collected—most counters fall into this category. For most systems, a frequency of one minute for sampled counters will be appropriate. A wide range of counters should be collected to cover the main areas where performance problems can occur.

Think about the information that will be required for confirmation or extension of your current level of understanding. Application statistics in particular are vital, for example to analyse in terms of costs per transaction rather than overall resource usage on the system. Collecting CPU us-

age on a per-thread or per-process basis can also allow problems to be narrowed down very quickly.

As taking measurements can be dull and error-prone, again, **automate** as much as possible. Have a standard set of metrics that are collected which you can modify over time. Start by collecting what you can do easily, and put more work in if there are problems which require further information. Make sure that the appropriate meters are actually working (e.g., turn disk statistics on in Windows). Check that measurements make sense for a simple test that you understand, before relying on them for analysing a more complex situation.

Analyse

Once data has been collected, it needs to be analysed in order to provide evidence to support or alter your current understanding of how the system behaves. This is a much easier task if you have ensured that you have expectations of measurements beforehand, and undertake them for a specific purpose.

Simple inspection of data can provide useful insights. Using a spreadsheet to display data visually can be a great help here. Examples include spotting "glitches" in a graph, and being able to collate the time that these happened with other data—for example a reduction in throughput happening at the same time as a database checkpoint. Using traces, flaws in program logic can often be seen by inspection, for example, calling the same function twice (this happens!). Another common problem is when tracing or other debug information is left on by mistake—inspection of logs etc. can reveal this.

Inspection is all about comparison with your expectations—expecting a smooth graph, a particular flow of logic, a certain pattern of resource usage etc.. In general, comparison is the other method of analysis, including comparison with:

- **itself**; for example, looking across several samples within a run
- output from a **model**
- **historical statistics** from the same system before a problem arose
- statistics from **other systems**
- **published information** from white papers or benchmarks
- **theory**; for example, does Little's Law hold? (If not, your data is wrong).

All these can be useful sources in analysing the data and coming up with appropriate information and insights into system behaviour. For deeper insight into behaviour, it is important to structure data and look for "gaps"—for example, elapsed time not accounted for in the sum of time allocated to individual resources.

Structure

The evidence obtained from analysis of system measurements needs to be structured using performance models. These need not be at all complex—it might simply mean summarising results in a spreadsheet.

Structuring the available evidence leads to an **understanding** of system behaviour, rather than a collection of unrelated information. It closes the loop, either demonstrating that the information available "makes sense", or highlighting areas for further investigation by giving expectations for further runs and suggesting appropriate experiments. **Building up understanding about how the system works is the most important part of performance engineering**—and one frequently omitted.

When trying to produce this kind of structured description of system behaviour, shortfalls in the statistics being collected are often discovered—this requires an assumption to be made about the missing area within the model, which then needs to be checked through further experiment with additional metering. Modelling also provides expectations to use when analysing data.

An example of this kind of structuring is to build an end-to-end model of response time. This might consist of some time waiting for the network, some processing time, some disk time, and some network time coming back. If this simplistic analysis comes out much lower than the observed response time, queuing time waiting for the processor or disks might be the answer—further runs might then be undertaken to provide the information needed to calculate this, or to suggest alternative reasons for this "lost time". Once all the times add up (within

reasonable error bounds), tuning can focus on the larger components of the total cost.

In order to do this structuring properly, it will be necessary to understand the business context in which the system is operating; i.e., what the system is for. This often means that performance experts spend a lot of time talking to people about the system at a non-technical level before going into the technical details. This becomes particularly important in the tuning phase as the changes proposed to tune the system are almost always a compromise, which must balance technical and non-technical issues; for example, performance versus cost, throughput versus response times, response to on-line users versus batch performance, security versus response, etc..

Tune

Tuning should be based on an understanding of the system, gained from analysing data in a structured way. All too often, people tune by doing "good" performance things until the performance is perceived to be "good enough" (or until they run out of time, money, or their customer runs out of patience). Although it can seem frustrating to be spending time gaining an understanding of the system, and in particular its *non-functional requirements* (security, reliability, user response times, expected scalability etc.) at a time when people are demanding improved performance immediately, this is vital to putting appropriate tuning in place. It might seem a good approach, for example, to tune the operations with the longest response times, while in reality the customer's priority may be to deliver good response to a certain group of users (e.g., telesales) rather than a particular set of transactions. Identifying such requirements is therefore paramount.

When making changes to the system, try to tune one thing at a time wherever possible, in order to be clear about the effects of each individual change. In practice, although it is often difficult to do this (e.g., requiring too many runs) thinking carefully about which changes are made in what order can speed up the rate of improvements significantly. In any case, use performance models to develop a theory about what effect the tuning will have—in other words to form an expectation of what results you will obtain.

Examine

When analysing statistics and making detailed changes to a system, it is easy to lose sight of the wider picture. This makes it important to step back at intervals and consider:

- have the requirements been met—should we stop? Tuning systems beyond their requirements is unnecessary, but not uncommon
- have the requirements changed? Often the initial requirements (or your understanding of them) will change over the course of the exercise
- are we still able to meet the project plan? Sometimes your understanding of the system will reveal that certain requirements can never be practically met; e.g., the system cannot scale beyond 64 concurrent users—these need to be communicated as soon as they are discovered.

Check with your customer to see if they are satisfied with what you are doing.

Repeat

Go back around the loop until your requirements are met and your customer is satisfied.

Step 3: Solve The Long Term Problem

The exercise described above provides a short term fix for the system, based on an understanding of its behaviour. This understanding should then be employed to determine the long term solution to ensure that the same problem does not occur again.

There are a number of mechanisms for doing this.

- **New processes:**
  Change processes, and automate them where appropriate, to avoid problems. This can be something simple such as running reports as a service overnight rather than ad-hoc during the day, or closing and restarting services automatically every night to work around memory leaks.

- **Monitoring and regular analysis:**
  Put metering in place to track important parameters, to avoid being surprised by the same problem again; e.g., tracking disk free space, memory utilisation etc. These must be regularly analysed if they are to be effective.

- **Alerting:**
  Use systems management tools to send alerts (via management workstations, email, pager etc.) when resources being monitored cross critical boundaries.

- **Capacity plan:**
  As resource requirements grow over time, capacity planning methods should be used (trend analysis for example), to anticipate the need for further resource, and put it in place before problems occur.

Hints And Tips

This section provides some practical advice on how to approach performance problems of different types.

Low Throughput

The first step is to understand what the requirements are—why is the throughput considered to be low? What are you aiming to achieve? What is the business impact if this proves impractical? This gives you a context within which to work.

In order to understand throughput, ask "where is time being spent in the system?" The possible answers to this can be summed up with "MANIC" below. Build a simple resource cost model using these elements.

- **Memory**:
  Memory is a very common problem area on Windows systems. Collect statistics on total occupancy, free memory and paging.

- **Application:**
  This is the area most often overlooked, but it is vital for analysis. It includes user applications, databases, transaction monitors, queuing systems etc.. Collect statistics relevant to the application, e.g., user interactions, lock activity. Often the statistics that you would like are not there—this is why it is so important to build application meters in at design time.

- **Network:**
  It can be difficult to capture the right network information using standard software. Collect basic network statistics (bytes and packets sent and received). Consider recording key times in the application, or look for other built-in statistics, for example, time stamp requests sent to a server, and when the response is received.

- **I/O:**
  Typically disk traffic, but includes tapes and other devices. Collect transfers per second and service time for each device.

- **CPU:**
  This is the most commonly monitored resource. However, lack of CPU resource is often not the root problem. Collect overall utilisations, time under interrupt, process/thread based statistics. Consider using profiling to break down CPU time further.

The resource cost model looks at the utilisation of each of the components of the system. If the utilisation of any component is very high, this is probably the performance bottleneck. Options in each of the areas are discussed below.

- **Memory**:
  If the system is paging, more memory is required by the application. This can be achieved by (i) reducing the memory used by other applications, (ii) changing application behaviour, or (iii) adding more memory to the system. If there is a memory leak, option (iii) simply defers the problem rather than fixing it. Other options include restarting services automatically each night to clear memory leaks, reducing the memory configured for file system or database caches and limiting the number of concurrent users (or processes, threads etc.) on the system.

- **Application:**
  If the application is waiting for a resource, such as a database lock, options include configuring more or reorganising existing resource (e.g., spreading a highly contended database table across more database blocks) and changing application behaviour (e.g., report unavailable or provide approximate information rather than wait).

- **Network**:
  It is important to separate problems waiting for what is on the other end of a network (e.g., another server) and problems with the network itself. A network analyser can be invaluable in assisting with this. If you do

have a bottleneck, options include separation of sections of the network to reduce other traffic, changing behaviour (e.g., to use unicast rather than broadcast messages) and configuring more capacity.

- **I/O:**
  Disk bottlenecks may be the result of (i) uneven distribution of load across disk devices (ii) contention between multiple demands on the same device (iii) inappropriate application behaviour or (iv) not enough resource. Options for (i) include striping files across multiple disk spindles (e.g., RAID-0) and changing the placement of database tables or other files. Options for (ii) include changing file placement and moving applications on to different sets of spindles altogether. For (iii), options include caching in the disk controller or more disk spindles. As disks become larger this problem becomes more prevalent, with fewer disks being configured to meet storage requirements.

- **CPU:**
  Processor cycles may be being consumed for activities that are not required. Context switching between threads or processes, for example, can consume a lot of resource. This can often be reduced, for example, by reducing the number of concurrent threads in the system, and thereby lowering CPU utilisation.

If there is no obvious bottleneck, consider the following questions.

- As you increase the load on the system, how does its behaviour change? Look for increased queuing (e.g., longer disk service times) indicating a bottleneck. Alternatively, look for errors as connections are refused.

- How much concurrency can be supported, and what is limiting this? Look for thread pools of a fixed size, locks on database objects, etc..

- Check out resource costs per transaction as load varies; does it increase linearly? If not, investigate the components which do not scale.

- Look at all the components in the end-to-end system ; is the bottleneck within the scope that you are considering? An application server, for example, may be waiting for the database, without having any performance problems itself.

Long Response Times

The first step is to understand what the requirements are and why are response times considered to be low? Are they poor across the board, or only for certain transactions, users, times of day, etc.. What are you aiming to achieve? What is the business impact if this proves impractical?

In order to understand response times, build a simple end-to-end model for each of the operations that the customer is most concerned about. The end-to-end model looks at the time spent at each stage. Start with network delays (do not forget it goes both ways), CPU time, and I/O time; does this add up to the response times seen? If so, focus on the larger and more easily tuned components. For example, a 20 second response time could be broken down as shown below in Figure 8.

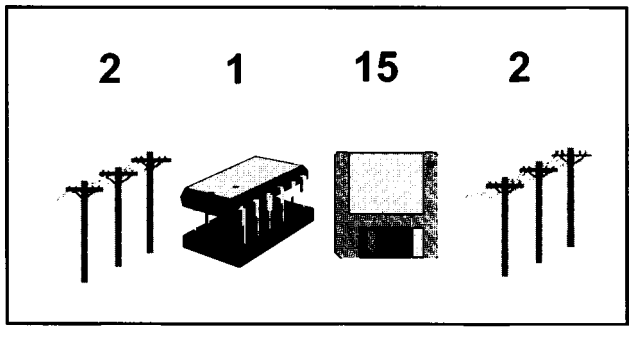


Figure 8: Example of response time breakdown

If individual service times don't add up to the response times seen, add in time queuing for CPU and queuing for disk. If that doesn't account for the time, there may be delays in the application (e.g., polling a lock or a physical device until it is ready), or the statistics may not mean what you think. Try to get the statistics to "make sense" (e.g., do the individual CPU costs by process add up to the total CPU cost? If not, perhaps a process is created and destroyed between monitoring samples, using resource in a way that is not recorded).

If there is no obvious problem area, consider the following questions.

- What are the response times with a single user? If these are also poor, the problem is probably with the application, otherwise the problem is probably scalability or system configuration (e.g., size of database cache).

- Is response time bad for all transactions; if not, what makes these different? For example, updating transactions may be slow, while read-only ones are fast.

- If you run the operation locally (e.g., on the server) is it much faster? This could indicate a network or protocol problem.

Variable Performance

The first step is to understand what the requirements are. Is the worst response time considered to be unacceptable? Do all operations vary in performance, or only for certain transactions, users, times of day, etc.. What are you aiming to achieve? What is the business impact if this proves impractical? Variable performance is particularly unpopular with users, even if the worst performance seen would have been satisfactory if it was consistent.

Dealing with variable performance is all about correlating data. What is it about the times when the system performs well that is common? What is the difference between these times and those when the system performs poorly?

If the differences are unclear, consider the following questions.

- Can you reproduce the variability in a test run? This can help, because you can change a test environment more readily.

- Take statistics and structure them in a model—which components are variable? If you cannot "catch" the system misbehaving, how could components vary?

- Do the operations with poor performance share common parameters? For example, all accessing a particular database table or all customers from one region.

- Do you need to reorganise your data? For example, rebuilding tables, removing disk fragmentation.

- Consider some possibilities. Are other applications running on the server interfering? Is a periodic function running (e.g., database checkpoint)? Are other users of this application interfering (e.g., reports versus on-line)? Is a buffer or cache filling up and being cleared? Are some areas of a database over full (e.g. chained blocks or hash buckets)?

Some Common Problems

This section describes a few common performance problems to look out for and indicates what can be done about them.

Memory leak

If the free memory on the system is steadily reducing, there may be a memory leak (other possibilities include memory not being freed by, for example, user connections which log out). If there is a leak in a service, this can be restarted—consider using a scheduler to cause this to happen automatically each night.

Poor file layout

If different files which are accessed heavily are placed on the same devices (typically the same disk spindles), they will interfere with each other. A common example of this is having several stripe sets across all the disks which interfere with each other. Another problem with layout can be a "hot" file or database area—this can sometimes be fixed by striping the file across several disk spindles. Alternatively, database blocks can be depopulated (set a high "free" percentage) to "space out" the data.

Log file bottleneck

Transactional systems (most often databases) keep a log of all the changes made to the system. This is vital in rebuilding the system after failure. In a system with much updating, this file can become a bottleneck. Ensure that it runs on its own (mirrored) spindle(s). Options for further improvement include splitting the database into several partitions (each having its own log file), using a storage system with a cache (supported by an Uninterruptible Power Supply) and scheduling bulk updates to a later time (or other application and configuration changes to reduce updating).

Tracing (or debug etc.) left on

It happens; so it's worth checking.

Operations packaged poorly

This is very common with multiple tier database applications—many separate requests are sent between client and server, the client processing the data that the server returns. Dramatic performance improvements can be obtained through the use of database procedures, which perform several operations for a single client request, and move data processing onto the server. Rather than send 100 rows over the wire to process on the client, for example, a database procedure can process these locally and send only the result. There are other analogous situations to this where many client-server conversations are had unnecessarily.

Failure to take expert advice

Another common failing is to employ technology in an inappropriate way due to lack of experience. Getting an expert to provide a "sanity check" can be very valuable, but is often overlooked.

Database query execution plans

This area can provide performance improvements of several orders of magnitude. Having an expectation of system performance is particularly important here. Long database queries may well be the result of an inappropriate algorithm for processing the data—called a query execution plan. These can be improved as a result of "hints" on the query, by improving the statistics used by the database to arrive at its plans, or by changing the way the query is phrased to return the same data in a different way.

Conclusions

Many people think that performance is an esoteric art, but in reality it is all about common sense, keeping a level head and seeing the wood for the trees. This article has tried to give some systematic approaches to performance, but there follow some simple truths which if observed will avoid many issues with performance.

1. Consider Performance Engineering at every stage of the development life-cycle.

2. Whenever you do some performance work, step back and think about what questions you are trying to answer.

3. Performance is all about understanding. What are the requirements? Where is the evidence of poor performance? How does the system work?

4. Capture your understanding in performance models.

5. Take a broad view. Some performance meters may seem a bit high, but what is the effect on the user performance? Is the computer system the root cause of perceived problems?

6. Whenever performance issues are identified they must be given an owner or they will not be resolved.

Acknowledgements

The authors would like to thank John Popplewell for all his assistance in the preparation of this paper. John is an ICL Distinguished Engineer renowned for his performance expertise.

Biographies

Stuart Forbes

Stuart Forbes is a Systems Architect based in Manchester. He has worked on a variety of development and customer projects specialising in performance. He is currently involved in the design of enterprise Windows NT solutions.

Stuart joined ICL in 1986, he is a member of the IEE and an ICL Distinguished Engineer.

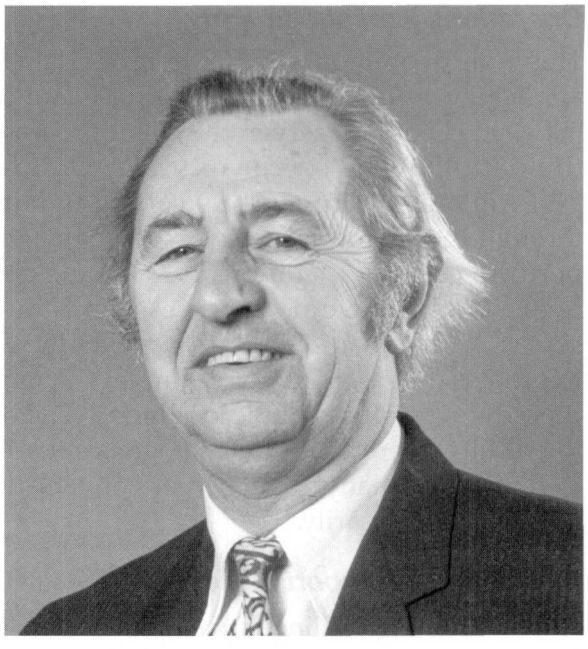
email address is stuart.forbes@icl.com.

Ben Thornton

Ben Thornton is a Systems Architect who has spent over ten years working on databases and enterprise level Operating Systems, specialising in their performance. He is a member of the British Computer Society and a Chartered Engineer.

email Ben.Thornton@icl.com.
personal web site at www.wuli.demon.co.uk.

Obituary—Jack Howlett



Dr Jack Howlett, CBE, the founding editor of the ICL Systems Journal and an early computer pioneer, died on 5th May 1999 at the age of eighty six.

Professor Victor Maller (present Editor of the Systems Journal) writes:

Jack remained on the Editorial Board after he retired from the editorship and continued to make an immense contribution to the success of the Journal. I first met him some twenty five years ago and wish to take this opportunity to express my particular appreciation for the unfailing help, encouragement and support I always received. His colleagues on the Board will greatly miss both his wise counsel and his friendship.

Jack was a man of many parts. He was a mathematician by profession, whose first job, after gaining his PhD in 1936, was to apply statistical methods to coal consumption on the London, Midland and Scottish Railway. During the Second World War he was a member of a team, under Professor Hartree at Manchester University, engaged in applying mechanical computational methods, such as the "differential analyser", to various wartime problems, including work on "Tube Alloys", the atomic bomb project, although Jack was very reticent about discussing this work.

After the war, he joined the Atomic Energy Research Establishment at Harwell to lead the Computing Section of the Theoretical Physics Division and then, in 1961, he became the first Director of the Atlas Computer Laboratory, where he remained until his "first" retirement in 1975. Jack was recruited by Peter Hall to ICL as a consultant and started the ICL Technical Journal, as it was then called, producing the first issue in 1978.

Jack was also a linguist with a fluent command of French and used this facility to translate books on computing into English. He had a wide knowledge of literature, music and art and would speak with equal authority on subjects ranging from the unaccompanied violin works of Bach to the paintings of Renoir.

He was appointed CBE in 1969, was a Fellow of St Cross College, Oxford from 1966, a Fellow of the British Computer Society, a Fellow of the Statistical Society, a Member of the London Mathematical Society and a Member the Institution of Electrical Engineers.

I asked Jack, not so long ago, what his recipe was for continuing physical and mental alertness. He looked at me with a twinkle in his eye and said, "Plenty of good food, good wine and a job with commitment." There was no answer to that.

Jack is survived by his wife, Joan, four sons and a daughter.

Friends and colleagues of Jack, who knew him for at least part of his professional career, offered the following contributions to this obituary, for which I am extremely grateful.

Michael Kay (ICL Fellow and member of the Editorial Board) writes:

Jack Howlett was the founding editor of the ICL Technical Journal, which became the ICL Systems Journal in 1996, and remained an active member of the Editorial Board from its inception in 1978 until his death in May 1999 at the age of 86.

Jack's vision for the Journal was established clearly in the Foreword to the first issue, which appeared above the signature of the then Managing Director, Christopher Wilson:

The electronic digital computer is one of the most important inventions of all time. ICL has been in this development from the very beginning, and has made some of the most important contributions. This has been possible because we have always had on our staffs people of great technical excellence and lively imagination, and we must have people of this kind if we are to continue as one of the leaders in this complex, exacting and fast-changing industry.

Such people are always thinking independently and along novel lines, whether producing new ideas or looking afresh, and often critically, at existing products, practices and concepts. A great amount of this type of thinking, with attendant discussion, is always going on inside ICL, far more than is generally realised. Much of it should be more widely known because it contributes to the knowledge and development of the computational art.

Throughout his time as Editor, and subsequently, he strove to achieve these two complementary objectives: of communicating ICL's technical ideas to a wider audience, and of encouraging and recognising ICL's best engineers.

Getting the Journal established in ICL was a significant achievement, especially for a man who was not an ICL insider. He had to persuade the technical people to contribute papers, he had to persuade the management to maintain the funding, he had to persuade the sales force to make it available to the customers, and he had to persuade sometimes sceptical marketing people that releasing all this technical information was in the company's best interests. He had to do all this, moreover, without sacrificing the integrity and quality of the publication. The fact that the Journal still exists more than twenty years later shows just how well he succeeded—there are very few products or publications in ICL that have lasted as well.

Jack achieved this by charming everyone. He had a way of making people feel appreciated, of seeing things from their point of view, and quite simply of making everyone want to help him achieve his objectives.

He had a very strong vision of what the Journal should be: a serious publication worthy to take its place on the shelves of a University Library, at the same time a promotional vehicle for ICL to demonstrate its engineering achievements to the world; a journal that existed for the benefit of its readers but also a training ground for its authors.

Jack's contribution to ICL was recognised last year when he became a member of the ICL majority club, which employees are invited to join after 21 years with the company. To acknowledge the unique feat of giving 21 years of service, having only joined the company at the age of 65, he was awarded the exceptional honour of a double majority, normally attained only after 42 years of service.

Before coming to ICL, Jack had a distinguished career as Director of the Atlas Computer Laboratory at AERE Harwell. Born in 1912 and educated at Stand Grammar School, Manchester and at Manchester University, his early training was in Mathematics, and his abiding interest in numerical analysis. His first job was with the Research Department of the LMS Railway Company. He often described his relief when one of his calculations—on the efficiency of a new braking system—was tested successfully with a real live steam locomotive, complete with an experienced and sceptical footplate crew with little inclination to show respect for a youngster and his paper calculations. Apparently the engine came to a stop within inches of the predicted place.

In 1940 Jack joined a team under Professor Hartree at Manchester University building a mechanical differential analyser, and using it to perform a variety of complex calculations, including some that contributed to the atomic bomb project. This work led naturally to Jack's appointment as head of the Computing Section in the Theoretical Physics Division at Harwell in 1948. In serving all the computational needs of the establishment, the Computing Section soon outgrew its parent department.

Harwell acquired a Ferranti Mercury machine in 1958, but the growing computational needs led to the planning of the Atlas project, a collaboration with Manchester University and Ferranti. A decision was made that the machine should be installed at Harwell, but controlled by the National Institute for Research in Nuclear Sciences, and that it should make computer facilities

available to all UK universities. What might have been an uneasy management compromise turned into a great success under Jack's leadership, making the Atlas Computer Laboratory into a powerful research centre and a magnet for the most able academics, particularly in numerical methods. Jack was Director of the Laboratory from its creation in 1961 until 1975, when he relinquished the post in order to chair the National Committee on Computer Networks.

Jack's interests were very wide. He always stayed abreast of new ideas in computing, and applied a firm critical judgement to every paper he edited. He was keen to publish good quality historical research, but kept this very much in a secondary role, a policy which the Journal has maintained to this day. In retirement, as well as editing the ICL Technical Journal, Jack exercised his linguistic abilities by translating books on computing from French, often improving them as he did so. He loved walking in the Lake District, and took a great interest in music and painting. He combined a sharp intellect, which never faded, with a thoroughly practical approach to getting things done, and an old-fashioned courtesy to all those he worked with. He hosted meetings of the Editorial Board with panache in the relaxed and civilised atmosphere of his London club, ensuring that there was always food, wine and conversation to enjoy after the official business had been efficiently disposed of. Between meetings, he got the Journal out on time with unfailing regularity, helping errant authors to achieve their deadlines, and doing anything that needed doing cheerfully: if all else failed he would address the envelopes himself.

ICL was privileged that a man of Jack's stature should take on the task of editor, should perform the role in such a distinguished manner, and should continue to support the Journal for such a long period of time.

Peter Hall (retired Director of ICL) writes:

I would like to describe an event which occurred in the early 60s which I think illustrates the fact that Jack could inspire such trust in people and organisations that bureaucracy and red tape were swept aside and things got done, which, in his absence, would never have happened, or, if they did, it would be years later.

In the early 1960s we in Ferranti found ourselves spending a fortune on the development of the Manchester University "fastest and largest computer in the world". The trouble was that the University and ourselves had only aspirations as to how fast it would actually go, and it would be sometime before we knew. Meanwhile I was committing more and more of the Ferranti fortune—at that time it really was family money—with no certainty of getting any of it back. Meanwhile a guy called Jack Howlett seemed to want one—he was obviously our best bet.

I explained to Jack that to order an Atlas we needed a contract from the authority which committed them to over £3 million for a machine which we could not specify in performance, reliability or delivery date. £3 million was, I think, about thirty times what any one had paid for a computer before. Jack, as you would expect, did his best to get a commitment from us, but all we could offer was our "best endeavours". So he worked on the contracts men with the result that we got a contract for the price we wanted, for delivery sometime, of a machine of unspecified performance and reliability, but which performed to the satisfaction of the authority—which meant Jack—and no nasty small print. This contract depended totally on trust between Jack and the authority on the one hand, and Jack and Ferranti—essentially me—on the other. It all worked out. It enabled the Atlas Laboratory to get off the ground, and Ferranti to establish the Atlas computer in the market place; and importantly to rescue some of the Ferranti family's money.

To my knowledge there has never been a computer contract like it since. I truly believe it would never have been possible without Jack. It was not just his understanding of the technical issues, but more importantly his obvious honesty and integrity. The authority knew that Jack would not accept anything which was less than could reasonably be expected, and I knew that he would not demand of us anything that was unreasonable.

It was my relationship with Jack during the Atlas days that led me, on his retirement, to ask him to join us and start the ICL Technical Journal. The success of the Journal owes everything to Jack. I mention this only because it

clearly illustrates what was perhaps most dominant in his character—his ability to get on with people of all sorts—make them feel at ease—make them feel great—and to get the most out of them.

Our friendship, born out of the Atlas contract nearly 40 years ago, lasted until the end. I shall remember him for lots of things. Staying with us on the way to his annual walking holiday in the Lake District; with Joan entertaining my wife and me at concerts in Abingdon; the four of us at the opera in London; our annual lunch with a few ICL colleagues; great times together at Dickens Pickwick Club dinners and, not least, fine wine, good food and good company at his favourite place in London—the Savile Club.

I shall also remember him as one who was always concerned with the health and well being of his many friends. We will miss him.

Previous Issues

Vol. 10 Iss. 2 – November 1995

The Architecture of the ICL GOLDRUSH MegaSERVER
The Hardware Architecture of the ICL GOLDRUSH MegaSERVER
CAL in Higher Education – Potential and Pitfalls
The UK Technology Foresight Programme
Making the Internet Safe for Business
Developing Financial Services Kiosks
High Availability Manager
The Virgin Global Challenger
Design of the Format for EDI Messages Using Object–Oriented Techniques
New Aspects of Research on Displays

Vol. 10 Iss. 1 – May 1995

Object databases and their role in multimedia information systems
The ICL Multimedia Desktop Programme
Multimedia Information used in Learning Organisations
The Software Paradigm
Single Sign–on Systems
Why is it difficult producing safety–critical software?
Experiences using the Ingres Search Accelerator for a Large Property Management Database System
RAID
Improving Configuration Management for Complex Open Systems

Vol. 9 Iss. 2 – November 1994

Establishing Co-operation in Federated Systems
An ANSA Analysis of Open Dependable Distributed Computing
An Open Architecture for Real–Time Processing
Updating the Secure Office System
POSIX Security Framework
SQL Gateways for Client-Server Systems
Asynchronous transfer mode – ATM
The ICL search accelerator™, SCAFS™: functionality and benefits
Open Teleservice – A Framework for Service in the 90s
LEO, A personal memoire

Vol. 9 Iss. 1 – May 1994

Client-server architecture
How ICL Corporate Systems support Client-server: an Architectural Overview
Exploiting Client-server Computing to meet the needs of Retail Banking Organisations
A practical example of Client-server Integration
From a Frog to a Handsome Prince: Enhancing existing character based mainframe applications
Legacy systems in client-server networks: A gateway employing scripted terminal emulation
The Management of Client-server Systems
Dialogue Manager: Integrating disparate services in client–server environments
Distributed Printing in a Heterogeneous World
Systems Management: an example of a successful Client–server Architecture
PARIS – ICL's Problem & Resolution Information System

Vol. 8 Iss. 4 – November 1993

Toward the 4th Generation Office: A Study in Office Systems Evolution
IPCS – Integrated Product Configuring Service
CGS – The ICL Configurer Graphics Service
Location Transparency in Heterogeneous Networks
Future Office Interconnection Architectures for LAN and Wide Area Access
Parallel Lisp and the Text Translation System METAL on the European Declarative System
Detecting Latent Sector Faults in SCSI Disks

Vol. 8 Iss. 3 – May 1993

An Introduction to OPEN*framework*
The Evolution of the OPEN*framework* Systems Architecture
Creating Potential for Change
OPEN*framework* in Action at DEVETIR

Strategic Information Systems planning: A Process to Integrate IT and Business Systems
Describing Systems in the OPEN*framework* Integration Knowledge Base
Multimedia and Standards for Open Information
VME–X: Making VME Open
A New Approach to Cryptographic Facility Design
CHISLE: An Engineer's Tool for Hardware System Design
Distributed Detection of Deadlock

Vol. 8 Iss. 2 – November 1992

Open Networks – The Key to Global Success
Infrastructure of Corporate Networks in the Nineties
Broadband Networking
FDDI – The High Speed Network of the Nineties
The Evolution of Wireless Networks
Communications Technology for the Retail Environment
RIBA – A Support Environment for Distributed Processing
Information Technology: Support for Law Enforcement Investigations and Intelligence
Standard for Keyboard Layouts – The Origins and Scope of ISO/TEC 9995
ESS – A Solid State Disc System for ICL System for ICL Series 39 Mainframes

Vol. 8 Iss. 1 – May 1992

Defining CASE Requirements
ICL's ICASE Products
The Engineering Database
CASE Data Integration: The Emerging International Standards
Building Maintainable Knowledge Based Systems
The Architecture of an Open Dictionary
The Use of a Persistent Language in the Implementation of a Process Support System
ALF: A Third Generation Environment for Systems Engineering
MASP/DL: The ALF Language for Process Modelling
The ALF User Interface Management System
A New Notation for Dataflow Specifications

Vol. 7 Iss. 4 – November 1991

Systems Management: A Challenge for the Nineties – Why now?
The Evolution within ICL of an Architecture for Systems Management
Manageability of a Distributed System
Distribution Management – ICL's Open Approach
Experience of Managing Data Flows in Distributed Computing in Retail Businesses
Generation of Configurations – a Collaborative Venture
Operations Management
OSMC: The Operations Control Manager
The Network Management Domain
An Overview of the Raleigh Object–Oriented Database System
Making a Secure Office System
Architectures of Knowledge Base Machines
The Origins of PERICLES – A common on–line Interface

Vol. 7 Iss. 3 – May 1991

Introduction to the technical characteristics of ISDN
ISDN in France: Numéris and its market
The Telecoms Scene in Spain
Future Applications of ISDN to Information Technology
A Geographical Information System for Managing the Assets of a Water Company
Using Constraint Logic Programming Techniques in Container Port Planning
Locator – An Application of Knowledge Engineering to ICL's Customer Service
Designing the HCI for a Graphical Knowledge Tree Editor: A Case Study in User-Centred Design
X/OPEN – From Strength to Strength
Architectures of Database Machines
Computer Simulation for the Efficient Development of Silicon Technologies
The use of Ward and Mellor Structured Methodology for the Design of a Complex Real Time Syste

Vol. 7 Iss. 2 – November 1990

The SX Node Architecture
SX Design Process
Physical Design Concepts of the SX Mainframe
The Development of Marketing to Design: The Incorporation of Human Factors into Specification and Design
Advances in the Processing and Management of Multimedia Information
An Overview of Multiworks
RICHE–Rèseau d'Information et de Communication Hospitalier Européen (Healthcare Information and Communication Network for Europe)
E.S.F – A European Programme for Evolutionary Introduction of Software Factories
A Spreadsheet with Visible Logic
Intelligent Help – The Results of the EUROHELP Project
How to use Colour in Displays – Coding, Cognition and Comprehension
Eye Movements for A Bidirectional Human Interface
Government IT Infrastructure for the Nineties (GIN): An Introduction to the Programme

Vol. 7 Iss. 1 – May 1990

Architecture of the DRS6000 (UNICORN) Hardware
DRS6000 (UNICORN) software: an overview
Electromechanical Design of DRS6000 (UNICORN)
The User–System Interface – a challenge for application users and application developers?
The emergence of the separable user interface
SMIS – A Knowledge–Based Interface to Marketing Data
A Conversational Interface to a Constraint–Satisfaction System
SODA: The ICL interface for ODA document access
Human – Human co–operation and the design of co–operative mechanisms
Regulatory Requirements for Security – User Access Control
Standards for secure interfaces to distributed applications
How to Use Colour in Displays – 1. Physiology Physics & Perception

Vol. 6 Iss. 4 – November 1989

Time to Market in new product development
Time to Market in manufacturing
The VME High Security Option
Security aspects of the fundamental association model
An introduction to public key systems and digital signatures
Security classes and access rights in a distributed system
Building a marketeer's workbench: an expert system applied to the marketing planning process
The Knowledge Crunching Machine at ECRC: a joint R&D project of a high speed Prolog system
Aspects of protection on the Flagship machine: binding, context and environment
ICL Company Research and Development Part 3: The New Range and other developments

Vol. 6 Iss. 3 – May 1989

Tools, Methods and Theories: a personal view of progress towards Systems Engineering
Systems Integration
An architectural framework for systems
Twenty Years with Support Environments
An Introduction to the IPSE 2.5 Project
The case for CASE
The UK Inland Revenue operational systems
La solution ICL chez Carrefour a Orleans
A Formally–Specified In–Store System for the Retail Sector towards a Geographic Information System
Ingres Physical Design Adviser: a prototype system for advising on the physical design of an Ingres relational database
KANT – a Knowledge Analysis Tool
Pure Logic Language
The 'Design to Product' Alvey Demonstrator

Vol. 6 Iss. 2 – November 1988

Flexible Manufacturing at ICL's Ashton plant
Knowledge based systems in computer based manufacturing
Open systems architecture for CIM
MAES – An expert system applied to the planning of material supply in computer manufacturing

JIT and IT
Computer Aided Process Planning (CAPP): Experience at Dowty Fuel Systems
Use of integrated electronic mail within databases to control processes
Value engineering – a tool for product cost reduction
ASP: Artwork specifications in Prolog
Elastomer technology for probing high-density printed circuit boards
The effects of back-driving surface mounted digital integrated circuits
Reliability of surface-mounted component soldered joints produced by vapour phase, infrared soldering techniques
Materials evaluation
On the human side of technology

Vol. 6 Iss. 1 – May 1988

ICL Series 39 support process
The ICL systems support centre organisation
ICL Services Product Centre
Knowledge engineering as an aid to the system service desks
Logic analysers for system problem solving
Repair – past and future
OSI migration
A Network to Support Application Software Development
Universal Communications Cabling: A Building Utility
Collecting and generalising knowledge descriptions from task analysis data
The architecture of an automated Quality Management System
ICL Company Research and Development Part 2: Mergers and Mainframes, 1959–1968

Vol. 5 Iss. 4 – November 1987

Open Distributed Processing
The Advanced Network Systems Architecture project
Community management for the ICL networked production line
The X/OPEN Group and the Common Applications Environment
Security in distributed information systems: needs, problems and solutions
Cryptographic file storage
Standards and office information
Introducing ODA
The Technical and Office Protocols – TOP
X400 – international information distribution
A general purpose natural language interface: design and application as a database front-end
DAP–Ada: Ada facilities for SIMD architectures
Quick language implementation

Vol. 5 Iss. 3 – May 1987

What is Fifth Generation? – the scope of the ICL programme
The Alvey DHSS Large Demonstrator Project
PARAMEDICL: a computer-aided medical diagnosis system for parallel architectures
S39XC – a configurer for Series 39 mainframe systems
The application of knowledge-based systems to computer capacity management
On knowledge bases at ECRC
Logic languages and relational databases: the design and implementation of Educe
The semantic aspects of MMI
Language overview
PISA – a Persistent Information Space Architecture
Software development using functional programming languages
Dactl: a computational model and compiler target language based on graph reduction
Designing system software for parallel declarative systems
Flagship computational models and machine architecture
Flagship hardware and implementation
GRIP: a parallel graph–reduction machine

Vol. 5 Iss. 2 – November 1986

The Management into the 1990s Research Programme
Managing strategic ideas: the role of the computer
A study of interactive computing at top management levels

A management support environment
Managing change and gaining corporate commitment
An approach to information technology planning
Preparing and organising for IPSE
Global Language for Distributed Data Integration
The design of distributed secure logical machines
Mathematical logic in the large practical world
The ICL DRS300 management graphics system
Performance of OSLAN local area network
Experience with programming parallel signal–processing algorithms in Fortran 8X

Vol. 5 Iss. 1 – May 1986

ICL company research and development, 1904–1959
Innovation in computational architecture and design
REMIT: a natural language paraphraser for relational query expressions
Natural language database enquiry
The *me too* method of software design
Formal specification – a simple example
The effects of inspections on software quality and productivity
Recent developments in image data compression for digital facsimile
Message structure as a determinant of message processing system structure

Vol. 4 Iss. 4 – November 1985

History of the ICL content-addressable file store, (CAFS)
History of the CAFS relational software
The CAFS system today and tomorrow
Development of the CAFS-ISP controller product for Series 29 and 39 systems
CAFS–ISP: issues for the applications designer
Using secondary indexes for large CAFS databases
Creating an end-user CAFS service
Textmaster – a document retrieval system using CAFS–ISP
CAFS and text: the view from academia
Secrets of the sky: the IRAS data at Queen Mary College
CAFS file-correlation unit

Vol. 4 Iss. 3 – May 1985

Overview of the ICL Series 39 Level 30 system
VME nodal architecture: a model for the realisation of a distributed system concept
Processing node of the ICL Series 39 Level 30 system
Input/output controller and local area networks of the ICL Series 39 Level 30 system
The store of the ICL Series 39 Level 30 system
The high–speed peripheral controller for the Series 39 system
Development of 8000–gate CMOS gate arrays for the ICL Level 30 system
Development route for the C8K 8000–gate CMOS array
Design automation tools used in the development of the ICL Series 39 Level 30 system
Design and manufacture of the cabinet for the ICL Series 39 Level 30 system
Manufacturing the level 30 system I Mercury: an advanced production line
Manufacturing the Level 30 system II Merlin: an advanced printed circuit board manufacturing system
Manufacturing the Level 30 system III The test system

Vol. 4 Iss. 2 – November 1984

Modelling a multi–processor designed for telecommunication systems control
Tracking of LSI chips and printed circuit boards using the ICL Distributed Array Processor
Sorting on DAP
User functions for the generation and distribution of encipherment keys
Analysis of software failure data(1): adaptation of the Littlewood stochastic reliability growth model for coarse data
Towards a formal specification of the ICL Data Dictionary

Vol. 4 Iss. 1 – May 1984

The ICL University Research Council
The Atlas 10 computer
Towards better specifications

Solution of the global element equations on the ICL DAP
Quality model of system design and integration
Software cost models
Program history records: a system of software data collection and analysis

Vol. 3 Iss. 4 – November 1983

Expert system in heavy industry: an application of ICLX in a British Steel Corporation works
Dragon: the development of an expert sizing system
The logic language PROLOG–M in database technology and intelligent knowledge–based systems
QPROC: a natural language database enquiry system implemented in PROLOG
Modelling software support

Vol. 3 Iss. 3 – May 1983

IPA networking architecture
IPA data interchange and networking facilities
The IPA telecommunications function
IPA community management
MACROLAN: a high–performance network
Specification in CSP language of the ECMA–72 Class 4 transport protocol
Evolution of switched telecommunication networks
DAP in action

Vol. 3 Iss. 2 – November 1982

The advance of Information Technology
Computing for the needs of development in the smallholder sector
The PERQ workstation and the distributed computing environment
Some techniques for handling encipherment keys
The use of COBOL for scientific data processing
Recognition of hand–written characters using the DAP
Hardware design faults: a classification and some measurements

Vol. 3 Iss. 1 – May 1982

Software of the ICL System 25
Security in a large general–purpose operating system: ICL's approach in VME/2900
Systems evolution dynamics of VME/B
Software aspects of the Exeter Community Health Services Computer Project
Associative data management system
Evaluating manufacturing testing strategies

Vol. 2 Iss. 4 – November 1981

Architecture of the ICL System 25
Designing for the X25 telecommunications standard
Viewdata and the ICL Bulletin System
Development philosophy and fundamental processing concepts of the ICL Rapid Application Development System RADS
A moving–mesh plasma equilibrium problem on the ICL Distributed Array Processor

Vol. 2 Iss. 3 – May 1981

A dynamic database for econometric modelling
Personnel on CAFS: a case study
Giving the computer a voice
Data integrity and the implications for back–up
Applications of the ICL Distributed Array Processor to econometric computations
A high–level logic design system
Measures of programming complexity

Vol. 2 Iss. 2 – November 1980

The ICL Information Processing Architecture, IPA
VME/B: a model for the realisation of a total system concept
Birds, Bs and CRTs
Solution of elliptic partial differential equations on the ICL Distributed Array Processor
Data routing and transpositions in processor arrays

A Bayesian approach to test modelling

Vol. 2 Iss. 1 – May 1980

Security and privacy of data held in computers
CADES – software engineering in practice
ME29 Initial Program Load: an exercise in defensive programming
Project Little – an experimental ultra–reliable system
Flow of instructions through a pipelined processor
Towards an 'expert' diagnostic system
Using Open System Interconnection standards

Vol. 1 Iss. 3 – November 1979

Meteosat 1: Europe's first meteorological satellite
An analysis of checkpointing
Statistical and related systems
Structured programming techniques in interrupt–driven routines
The content addressable file store – CAFS
Computing in the humanities
The data dictionary system in analysis and design

Vol. 1 Iss. 2 – May 1979

Computers in support of agriculture in developing countries
Software and algorithms for the Distributed Array Processor
Hardware monitoring on the 2900 range
Network models of system performance
Advanced technology in printing: the laser printer
The new frontier: three essays on job control

Vol. 1 Iss. 1 – November 1978

The origins of the 2900 series
Sizing computer systems and workloads
Wind of Change
Standards for open–network operation
Distributed computing in business data processing
A general model for integrity control

To order back issues

Contact

Chrissie Wilson

Group Technical Directorate
ICL, Cavendish Road, Stevenage, Hertfordshire, SG1 2DY
Telephone +44 (0)1438 786321
Email: Chrissie.Wilson@icl.com

or

The Editor, V.A.J. Maller
Telephone +44 (0)1438 833514
Email: V.A.J.Maller@lboro.ac.uk
or: Victor.Maller@icl.com

ICL Systems Journal

Guidance for Authors

Content

The ICL Systems Journal has an international circulation. It publishes papers of a high standard that are related to ICL's business and is aimed at the general technical community and in particular at ICL's users, customers and staff. The Journal is intended for readers who have an interest in computing and its applications in general but who may not be informed on the topic covered by a particular paper. To be acceptable, papers on more specialised aspects of design or application must include some suitable introductory material or reference.

The Journal will not usually reprint papers already published but this does not necessarily exclude papers presented at conferences. It is not necessary for the material to be entirely new or original. Papers will not reveal material relating to unannounced products of any of the ICL Group of companies.

Letters to the Editor and book reviews may also be published.

Authors

Within the framework defined in paragraph 1, the Editor will be happy to consider a paper by any author or group of authors, whether or not employed by a company in the ICL Group. All papers will be judged on their merit, irrespective of origin.

Length

There is no fixed upper or lower limit, but a useful working range is 4,000-8,000 words; it may be difficult to accommodate a long paper in a particular issue. Authors should always keep brevity in mind but should not sacrifice necessary fullness of explanation.

Abstract

All papers should have an Abstract of approximately 200 words, suitable for the various abstracting journals to use without alteration.

Presentation

Printed (typed) copy

A typed copy of the manuscript, single sided on A4 paper with the pages numbered in sequence, should be sent to the Editor. Particular care should be taken to ensure that mathematical symbols and expressions, and any special characters such as Greek letters, are clear. Any detailed mathematical treatment should be put in an Appendix so that only essential results need be referred to in the text.

Electronic version

Authors are encouraged to submit either a magnetic disk version of their manuscript or a compressed e-mail attached file or both. The format of the file should conform to the standards of any of the widely used word processing packages or be a simple text file.

Diagrams

Line diagrams will usually be redrawn and professionally lettered for publication, so it is essential that the originals are clear. Axes of graphs should be labelled with the relevant variables and, where this is desirable, marked off with their values. All diagrams should be numbered for reference in the text and the text marked with the reference and an appropriate caption to show where each should be placed. Authors should check that all diagrams are actually referred to in the text and that copies of all diagrams referred to are supplied. Authors wishing to submit drawings in electronic form should ensure that they are separated from the main text and, ideally, be in the form of EPS files. If an author wishes to use colour, then it is very helpful that a professional drawing package be used, such as Adobe Illustrator. PowerPoint diagrams, however, may be used to indicate the author's requirements.

Photographs

Authors who wish to include photographs in their papers should provide good quality slides or prints, not digital images unless they have been taken with top quality professional equipment. High resolution scanned images are also acceptable in tiff or jpeg format.

Tables

As with diagrams, these should all have captions and reference numbers. If they are to be provided in electronic form, then either a standard spreadsheet (Excel) should be used or the data supplied as a file of comma/tab separated variables. A printed version should also be supplied, showing all row and column

headings, as well as the relevant units for all the quantities tabulated.

References

Authors are asked to use the Author/Date system, in which the author(s) and the date of the publication are given in the text, and all the references are listed in alphabetical order of author at the end; e.g. in the text: "...further details are given in [Henderson, 1986]" with the corresponding entry in the reference list:

> HENDERSON, P., "Functional Programming Formal Specification and Rapid Prototyping," IEEE Trans. on Software Engineering SE 12, 2, 241-250, 1986.

Where there are more than two authors it is usual to give the text reference as "[X et al ...]".

Authors should check that all text references are listed; references to works not quoted in the text should be listed under a heading such as Bibliography or Further reading.

Style

The Editor aims to maintain a consistent style in terms of structure, grammar and spelling. The Oxford English Dictionary and its companion works form the basic reference and the Editor will be pleased to advise authors upon request.

Referees

The Editor may refer papers to independent referees for comment. If the referee recommends revisions to the draft, the author will be asked to make those revisions. Referees are anonymous. Minor editorial corrections, to conform to the Journal's general style for spelling, punctuation or notation, will be made by the Editor.

Proofs, Offprints

Proofs are sent to authors for correction before publication, usually in electronic form, either as PDF or as output files from the production system used for the Journal: PageMaker, InDesign, Illustrator and Photoshop.

Copyright

ICL Systems Journal Autumn 1999

ICL Group Technical Directorate
Cavendish Road
Stevenage
Hertfordshire
SG1 2DY