ICL TECHNICAL JOURNAL

Volume 8 Issue 4 November 1993

Published by
International Computers Limited
at
Oxford University Press

ICL

ICL TECHNICAL JOURNAL

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

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

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

1993 subscription rates: annual subscription £60 UK and Europe and $120 rest of world; single issues £36 UK and Europe and $72 rest of world. Orders with remittances should be sent to the Journals Subscriptions Department, Oxford University Press, Walton Street, Oxford OX2 6DP.

Typeset, Printed and Bound by The Charlesworth Group, Huddersfield, UK, 0484 517077

ISSN 0142–1557

ICL

Contents

Volume 8 Issue 4

Résumés

David Hollingsworth
Architecture des systèmes de bureautique, systèmes client-serveur, ICL, Bracknell, Royaume-Uni
Le bureau de quatrième génération: étude de l'évolution de la bureautique

L'évolution des systèmes de bureautique est gérée depuis vingt ans par les développements technologiques et les exigences toujours plus importantes du monde des affaires. Cet article analyse l'impact de cette évolution sur l'architecture des systèmes de bureautique actuels. Il étudie l'origine de la bureautique, la machine de traitement de texte autonome, puis l'incorporation de fonctions telles que le courrier électronique et les échanges d'informations, enfin, la génération actuelle, caractérisée par la structure client-serveur, les applications de productivité de groupe et les informations multisupport. L'article examine alors l'avenir de l'architecture des systèmes de bureautique de quatrième génération.

C. W. Bartlett
Consultant indépendant, Manchester, Royaume-Uni
IPCS – Integrated Product Configuring Service

En 1991–1992, GPT-NSG (Network Systems Group de GPT), en collaboration avec ICL, a installé des systèmes experts de configuration destinés à être utilisés par le personnel d'assistance technique, dans le cadre de la première phase de réponse à un appel d'offre (ITT) et de la génération d'instructions de fabrication détaillées relatives aux commandes passées par ces clients.

GPT-NSG est un fabricant britannique important de composants de réseaux de télécommunications. L'utilisation des systèmes de configuration a eu pour conséquence une réduction significative des temps de réponse à un appel d'offre et a permis de produire des devis dont les composants et les prix se sont avérés plus précis.

Après un investissement initial au niveau des systèmes et de la méthodologie relativement élevé, mais rentabilisé en un an d'utilisation, le service peut maintenant être étendu rapidement et sans investissement important.

L'étude présentée dans cet article démontre que la teçhnologie utilisée par la fabrication de systèmes experts est maintenant capable de répondre aux exigences de projets à grande échelle.

S. M. Sharman
Département d'informatisation des collectivités locales et des services médicaux
CGS – le service graphique de configuration ICL

CONFIGURER permet depuis 1986 au personnel ICL de générer des propositions de ventes relatives au matériel fabriqué par la société. L'utilisateur de ce système peut ainsi déterminer avec précision la configuration du système requis par le client, lors d'une séance interactive de questions et de réponses. Des règles complexes sont alors appliquées au système proposé, dans le but de vérifier sa conformité à deux niveaux: premièrement, que la configuration est techniquement correcte, deuxièmement, afin de générer la liste des composants secondaires, les câbles, par exemple, qui complètent le système livré aux clients. L'article est essentiellement consacré à un composant récemment intégré au système, appelé CGS (Configurer Graphics Service). Il décrit ses origines, l'historique de son développement, les fonctions qu'il propose et définit le cadre de son utilisation.

A. Drahota
ICL
P. H. Ford, I. G. Vincent
University of Nottingham
Transparence topologique dans un réseau hétérogène

La transparence de la décentralisation d'un réseau, et en particulier, la transparence topologique, est un concept fondamental du développement d'une nouvelle génération de systèmes décentralisés. Le produit ICL DAIS gère un ensemble de mécanismes qui garantissent la transparence d'un réseau, y compris au niveau de la répartition des systèmes. Cet article décrit la mise en oeuvre du mécanisme de transparence topologique DAIS dans le cadre du projet OASIS, subventionné par DTI et SERC, et auquel ont collaboré ICL, Hydro-Electric Plc, l'université de Nottingham et Gid Ltd. Il examine en particulier la capacité du mécanisme de gérer à titre constant la transparence topologique dans un environnement hétérogène, composé d'ordinateurs hôte et de réseaux.

Uri Baran
Service Points de vente ICL, Bracknell, Royaume-Uni
Architectures futures des interconnexions de systèmes de bureautique dans les réseaux LAN et Wide Area Access

Cet article est consacré aux problèmes d'interconnexion des systèmes de bureautique futurs. Il présente une architecture dont les caractéristiques répondent à toutes les exigences de communication et examine en particulier ISDN, qu'il considère être le service le mieux adapté à l'environnement de communication. Il analyse également divers problèmes pratiques, ainsi que les technologies futures, qui interviennent dans le développement d'une architecture de ce type et garantissent sa mise en application et sa longévité.

Carsten Hammer
Siemens AG, Corporate Research and Development Dept. ZFE BT SE 43 Otto-Hahn-Ring 6 D-81739 Munich, Allemagne
Lisp parallèle et le système de traduction de texte METAL dans le cadre du système EDS (European Declarative System)

Cet article décrit le sous-système du langage de programmation Lisp du système EDS (European Declarative System). Il explique le modèle de parallélisme non lié à un type de machine particulier permettant le portage de programmes séquentiels existants sous forme parallèle. Il examine en particulier le système de traduction de langages naturels METAL et met en évidence les avantages qu'offrent les concepts de parallélisme aux programmes Lisp de grande taille.

Hannu H. Kari et Heikki Saikkonen
Department of Computer Science Helsinki University of Technology Espoo, SF-02150, Finland

Fabrizio Lombardi
Department of Computer Science Texas A&M University College Station, TX 77843, US

Detection des secteurs defectueux latents des disques SCSI

Cet article décrit de nouvelles méthodes de détection des secteurs défectueux latents dans un sous-système de disque, dont la cause correspond à la détérioration du périphérique de stockage magnétique. En règle générale, le caractère aléatoire de l'accès aux secteurs d'un disque a pour conséquence l'utilisation rare de certains secteurs. La détérioration d'un secteur qui n'est que rarement utilisé est par conséquent susceptible de ne pas être détectée immédiatement. Pour détecter la détérioration latente d'un secteur, un disque est vérifié dans sa totalité à fréquence régulière. Cet article définit un algorithme adaptatif, dont le but est de mettre à profit la période d'inactivité d'un disque utilisé régulièrement pour procéder à sa vérification, s'il est conforme aux normes d'interface SCSI-II.

Zusammenfassungen

David Hollingsworth
Office Systems Architect, Client-Server Systems, ICL, Bracknell, Großbritannien
Der Weg zum Büro der 4. Generation: eine Studie der Entwicklung von Bürosystemen

Büro-Informationssysteme haben sich in zwei Jahrzehnten ständigen Wandels entwickelt, getrieben von technischem Fortschritt und den wachsenden Anforderungen der Wirtschaft. Dieser Artikel analysiert, wie dieser Wandel die Architektur heutiger Bürosysteme beeinflußt hat. Er zeichnet die Entwicklung des Bürosystems von seinen Anfängen in auf Einzelrechnern basierenden Textsystemen über die Integration von Einrichtungen wie elektronische Post und Datenaustausch bis zur heutigen Generation von Büroprodukten nach, die sich durch Client/Server-Struktur, Groupware-Anwendungen und Multi-Media-Technik auszeichnen. Der Artikel schließt mit einer Erörterung aktueller Trends, die sich formend auf die Architektur des entstehenden Büros der 4. Generation auswirken werden.

C. W. Bartlett
Unabhängiger Berater, Manchester, Großbritannien
IPCS – Integrierter Produkt-Konfigurationsservice

Im Zeitraum 1991/92 installierte die Network Systems Group von GPT (GPT-NSG) in Zusammenarbeit mit ICL eine Reihe wissenbasierter Konfigurationssysteme für Mitarbeiter des Technischen Supports zur ersten Reaktion auf Ausschreibungen und zur Bereitstellung detaillierter Fertigungsanweisungen für empfangene Aufträge.

GPT–NSG ist ein großer britischer Lieferant von Ausrüstung für Telekommunikations-Netzwerke. Der Einsatz dieser Konfigurationssysteme führte zu schnellerer Reaktion auf Ausschreibungen und höherer Genauigkeit der angebotenen Ausrüstung und der Preisgestaltung.

Die recht hohen anfänglichen Ausgaben für Werkzeuge und Methodologie, die sich schon im ersten Betriebsjahr amortisierten, schafften eine Situation, in der der Service jetzt billig und schnell erweitert werden kann.

Die hier behandelte Fallstudie zeigt, daß die wissensbasierte Technologie, die bei der Produktion von Konfigurationssystemen eingesetzt wird, inzwischen so ausgereift ist, daß umfangreiche Projekte mit sehr hohem Vertrauen in einen erfolgreichen Abschluß in Auftrag gegeben werden können.

S. M. Sharman
Local Government and Health IT Unit (Abteilung Informationstechnik für Kommunalverwaltung und Gesundheitswesen), Bracknell, Großbritannien
CGS – Der ICL Configurer Grafik-Service

Seit 1986 steht ICL-Mitarbeitern als Hilfsmittel zur Erstellung von Hardwareangeboten das System CONFIGURER zur Verfügung. Dieses System ermöglicht es Benutzern im Dialog eine exakte Angabe der Komponenten au erhalten, die für bestimmte Kundenanforderungen benötigt werden. Auf die daraus resultierende Spezifikation wird eine Reihe komplexer Regeln angewandt, nach denen erstens geprüft wird, ob die eingegebene Konfiguration technisch korrekt ist, und zweitens wird automatisch eine Liste benötigter Zusatzteile, wie zum Beispiel Kabel, angefertigt, um die Vollständigkeit der ausgelieferten Hardwarepakete sicherzustellen. Dieser Artikel befaßt sich mit einer relativ neuen Erweiterung des Systems, die als Configurer Graphics Service bezeichnet wird. Es werden die Anfänge des Dienstes, seine Entwicklungsgeschichte, seine Funktionen sowie die technischen Verfahren, die zur Verwirklichung dieser Funktionen angewandt wurden, beschrieben.

A. Drahota
ICL
P. H. Ford, I. G. Vincent
Universität Nottingham, Großbritannien
Ortstransparenz in heterogenen Netzwerken

"Verteilungstransparenz" (Distribution Transparency) ist ein Schlüsselbegriff bei der Entwicklung einer neuen, überschaubaren Generation verteilter Systeme. Ein Aspekt der Verteilungstransparenz ist die Ortstransparenz (Location Transparency). Das ICL-Produkt DAIS implementiert umfangreiche Transparenzmechanismen, darunter einen Mechanismus für Ortstransparenz. Dieser Artikel beschreibt die Implementierung des Ortstransparenz-Mechanismus von DAIS, wie er in dem Gemeinschaftsprojekt OASIS erreicht wurde, das von ICL Hydro-Electric Plc, der Universität Nottingham und Gid Ltd. durchgeführt und vom britischen Ministerium für Industrie und Handel (DTI) und dem Forschungsrat SERC gefördert wird. Ein besonderes Merkmal dieser Implementierung ist die Fähigkeit, Ortstransparenz in einer heterogenen Umgebung mit Hostrechnern und Netzwerken aufrechtzuerhalten.

Uri Baran
ICL Retail, Bracknell, Großbritannien
Zukünftige Büro-Vernetzungsarchitekturen für den Zugriff auf lokale und Weitbereichsnetze

Dieser Artikel befaßt sich mit den Vernetzungsfragen, die bei der Entwicklung zukümftiger Büroumgebungene auftreten. Er stellt eine Architektur für alle Kommunikationsanforderungen vor und konzentriert sich auf ISDN als den Dienst, der den meisten Anforderungen am besten entspricht. Außerdem werden praktische Fragen der Bereitstellung von Mitteln zur Implementierung dieser Architektur und zukünftige Technologien zur Sicherung ihrer Dauerhaftigkeit angesprochen.

Carsten Hammer
Siemens AG Corporate Research and Development Dept. ZFE BT SE 43 Otto-Hahn-Ring 6 D-81739 München
Parallel-Lisp und das Textübersetzungssystem METAL auf EDS

Der Artikel beschreibt das Lisp-Subsystem des European Declarative Systems EDS. Das maschinenunabhängige Modell der Parallelität, das einfaches Portieren existierender sequentieller Programme in parallele Form ermöglicht, wird erklärt. Eine wichtige Anwendung ist das Übersetzungssystem METAL für natürliche Sprache, dessen Eigenschaften beschrieben werden und anhand dessen demonstriert wird, wie große Lisp-Programme von den Konzepten der Parallelität profitieren können.

Hannu H. Kari und Heikki Saikkonen
Department of Computer Science (Fachbereich Informatik) Technische Universität Helsinki Espoo, SF-02150, Finnland

Fabrizio Lombardi
Department of Computer Science (Fachbereich Informatik) Texas A & M University College Station, TX 77843, USA
Erkennung latenter Sektorfehler auf SCSI-Platten

Dieser Artikel stellt zwei neue verbesserte Verfahren zur Erkennung latenter Sektorfehler in Platten-Subsystemen vor, wie sie durch Verschlechterung des magnetischen Speichermaterials entstehen können. Der Zugriff auf die Sektoren einer Platte erfolgt oft nach einem ungleichmäßigen Muster, so daß bestimmte Sektoren nur selten genutzt werden. Im Falle der Verschlechterung des Speichermaterials auf den selten benutzten Sektoren kann ein latenter Plattenfehler lange unerkannt bleiben. Um latente Sektorfehler zu erkennen, wird die Platte periodisch geprüft. In diesem Artikel wird ein adaptiver Algorithmus für gängige Platten, die den Schnittstellennormen SCSI-II entsprechen, vorgestellt, mit dem die Leerlaufzeit der Platte zum Prüfen genutzt werden kann.

Toward the 4th Generation Office: A Study in Office Systems Evolution

David Hollingsworth
Office Systems Architect, Client-Server Systems, ICL, Bracknell, UK

Abstract

The Office Information System has evolved over two decades of change, driven by technological advances and increasing business demands. This paper analyses the impact of these changes in shaping the architecture of today's office system. It traces the evolution of the office system from its origins in standalone word processor technology, through the incorporation of facilities such as electronic mail and information interchange, into the current generation of office products characterised by client-server structure, groupware applications and multi-media information. In conclusion, the paper discusses some of the current trends which will shape the architecture of the emerging 4th Generation Office.

1 Introduction

The office has certainly been with us since the era of the quill pen, if not the clay tablet. Whilst the technology may have changed, many basic office concepts – reports, filing cabinets, mail – have existed for generations. In looking at the evolution of the modern office information system, this paper starts with the question "what is an office system?".

To quote from the IFIP working group on Office Information Systems, (Verrijn-Stuart, 1988):

> 'An **office** is the organisational aspect embodying the activity of individuals or groups of individuals within the organisation where one deals with the organisation's information streams, which are often multi-media based;
> An **office information system (OIS)** is a specific information system fulfilling the needs of the organisation in connection with some individual or group task'

These definitions, whilst general, identify several important characteristics of the office environment.

Firstly, the office is populated by individuals, generally not IT specialists and without deep expertise, or interest, in the details of the underlying office system technology. This has placed particular emphasis on ease of use and continuation of familiar office metaphors during the evolution of the electronic office.

Secondly, office activities involve individuals and, particularly, interactions between groups of individuals within, and increasingly between, organisations. The Gartner Group (Howard, 1992) described the essential characteristics of group working as "co-ordination, co-operation and communication", reflecting the fact that virtually all activities within the office form part of a wider set of business processes.

Thirdly, information used within the office is essentially multi-media in nature (combinations of text, graphics, audio, image, facsimile, etc) and, whilst the percentage captured electronically is increasing, there remains much which lies outside the scope of today's information systems, or in isolated pockets of technology with limited opportunity for integration with the wider electronic office environment.

2 Technology and Business Changes

Whilst the basic characteristics of the office environment may have remained relatively unchanged over many years, there have been very significant changes in the technology available to support the office worker and in the business environment to which the technology has been applied.

These two themes are clearly inter-related; technological advances offer the opportunity for business change, which itself generates demand for further improvements in technology to support that change.

2.1 Technological Changes

The IT and telecommunications industries have undoubtedly created the technological framework to enable business to become more global, more time-critical, more productive and more competitive. Indeed, the pace of technological change has been so rapid that it is only in looking back that one can identify the really significant events.

To quote again from Verrijn-Stuart (1988):

> 'The problem with new areas of the enterprise is the volatility of the concepts used in describing what one is engaged in. The reason is obvious. What, much later, will turn out to be fundamental is only gradually discovered. Meanwhile the attention of the growing [IT] profession shifts continually as new ideas, not all equally profound, are proposed one after the another.'

Fundamental technological change in office systems has had two key characteristics, the scope for *commodity pricing* and *critical mass* in terms of usage.

These are obvious attributes: the office market is now measured in tens of millions of users and most office tasks require interaction with other users. The telephone, typewriter, photocopier, word processor and facsimile machine bear witness to the longevity of fundamental office technologies which meet these twin criteria.

The potential significance of many technological developments was understood early in their life cycle. Ringland (1987) accurately predicted many of the implications of increasing processor, storage and transmission capabilities on the office applications of the 1990s, covering aspects such as electronic mail, multi-media and image processing. Fuller (1991) similarly identified the potential (not yet realised in the mass office market) of ISDN services to support co-operative working based on the integration of telephony and information handling technologies. However, the single development which has subsequently most shaped the office must surely be the PC.

The exploitation of the PC within the office has not been just about increasing hardware power at the desktop; it has been about the establishment of the PC at the centre of an architecture facilitating a myriad of add-on hardware cards, a software environment supporting local applications for manipulating office information, and, in conjunction with LAN technology, providing a user with access to services and information throughout the organisation.

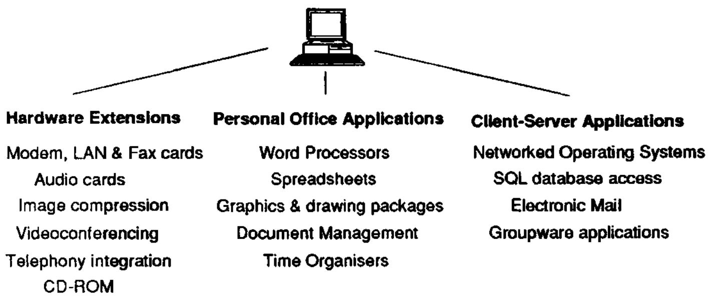

Fig. 1 The influence of the PC on Office Systems

Of course, PC technology is only part of the story; equally important have been developments in software and applications to support efficient handling of office information. Full text and keyword searching software represented an early step towards processable text. Software standards to support compound documents and the linking and embedding of information within documents have greatly increased the flexibility of electronically processable office information. Software to extract and convert structured information held in traditional computer databases has allowed a measure of integration between office systems and data processing applications. Developments continue apace; Campbell-Grant and Smethurst (1993) describe the range of standards and information relationships emerging within multimedia technology which will be an important influence on future office systems.

Major advances have occurred in the user interface (Hutt and Flower, 1990 and Edmonds, 1990), driven by the need to provide ease of use for unskilled staff. Consistent MMI styles, on-line help systems and graphical interfaces supporting windowing, icons and a mouse have enabled new applications to be adopted at a previously impossible rate. User-oriented PC scripting tools, mail filters and customisation facilities for menus and windows-based toolbars lead towards an interface which may be tailored to a user's individual preferences and working patterns.

The drive towards ease-of-use has continued into application development tools, focussing on the need to empower users to create their own simple applications quickly and without recourse to scarce and expensive DP skills. The ICL OfficePower User Defined Application facility (UDAP) is based on the concept of a simple forms-driven application and has found extensive support amongst its users; a recent survey indicated that more than 90% of OfficePower customers made use of the facility. More recently, products such as Lotus Notes have shown the significance of forms-based application tools as a vehicle for the rapid development and customisation of generic business applications at the department or workgroup level.

2.2 *Business Changes*

If the pace of technological change has been dramatic, this is no less true of its consequences on the nature of business environment. The trends towards departmental autonomy, flattening of organisational hierarchies and "empowerment" of the individual have been based on a huge upsurge in workgroup and personal IT capability. Increasingly, it is office systems which will form the backbone of an organisation's information processing in the 1990s.

Scott Morton (1991) has identified the significance of an electronically based business network as the pivot between the use of IT as an internal integration tool and its strategic exploitation as an agent of business transformation. For many organisations electronic mail has emerged as a key foundation of the business network and is now established as one of the core components of the third generation OIS.

Further progression from the business network into the strategic exploitation of IT was shown by Scott Morton to depend upon a number of factors, including:

- Flexible, co-operative processing between individuals as an organisation's business processes are redefined.
- The ability to extend, and redesign, the business network to cater for inter-organisational business relationships.

Both requirements go far beyond the use of electronic mail as a ubiquitous mechanism for information interchange. The former is being addressed, in part, by emerging groupware-based applications such as workflow management, which can exploit the interoperability facilities and user directories established by electronic mail. The latter has significant implications in areas such as information security and the legal status of electronic communications. As yet there are few standards in these areas, so technical issues of incompatibilities between applications and interchange formats may hinder progress towards global business networks. This topic is further considered in sections 4.4 and 5.7 of this paper.

3 Office Systems Evolution

The functional capabilities of the OIS have continued to advance over many years, during which numerous technologies have been introduced, some immediately successful, others not so. Despite this apparent continuum in office systems development, three distinct periods in their evolution may be identified, described by the Gartner Group (Howard, 1992) as the "three generations of office systems". Whilst there is clearly much overlap and only gradual shifts of emphasis between these alternative approaches, this separation provides a useful basis for analysis.

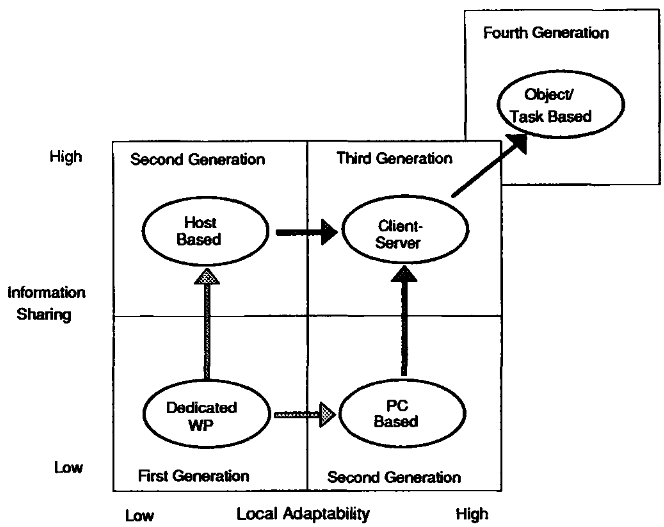

Fig. 2 The evolution of office systems

3.1 First Generation: Standalone Systems

The earliest true electronic office systems (1970s) were characterised by free-standing, dedicated word processor technology, with the emphasis on local automation of typing processes (the ICL 7700 and 8801 were typical products of this era). Although such systems were often constructed using hardware and software components derived from data communications terminals, they offered little or no capability for electronic interchange with existing DP systems or other word processing users. This stemmed partly from the limits of data communications technology – relatively low speed analogue circuits, and partly from the network structures in place – star topologies centred around mainframe computers, which themselves had few facilities for handling unstructured office data. The principal forms of office communication remained the telephone, paper mail and the telex system; there was no integration, information was simply copied or re-keyed as necessary.

Advances in the sophistication of both the word processor and document interchange followed. For example, the business form was transformed into an electronic template which could be stored and retrieved for repetitive use. The floppy disc and dial-up communications enabled simple information transfer between similar word processors and eventually to mainframe computers for storage and retrieval. Despite these advances, the first generation of office systems remained essentially a standalone capability for text manipulation.

3.2 Second Generation: Server Approach

The 1970s saw important developments in (1) the timesharing minicomputer and (2) early versions of free-standing desktop computer workstations, which were to lead the way to the personal computers of the 1980s. The second generation of office systems thus developed along two quite separate routes, one exploiting the flexibility of the emerging PC for the individual office user, the other providing the benefits of workgroup access to shared data and common office services on a multi-user host, typically a minicomputer, or in some cases, a mainframe.

Initially advances in minicomputer technology provided the opportunity to develop a more cost-effective solution to the traditional office word processing task; expensive components such as the computer platform and printers could be shared between a group of office users accessing the system through local, relatively cheap "dumb" terminals. Whilst the original motivation may have been considerations of cost and scaleability, subsequent exploitation of this approach offered major opportunities for sharing information and applications within the locally connected community of users.

Within this structure a variety of common office applications were easily developed. The underlying software on most minicomputers already provided a basic framework to support multi-user, multi-application working. Unix, for example, provides a hierarchic directory structure, including group access permissions, which could be easily mapped to the organisational structure of departments or workgroups. The process switching and memory management mechanisms enabled the separation of different user tasks with good protection yet with easy applications integration.

Vendors such as Uniplex and CCI (later absorbed into ICL), with OfficePower, offered office products built on Unix, delivering to the local workgroup not just standard word processing but also applications such as shared document libraries, electronic message passing, diary and meeting scheduling. Extensions to enable electronic interchange between different departmental systems followed, offering a basis for office systems to inter-work across an entire organisation. As the requirement to support a more global view of the office system grew, many of the group-orientated facilities were extended to operate across a network of servers, using electronic mail or other data communications services between servers and adding facilities to support a network-wide directory of users.

Broadly similar, host-based products built on proprietary software platforms were introduced by companies such as DEC (All-in-One), Wang (Wang Office) and Data General (Complete Electronic Office). During much of the 1980s this host-based approach tended to dominate the larger user organisations, because of its scaleability, manageability and support for sharing information and applications – all features difficult to achieve on emerging PC-based products. Some organisations were attracted by the security aspects of a host-based approach, where information access and interchange could be more strictly controlled and backup operations conducted from a centralised point. Moore (1991) discusses many of these considerations.

Many attributes of the server-based approach are equally valid when the host is a mainframe computer, indeed opportunities for information- and application-sharing should in theory offer advantages of scale as the number of connected users increases. However, mainframe office support systems (cf the IBM PROFS product) did not, in general, prove cost-effective and have been de-emphasised increasingly by vendors. There appear to be several reasons for this:

- The operating system architectures tended to be optimised towards transaction processing rather than the multi-user timesharing model offered by most minicomputers, resulting in comparatively resource-hungry office applications.
- The mainframe communications architectures emphasised block- and form-structured dialogue, making the handling of character-based text editing relatively inefficient.
- In an increasingly commodity market, the falling price of midrange computer platforms resulted in a systems purchase price well within the local budgets of increasingly autonomous business departments.

By failing to achieve critical mass in the market, the mainframe-based approach also failed to attract sufficient application vendors to create a full portfolio of office applications, a critical problem in a high volume market. Although none of the mainframe-based office products have made the transition into the third generation of office systems, a significant number remain in use providing electronic mail and document repository services.

3.3 Second Generation: PC Based

A weakness of the server approach using an unintelligent desktop environment is the comparative remoteness from the end-user of the user interface logic. Where character-sensitive information or document-formatting operations are being processed the response cycle from the keyboard, through the server application and back to the screen must be sufficiently fast to match the user's natural working speed. The original server products, based on directly connected text terminals and fast process switching at the host, were easily able to achieve this. However, the increasing requirement for the manipulation and integration of other forms of information such as line drawing or graphics was not so easily met by this approach.

Free-standing workstations were originally targeted at specific application areas such as engineering drawing or document composition, where significant amounts of relatively expensive processing power could be justified. However, it was the introduction and subsequent growth of the PC which really marked the start of second generation office products on the desktop.

It was not immediately apparent that the PC would become the strategic office desktop device. Early software packages for word processing and other office functions offered little functionality additional to that of server-based approaches; MS-DOS and other PC operating systems presented a complex command-line interface not well suited to office staff unskilled in computer technology and initial PC products had very limited communications capability.

However, each of these limitations was overcome in turn:

- The sheer growth in the PC market volume coupled with the emergence of MS-DOS as the de facto operating system, ensured the availability of a wide range of applications software at increasingly attractive prices. Over time the focus of much of the office personal applications software (word processors, drawing packages, spreadsheets, diary and time organisers, etc) inexorably followed this commodity market. Multiple language variants of popular software were developed at reasonable cost to cater for world-wide markets.
- The introduction of Windows, Icons, Mouse and Pointer ("WIMP") technology in the user interface greatly improved ease-of-use for unskilled staff and provided some degree of applications integration through the use of *cut and paste* techniques. It also laid the foundations

for manipulation of graphical and image information alongside text and tabular data.

- The introduction of Local Area Networks offered dramatic opportunities for improved communications between PCs and also with other IT services, providing a key foundation for the move to the third generation office.

During the 1980s there was a steady growth in the use of PCs within the office, relatively slowly at first, but accelerating rapidly later in the decade and into the 1990s. This growth was driven essentially by the flexibility and independence gained by processing applications locally rather than through a comparatively remote departmental or corporate server. The benefit of local applications independence often proved a weakness at the wider enterprise level; similar, but incompatible, PC software applications established in different parts of an organisation could prove a significant barrier to co-operative working and information exchange.

Initially, integration of the PC into the wider IS environment was very limited; some improvement followed as terminal emulators and file transfer packages became more widely available, but the introduction of LAN-based networking software was a major advance. PC networked operating systems (NOSs) provided facilities for administering a local workgroup, sharing files and printers and transferring electronic messages between users, although they could not easily scale to compete with the established server-based products for application throughout an enterprise. In the meantime, the cost differential between the PC and dumb terminal had narrowed to the point at which the additional flexibility of the PC made it a more natural choice for the office workstation. It was these steps that marked the start of the transition to the third generation of office systems.

3.4 *Third Generation: Client-Server*

There was no clear boundary between the second and third generations of office systems, rather a period during which elements of both second generation technologies merged to provide the foundation for a client-server approach to office applications. (ICL can claim one of the earliest client-server office systems with OFFICE 20, a set of office facilities for the DRS 20 range, exploiting its distributed microprocessor architecture. Although this preceeded the industry-standard PC applications by several years, it embodied important concepts of the client-server office; personal applications such as word processing and diary were implemented on the workstation, networked to mail and document filing applications running on the DRS server.)

On the one hand, the established server-based office products were further developed to allow the PC to handle personal applications at the desktop; this was typified by products such as the OfficePower Client Object Integration option and DEC's Teamlinks for the All-in-One office system.

These developments enabled PC applications and associated object types to be integrated into the existing framework of server-based functions such as electronic mail, filing cabinets and document libraries. Document conversion software provided a means of exchanging information between different PC packages and existing host based applications.

On the other hand, the PC-based office products were themselves being extended to incorporate similar, shared, server-based facilities such as electronic mailboxes, document libraries and meeting schedulers; Nokia's Alfaskop Office (now TeamOFFICE) and Lotus' Notes and cc:Mail products, along with parts of the Microsoft Office suite, typified the move into server-based functionality.

Most PC-based office servers were tied initially to specific software platforms such as OS/2, which exploited tightly integrated PC-LAN networking environments as the basic client-server infrastructure. However as PC-LAN capabilities have been extended to cover platforms such as Unix and, more recently, Windows NT, the scope of these office products has been extended onto potentially very powerful server platforms. The significance of this should not be underestimated; it marks the consolidation of the PC- and Server-based approaches into a third generation client-server office architecture capable of scaling into the enterprise environment.

An important component of the cost of such systems is management and administration; a substantial part of this varies with the number of interconnections which have to be configured and maintained between the different office servers within the enterprise. The following Table provides a simple example of this for a sample enterprise population of 10 000 users.

Table 1 Inter-node connections for a 10 000 user office system

Average Users/Node (u)	Number of Nodes (n = 10000/u)	Internodal connections Flat Mesh $^nC_2$	Hierarchic $\geqslant n$
10	1000	499500	$\geqslant 1000$
100	100	4950	$\geqslant 100$
1000	10	45	$\geqslant 10$

Column 3 covers two scenarios, a worst case, in which each server is configured to communicate directly with all others in a flat mesh, and a best case, in which each server is constrained to communicate in a strict hierarchy. In practice, the requirement will also be influenced by the physical geography of the offices, the organisational structure of the enterprise and quality aspects such as network performance, resilience and back-up, so for most organisations the optimum solution will lie somewhere between the two. However, the table clearly illustrates how the use of relatively large servers substantially reduces the scale of systems management and administration at the enterprise level. (See also (Anderson, 1993) for a cost breakdown.)

A key aim of the third generation office system can be summarised as "right sizing" – selecting the most appropriate type of platform and software environment for each of the individual office applications, whilst retaining the capability for them to work as a cohesive, integrated set. Information capture, document preparation, manipulation and rendering tasks can make use of flexible software running on powerful, inexpensive PCs whilst integrating with shared document stores, electronic mail infrastructure and workgroup applications, which can exploit the richer functionality of the server environment. Many of the individual applications have evolved from the second generation (both PC and server) but they have now become more tightly integrated within this client-server framework.

4 The 3rd Generation Office Architecture

The main functional components of the 3rd generation office system can be grouped as shown in Figure 3.

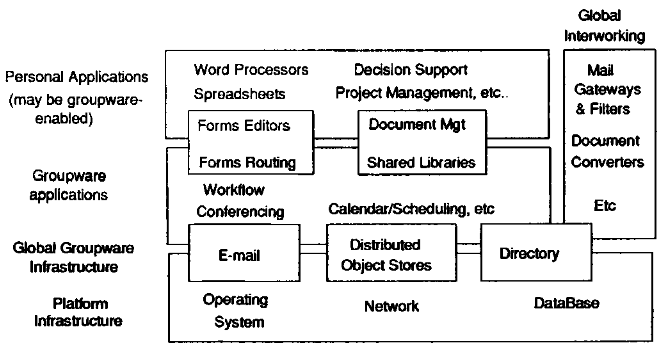

Fig. 3 Functional components of the 3rd generation OIS

4.1 Platform Infrastructure

The platform and network infrastructure provides the basic distributed computing framework on which the various office applications are built. To date this infrastructure has tended to provide distribution and transparency mechanisms local to the server and to its immediately connected population of clients – typically a logical PC-LAN group. Current products (for example based on Novell's Netware or Microsoft's LAN Manager and Windows for Workgroups) provide facilities such as file and printer sharing, procedure calls or message passing to support distributed applications, local email and a measure of workgroup administration. Across an enterprise, there will often be many such local workgroups, possibly using different vendors' products. Wider cooperative working across an enterprise thus requires some common services which can operate globally across the different work-group communities; this is termed "global groupware infrastructure" (see 4.4). Although several vendors are extending individual platform infrastructure products towards global operation, an important characteristic of the groupware infrastructure is its ability to operate in a heterogeneous product environment.

4.2 Groupware Applications

This is one of the fastest developing areas of new office systems as business reorganisation and process changes generate increasing requirements for more effective group- and task-oriented office working. Conferencing applications (for example ICL's Team Forum) enable groups of users to co-operate on informal tasks such as brainstorming or formulation of ideas; workflow applications provide a means of defining and managing activities involving formal group interactions. Forms-based applications linked to forms-routing packages can replace paper-based processes such as expense claims processing or loan authorisation; these may be implemented within a department or across an organisation. Early groupware applications tended to be written as free-standing programs, but more recent applications have been designed to operate in conjunction with common services such as email, directory and shared object stores.

4.3 Groupware-enabled Personal Applications

Some personal applications can be linked to groupware services to support interchange to a defined group of users. For example, a spreadsheet may be linked to a document routing application so that it is automatically sent to contributors who may need to supply information and subsequently return it to the owner. Personal document management applications may be similarly linked to groupware library applications to store, search or retrieve documents shared by a group of users.

4.4 Global Groupware Infrastructure

Most groupware applications, for example conferencing, document libraries and workflow, need to operate in a more global environment than a local client-server workgroup. To support this global workgroup environment, three functions in particular have emerged as core building blocks; these are located on the boundary between the platform infrastructure and groupware applications and form a point of integration between them.

4.4.1. *Electronic Mail* Electronic messaging had already established itself as an important component of the 2nd generation office. One key to its success was the "store and forward" nature of its service, enabling effective communication between users irrespective of time zones and working patterns. Gateways to facsimile and physical delivery services extended its capabilities into related messaging services.

However, its real value goes far beyond simple transfer of messages:

- It can support interchange of a wide variety of information types as attachments – forms, word processing documents, spreadsheets, image data, audio or video clips, software programs, directory information, in fact virtually any type of object encountered within the office.
- It can support delivery to multiple recipients and distribution lists (normally maintained in a directory), providing an important foundation for information transfer between defined workgroups.
- The recent establishment of industry standard APIs to access electronic mail services has enabled the development of "mail-enabled" applications, to exploit email as a vehicle for distributing information between applications.

4.4.2. *Distributed Object Stores* Whilst electronic mail provides an excellent framework for delivering information individually to users located around an organisation, it does not enable users to share information directly. Electronic mail distributes multiple copies of information; effective sharing depends upon some higher level coordination mechanism to control and propagate changes. Providing remote access to shared multi-media information held as a single copy at a particular physical location within the office network is often impractical due to the physical characteristics of the network and the complexities of configuring the logical access paths.

The solution to this requirement which has emerged is a shared object store which is replicated, in whole or part, across a community of office servers, providing each user with a single image of the information, maintained at the local server. The object store may then be used to support groupware applications such as conferencing or shared document libraries (for example as in ICL's Team Forum and Lotus' Notes products). Important characteristics of such products are:

- A directory (which itself may be distributed using replication) is used to maintain global information about users, their access permissions and server locations.
- Changes to the object store information are replicated across servers as necessary using email or other communications service.

- Data consistency is normally maintained by enforcing a single global permission to update and delete an object, whilst allowing multiple read and extend permissions; extensions are synchronised between servers on a time basis. The update permission may be dynamically reassigned between users, for example when checking out a document in a globally shared library for local updating by an individual user.

  Whilst not matching the levels of data integrity and consistency achieved by a 2-phase commit process, the replication approach has proved entirely adequate for most office applications.

4.3.3. *Directory* Electronic mail and distributed object stores both depend upon a global directory mechanism to maintain information about users' names and addresses, server location, access rights, membership of distribution lists, etc. Individual groupware applications also may need to maintain further user information such as interest in various conference topics, preferred document formats, passwords or application privileges, responsibilities within a workflow process, etc. As the scope of the electronic office has grown, so has the importance of a directory service to administer and co-ordinate information about users and services.

As there is substantial commonality in the user information needed by the various office applications, there are significant benefits in administration and management by locating it in a general purpose directory accessible to all such applications (and extensible by new applications). For administrative convenience, the directory needs to be distributed in all except the most simple office networks. The X.500 standard models the directory in object-oriented terms with attributes grouped into class hierarchies; it also contains provision for the replication of directory information. Whilst many office products include a directory following these principles, very few directories are capable of interchanging information with other products or provide open APIs for application access.

4.5 *Interchange gateways and converters*

The growth in office applications, particularly in the PC area, has led to many different, often incompatible formats and interfaces in areas such as mail interchange, revisable and compound document standards, directory information and groupware applications such as workflow. Although there is now a reasonably consistent architectural framework within the industry, insufficient standards have yet been established to enable easy integration of products and information interchange. To overcome some of these limitations, specialist vendors have developed email/directory gateways and document conversion products; recent products in this area support interworking between different groupware applications such as diary and meeting schedulers. At the same time various industry initiatives have been launched to define standards in areas such as mail and directory APIs, workflow data interchange and multi-media access and interchange.

5 Towards the 4th Generation

Although the basic drives towards technological advance and business change will continue, it is doubtful whether we shall see a significant shift away from the client-server office in this decade. The architecture is still comparatively young and promises sufficient flexibility to encompass most of the emerging technology trends, some of which are summarised below. The most likely characteristics of the 4th generation office are an increasing adoption of object-orientated technology, which underpins a number of the newer concepts in office information systems, and a closer link between office applications and business processes. These characteristics can be accommodated within a client-server framework, so continued evolution in this direction is more likely than a radical shift in technology.

5.1 *Mobility*

Advances in mobile data communications, coupled with laptop and handheld PC technology, are extending the office beyond its traditional physical boundaries. Many of these changes will be contained within the network infrastructure and thus will be relatively transparent to office applications, although there is likely to be a need to distribute information to such devices for use in a "nomadic" mode, off-line from the main office environment for intermittent periods. Object services for data replication, consistency and synchronisation are being discussed within organisations such as the Object Management Group and Unix International and could provide the foundation for such usage.

5.2 *Integration of Telephony*

For years the promise of closer integration between telephony and other office services, such as email and directories, has remained substantially unfulfilled, despite the occasional innovative product such as the ICL One-per-Desk and specific industry applications such as telesales. However, substantial progress has recently been made in standards for exchanging telephony control information with conventional office computer systems. These will enable, for example, computer analysis and redirection of incoming telephone calls or outgoing call control from computer applications via a graphical PC interface. Several vendors are working on developments leading towards common directories for all information services and facilities to integrate voice mailbox systems and electronic mail behind a single user access interface.

5.3 *Multi-media and Videoconferencing*

In the past multi-media has meant essentially static information; in the future it will be increasingly "dynamic" (video, sound, graphical animation, etc). Desktop videoconferencing and real-time news feed services channelled to the desk (for example broadcasting financial or other high value information) are expected to become more common. Initially they will make use of specialist communications services such as ISDN, but over time will integrate with the wider office LAN infrastructure. In architectural terms they will require software support for new types of office object which will be linked to specific hardware assisted applications.

5.4 *Distributed Object Management*

The distributed object store will develop so as to provide true location transparency to the accessing application and the scope of object relationships such as containment, linking and embedding will be extended across different platforms into basic infrastructure. The Object Management Group, with industry support, are working on an Open Object Management Architecture (OMG, 1990) and related technology proposals which are relevant to this area. From the user perspective an important result will be a single consistent image of his local (PC-based) object store and those global or public objects accessible via the server(s). However, this area will see several different vendors contesting the market, so it is possible that different de-facto standards will prevail in the market for some time, despite industry work on common standards.

5.5 *Information filtering and intelligent searching*

Ongoing increases in processor, storage and communications capacities will render huge amounts of information, much of it unstructured, accessible to the office worker. Intelligent filters and knowledge-based searching of mail and document stores will increasingly be demanded to analyse, extract, summarise and convert information to the form required for use by the office worker. Benjamin and Blunt (1992) introduced the concept of the *Knowbot* ie:

> 'a program designed to travel through a network, inspecting and understanding similar kinds of information irrespective of its language or form.'

Standards for object request brokers and common object services will (eventually) enable the construction of this type of application.

5.6 *Task and Process integration*

Many of the current generation of office systems have relatively weak links with business processes; activities within the business process may use office automation in various areas (for example email to move information between individuals) but the business process remains essentially outside the IT system and non-automated. (This contrasts with, for example, TP systems, where the application typically reflects a business process.) This problem is being attacked from two directions. Bottom-up, applications such as forms routing and simple workflow are automating more of the existing process; at the other extreme, sophisticated process management applications are being introduced via business re-engineering. Both are expected to advance significantly in the coming years. At the personal level, task management software will provide far better day-to-day organisation of the individual's time with deadline reminders, integration with mail trays and/or workflow applications to filter priority items and plan working time, etc. As office applications become more tightly integrated with core business functions they will become more mission-critical and increasingly demand backup, recovery and failure-containment capabilities previously associated with mainframe applications; these functions will themselves require to be integrated into the operational office processes.

5.7 Standards

Much of the promise of the client-server architecture can only be realised in practice by adopting industry standards for product integration and information exchange. Recent work on open APIs to access email and directory services shows promise and many individual products have established de facto standards, for example the Microsoft Object Linking and Embedding (OLE) specifications, the Lotus Notes API, etc. Industry work on standards for object technologies and multimedia is well established and work has recently started on standards for workflow applications. There is thus the prospect that standards will be established for some important parts of the architecture; however, many areas are likely to remain where lack of standards, or slow progress on their definition, will result in incompatible products, inhibiting the establishment of a global business network.

6 Summary

The office information system has always been, and remains, very much a moving target, whose major innovations have been determined by commodity pricing and achievement of critical mass in usage. Ongoing developments have emphasised ease of use, support for the unstructured, multimedia information prevalent in the office and improvements to the basic office tasks of communications, cooperative working and activity coordination.

This paper has traced the development of the electronic office system through three loosely defined generations:

- the earliest systems based on standalone word processing
- the divergent developments into host- and PC-based offerings during the second generation

- the merging of the two strands into a single client-server approach, building on the strengths of the PC as the desktop device and the server as the vehicle for shared data and applications.

An overall evolution towards the 4th generation office system may be summarised in the following figure:

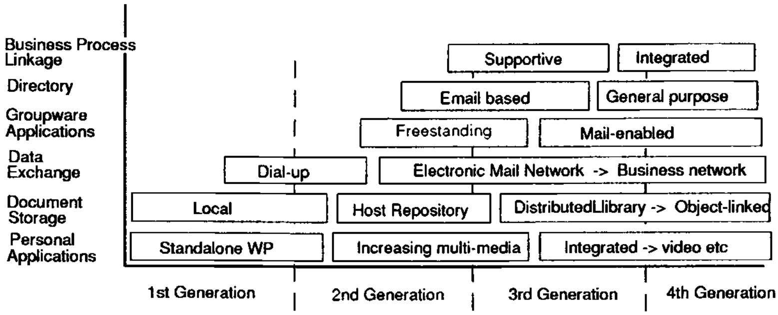

Fig. 4 Evolution towards the 4th generation office

The PC has emerged as a dominant component of the office system, supporting a wide range of local applications and providing the platform for client access to shared applications and information. The client-server architecture of today's third generation office systems has blended PC-LAN technology at the local workgroup level with host-based information management and distribution services to provide enterprise-wide capability. Three functions are of particular significance in delivering this global capability – electronic mail, distributed object stores and directory; these form the foundation for a wide range of emerging groupware applications to improve cooperative working.

Establishing a cohesive infrastructure in these areas will be an important step towards an effective electronic business network. For many organisations this will involve decisions on the use of gateways and document conversion to support interworking between the wide range of existing (and often incompatible) office applications.

Progress towards the 4th generation will continue on many fronts, retaining concepts from earlier generations as the foundation for adding new functionality in areas such as dynamic multi-media information and the integration of telephony and videoconferencing. The architecture of the client-server office is expected to endure; it will be enhanced to incorporate more use of object technologies and the increasing integration of business processes as the development of the business network reflects the 'revolutionary' (Scott Morton) phases in the application of IT within the enterprise. However, the fundamental role of the office system remains to support and empower its user community and, ultimately, progress can only be as fast as their rate of adaptation to and exploitation of new technology.

Acknowledgements

Many colleagues within the OfficePower Group and Client-Server Systems have contributed ideas and stimulated thoughts on this subject; I thank them all.

References

ANDERSON, M. The Groupware Phenomenon, *Gartner Group 12th OIS Conference proceedings*, 1993.

BENJAMIN, R.I. and BLUNT, J. Critical IT Issues: The Next Ten Years, *Sloan Management Review*, pp. 7–19, Summer 1992.

CAMPBELL-GRANT, I.R. and SMETHURST, C.R. Multimedia and Standards for Open Information, *ICL Tech. J.* 8(3) pp. 453–472, 1993.

EDMONDS, E.A. The emergence of the separable user interface, *ICL Tech. J.* 7(1) pp. 54–65, 1990.

FULLER, A. Future Applications of ISDN to Information Technology, *ICL Tech. J.* 7(3) pp. 501–511, 1991.

HOWARD, M. The Bottom Up OIS Market: A Difficult Balancing Act for IT Managers, *Gartner Group OIS Conference proceedings*, 1992.

HUTT, A.T.F. and FLOWER, F. The User-System Interface – a challenge for application users and application developers?, *ICL Tech. J.* 7(1) pp. 43–53, 1990.

OMG, *Object Management Architecture Guide*, available from the Object Management Group, 1990.

MOORE, B.J. Making a Secure Office System, *ICL Tech. J.* 7(4) pp. 801–815, 1991.

RINGLAND, G. Standards and Office Information, *ICL Tech. J.* 5(4) pp. 713–728, 1987.

SCOTT-MORTON, M. *The Corporation in the 1990s*, OUP, ISBN 0 19 506358 9, 1991.

VERRIJN-STUART, A.A. A further analysis of Office Systems problems, (IFIP TC8/WG8.4). *Office Knowledge: Representation, Management & Utilization* (Ed: Lamersdorf.), ISBN 0 444704515, 1988.

Biography

David Hollingsworth

David Hollingsworth graduated from the LSE with an honours degree in Economics and joined ICL in 1968. His early career was spent working on the development of TP software and its application in various customer projects. This was followed by several years developing sizing and performance methodologies for use in customer projects involving early 2900 series machines.

Following a move into product marketing he became responsible for networking requirements and was extensively involved in the development of ICL's networking architecture and planning its strategic direction towards open systems. He contributed to the formulation of OPEN*framework* as company networking architect and subsequently moved into the OfficePower group to take responsibility for its future office systems architecture.

IPCS - Integrated Product Configuring Service[1]

C. W. Bartlett
Independent Consultant, Manchester, UK

Abstract

During the 1991-2, GPTs Network Systems Group (GPT-NSG), in conjunction with ICL, installed a suite of knowledge-based configuring systems for use by Technical Support staff in the initial response to an invitation to Tender (ITT) and in the provision of detailed manufacturing instructions for those orders that are taken.

GPT-NSG is a large UK supplier of equipment for telecommunications networks. The use of the suite of configurers significantly improved both the response time for ITTs and the accuracy of the quoted equipment and prices.

A fairly high initial cost, recovered in the first year of operation, on tools and methodology led to a situation where the service can be expanded cheaply and quickly.

The case study presented here demonstrates that the knowledge-based technology involved in the production of configurers has matured to the point where large scale projects can be commissioned with a high degree of confidence of success.

1 Introduction

GPT-NSG is a major international supplier of a wide range of telecommunications equipment. Their market ranges from the supply of individual items of equipment, customised and/or fitted to the customer's specification, to the supply of complete, often nation-wide, telecommunications systems.

At both extremes, GPT-NSG has the problem of first responding quickly and accurately to enquiries for such equipment and, following the taking of

---

[1]A previous version of this paper was submitted to the 1992 competition for the D.T.I. Manufacturing Intelligence Award and received third prize.

such an order, ensuring that the delivered equipment is complete, mutually compatible and meets the customer's requirements.

ICL's Decision Support Product Centre was responsible for the initial development of ICL's own *product configuring system* [BARTLETT 1987] and in the time following the deployment of that system has designed and constructed a number of such configuring systems for other organisations.

A configuring system translates a high level description of a functional requirement into a low level listing of all the hardware and software that must be provided to meet that requirement.

Subsidiary to this main function are such additional optional features as costing the recommended system, generation of reliability predictions, spares requirements, documentation requirements and physical layout information. Configuring systems are a classic [McDERMOTT 1982] application of KBS techniques. They represent a cost-effective solution to a problem that is often impossible to achieve with conventional programming techniques – due to the rapid rate of change of the requirement compared with the implementation time.

2 The Problem being Addressed

Early in 1990, GPT-NSG had come to realise that it could improve Quality in respect of its handling of ITTs and any subsequent orders. This room for improvement manifested as difficulties in:-

- responding within the timescale of the ITT;
- producing an accurate, consistent response;
- producing accurate forward information for manufacturing;
- ensuring that equipment delivered to a customer was mutually compatible, complete but only complete, and conformed to the customer's requirements.

At that time, the responses were handled by largely manual systems based on paper data sheets. Due to the breadth and complexity of GPT-NSG's product range it was not possible for individual technical support people to be fully conversant with the technical details of every item in the range. As a result, individual technical support people had developed individual areas of expertise. The consequences of this were inevitable bottle-necks and delays and, because of the work load, an inability to investigate alternative solutions to a requirement. These problems became very much magnified on those occasions when GPT-NSG was assembling a quotation for a very large communications network involving many elements of their product range.

In November 1990, GPT-NSG contacted ICL and invited them to collaborate in the preparation of a joint report. This was presented to the Board of GPT-NSG in May 1991 and included:-

- A working Concept model, running on a standard PC, that demonstrated how knowledge-based configuring could be applied to the Automatic Cross-connection Equipment.
- A Cost/Benefit model showing the massive savings in operating costs that could be achieved by the wide-scale deployment of Configuring Systems. (The main benefit shown in this model was the avoidance of losses due to late invoicing to the customer.).
- The identification of a requirement for a front end, a Product Selection System, to the suite of Configurers that would match the requirements of large scale end-users of telecommunications equipment to the types of equipment GPT-NSG was able to deliver with due regard to commercial constraints and marketing policy.
- A proposal showing how GPT-NSG should be organised to control the development and deployment of the various configurers and how the technology and skills involved in the creation of configurers could be transferred to GPT-NSG staff.

This report was accepted by the Board of GPT-NSG and a program of work agreed with ICL. This comprised:-

- The delivery of configurers, complete with technical documentation and User Guides, for four of GPT-NSG's products:
  - A large (up to 512 ports) Automatic Cross-connection Equipment (ACE) for digital networks. ACE is used to automate the (many to many) connection of 2Mbit/s digital channels in on-going real time situations;
  - A range of compact primary multiplexors which accept a variety of analogue, voice and data traffic inputs and multiplex them into an aggregate 2Mbit/s channel;
  - A range of modular multiplexors with capabilities for multiplexing traffic from 2Mbit/s through various levels up to 140Mbit/s;
  - A range of multiplexors providing a Synchronous Digital Hierarchy.
- Development of a generic system for the maintenance and further development of the suite of configurers;
- A comprehensive report on the requirements for the Product Selection System;
- A proposal for a configuring system aimed at the software and hardware required for the management and control of large scale networks of GPT-NSG's products;
- A technology transfer programme for GPT-NSG's staff, this being rated as of an importance equal to that of the various tangibles.

The whole of this package was designated GPT-NSG's Integrated Product Configuring Service (IPCS).

Emphasis was placed on the development of the first four configurers. Analysis of GPT-NSG's business plans had revealed that these were where the most immediate improvements in cash flow could be expected. This was so even if they were deployed as stand-alone systems without being fully integrated into the rest of the IT support available to GPT-NSG staff.

Subsequent to ICL's involvement in this initial package, GPT-NSG has gone on to develop seven further configurers for other members of its product range. All of these are in productive use, and there is considerable demand from users for more configurers to complete the coverage of the full catalogue.

3 How the Problem was Solved

3.1 Development of IPCS

3.1.1 Delivery Requirements. Examination of the delivery possibilities for IPCS revealed a choice between existing UNIX-based workstations and networked PCs. Costing the alternatives showed an overwhelming case for using PCs.

The user preference was for a Windows-style application. This put an immediate boundary on the choice of tools. Some, fairly limited, comparative trials quickly showed Kappa PC[2] as possessing the desired richness of facilities coupled with a good operational speed. It also allows applications to be ported from the PC onto a UNIX system running ProKappa. This is expected to be of importance during the later stages of the project when the suite of configurers is brought together in the Product Selection System.

A further factor to be imposed was that of ensuring that output from the configurers is consistent with the input requirements of GPT-NSG's MRPII system (used to manage the flow of orders through the manufacturing process), thereby allowing quotations to be progressed right through to factory output without any need for re-inputting of data.

3.1.2 Required Functionality. The primary function of configurers is the generation of a listing of all the parts required to satisfy the requirement for equipment. Beyond this, other areas of functionality are required to support the response to an ITT and eventual supply of the equipment:-

- *Case management* – allowing the user to save the details of a particular enquiry and later retrieve it for possible amendment or re-evaluation against a changed set of configuring rules.

---

[2]Kappa – PC and ProKappa are products of IntelliCorp Inc. They are both state of the art hybrid development systems combining Object Oriented techniques with the inferencing techniques of classical KBS development systems.

- *Adaptive Engineering* – there are often cases where the requirement cannot be matched completely from existing parts. The configurer is required to recognise these situations and allow the user the opportunity to raise special engineering actions to cause the part to become available. These requirements and the user's justification are attached to the final report.
- *Documentation* – there are two aspects to this. Listings of relevant specifications are required for attachment to the contract. With some of GPT-NSG's products the user documentation is generated by a "pick and mix" process. Guidance is required as to what sections are to be included in this documentation supplied with the product.
- *Physical layout information* is often required. The exact requirements are dependent on the particular equipment. They include the generation of a floor "footprint", the generation of a CAD driving file for a detailed schematic of the layout of the equipment, and diagrams showing the positioning of Circuit Cards in Shelves.
- *Reliability predictions* are required for the total equipment and for the availability of an individual channel.
- *Power dissipation predictions* are required.
- The *overall mass of the configured equipment* is required.
- (A *costing* is required for the configured equipment. This is generated outside the configuring system by correlating the output with a cost/price database.)

The *configurer building and maintenance system* (CBAM) is a fundamental component of IPCS. It is the prime means of ensuring that the configurers present a common "look and feel". It also supports a common structure for all configurers and maximises the re-use of code. Apart from this, CBAM has a major role in maintaining the quality of the individual configurers. Each configurer is held as a set of three files:-

- The *drivers file*, covering the acquisition of the user requirement, and hence the Human Machine Interface, and containing the common structures and functionality required by all configurers;
- The *parts file*, containing the details of the parts used by a particular product and the functionality concerned with the use of the rules for configuring these parts; and,
- The *rules file* containing the various rules concerned with configuring the product.

The division of a configurer into these three source files has the benefit that the construction and maintenance system is able to perform a set of consistency checks (at the lexical level) on the compatibility of the rules and parts with respect to coverage, duplication and inappropriate matching of rules and parts. This considerably raises confidence in the correct operation of the configurer and significantly reduces the need for difficult and lengthy checking at run time.

3.1.3 Management Aspects. The initial development phase was conducted while GPT-NSG was undertaking a radical restructuring program. As a consequence, certain areas of expertise in configuring were expected to become unavailable within the duration of this initial phase. This had a direct effect on planning the project. It would have been desirable to construct a set of small prototypes for the first products, and then design a common structure and follow this by the design and construction of the construction and maintenance tool.

However, because of this expected skill shortage, it was necessary to plan the full scale development of the first configurer (that for the ACE equipment) as the first delivery. This led to a fairly difficult technical management problem with overlapped design and implementation of the configurers and their construction tools. The object oriented features of Kappa PC were of great benefit during this time, greatly facilitating the merging of the code for the various items.

Another aspect requiring careful handling was that of the management of user's expectations. The initial investigations into the requirements for and possibilities of a configuring system had generated considerable interest in the potential users. This tended to manifest itself as a "too little too late" attitude. The regular seeking of comment by means of working, interim releases did much to alleviate this.

The *production phase* was marked by a change in the management regime from project to line. Ownership of the various activities and artefacts is divided between:-

- Engineering who own the rules;
- Technical Support who own the configurers; and,
- Information Systems who own the tools and the physical means of delivery.

The strategy governing this production phase was one of maximum gain for minimum initial effort. Thus initially basic facility (parts listing only) configurers were developed for the whole product range and only when this was completed were they revisited and brought to full functionality.

3.2 Methodology and Techniques

3.2.1 Role of the System Developer. During the initial phase of the project, the lead was taken by ICL. In this role, ICL was responsible for:-

- The design and construction of the development and maintenance tools; and,
- The generic and detailed design of the four configurers; the elicitation of the knowledge required for these configurers; and their implementation.

To ensure that there was an effective transfer of the technology, a GPT-NSG person was assigned to work full time with ICL, and was responsible for one of the configurers. This transfer was further facilitated by making all work-notes as well as specifications freely available to GPT-NSG.

Valuable feedback was obtained from GPT-NSG during this phase by seeking comment on a series of increments of all items. This incremental approach was strengthened by the agreement of a number of formal delivery/payment points identified throughout the duration of the project.

Beyond the delivery of an item, ICL provided a 28 day warranty. During this period any bugs found were attended to urgently as a joint exercise between ICL and GPT-NSG. This gave a valuable boost to the transfer of the technology to GPT-NSG. Beyond these 28 days, ICL was willing to supply general support but there was little need for this.

3.2.2 Writing the Reports. In the production of the two reports ICL made full use of the *concept to definition* stages of the formal *marketing to design*[3] methodology [HUTT et al 1990]. In recording and analysing the results of the numerous interviews with GPT-NSG staff, extensive use was made of KANT[4] [STORRS et al 1989]. As a result of this, a line is traceable between all features identified in the reports and the various interviewees who provided input bearing on those features.

3.2.3 The Configurers. Superficially, configuring equipment would appear to be completely deterministic. In practice, with products having the generality and flexibility of GPT-NSG range, this is far from the case. In general, many configurations of a product can meet one user requirement. To resolve this flexibility, it is necessary to consider other factors. For example, marketing preferences and manufacturing constraints, and the need to be compatible with other equipment

---

[3]This is a formal methodology based on the results of three Alvey projects: "User Skills and Task Match Methodology" (ALV/MMI/PRJ/143); "Development of Methods for Early Evaluation of Interface Designs" (ALV/MMI/PRJ/122); and, "Human Interface Monitoring System" (ALV/MMI/PRJ/091).

[4]KANT is a Hypertext-based system that provides many powerful facilities for collating large amounts of, informal, knowledge arising from a multiplicity of sources. Of particular interest are its facilities for keeping records of the provenance of any derived knowledge and its ability to deduce the implications of any changes in the source material.
KANT was developed as part of the "DHSS Large Demonstrator" Alvey project.

being configured for, or already present at, the target *station*[5]. A further consideration is that the process followed by the configuring system should match that followed by an expert Support person doing the same task. Hence, as well as using technical documentation for the various products it was necessary to conduct extensive interviews with the Support and Product experts. The methodology used for these interactions was the *elicitation/teachback interview* [JOHNSON et al, 1987].

In the implementation of the configurers full use was made of the object oriented features of Kappa PC. Thus the structure of a product was analysed in terms of the grouping of the components into broad areas of functionality. This grouping was directly reflected in the object hierarchy. Associated with each of these functional areas was a set of (both forward and backward chaining) rules, these being divided into watertight compartments and executed in suitable sequences.

Report generation was by context-free methods assigned to the Root class of the Part hierarchy. Similar methods are provided for mass, reliability predictions, and power consumption.

The availability of object oriented technology was particularly valuable for generating layout information. Thus, each type of part was invested with the (procedural) knowledge of how to assemble itself into a higher order sub-assembly. For example, a card knows that it must fit itself into a particular type of shelf. Using this approach it is possible to easily mimic the physical construction by sending a series of messages to the various parts. Thus, for example, cards are told to go into shelves, shelves to go into racks and, racks and cable ducts to assemble together to form a station. As a consequence of this approach the area of the configurers concerned with physical layout is exceptionally easy to understand – even though it is all procedural coding.

3.3 Software, Timescale and Cost

Using Kappa PC, from IntelliCorp Inc., the development of the generic construction and maintenance system and the initial batch of configurers took 9 months. Subsequent to this, about 3 weeks are now required for the initial release of a new configurer (with basic functionality).

The cost of the initial phase (reports, construction system and four configurers) was £350,000. This initial investment allows further, basic parts identification, configurers to be produced at an average cost of £5,000. (Following coverage of the whole product range at this basic level the

---

[5]In telecommunications, a station is a physical location housing a multiplicity of telecommunications equipment.

configurers are brought up to the full level of functionality at an average cost of further £5,000 each.)

3.4 Key KBS Aspects of IPCS

- The Knowledge was *shared* [WIIG, 1990], i.e. it is specialised, its concepts were understood by experts with insights into difficult problems and who used expert strategies in providing solutions.
- The system was rule-based, the rules being separated from the inference engine. These rules encoded the expert knowledge of how to configure the product. The way in which the experts used this knowledge was implicitly modelled by the sequence and manner (forward/backward) that the rules were utilised.
- Knowledge about the product to be configured was coded using the object-oriented facilities of Kappa PC.
- The system recognised novel situations and responded by allowing the user to request special engineering action to satisfy the requirement.
- The system was easy to maintain. Not only were the rules separated from the inference engine, they were held in separate source files. This is also true with respect to information concerning the structure of the equipment and the processes used to configure it.
- The problem was too large and individually too complex to tackle economically by conventional means.
- Provision of a Graphical User Interface meant the configurers were very easy to use and required only minimal written documentation.
- Whilst the configurers did not offer explanation facilities to the user they did provide extensive context-sensitive help. This was accessed by using the right mouse button on any object visible to the user.

4 The System in Use

The first configurer, that for ACE, went operational in October 1991. By April 1992 a total of four configurers were in use. By the end of September 1992 there were a total of eleven configurers in daily use by Technical Support.

The system is mounted on a PC network. Access to the individual configurers is controlled via a simple front end that is used to replace the standard Windows PROGMAN.EXE. This replacement provides the required security of access.

By the end of 1992 there were some 35 uses of the configurers per working day.

5 Direct Benefits

5.1 Cash Savings

The expected savings per annum are shown below. This shows that, as planned, the project broke even in its first year. This was an important target that had to be agreed before permission to proceed could be obtained from GPT-NSG's Board.

Note that the savings are expected to increase year by year. There are two factors at work here, the increasing coverage and facilities of IPCS and the growth of GPT-NSG's business.

1991	1992	1993	1994	1995	1996	Total
£k0	£k281	£k428	£k538	£k592	£k651	£k2,490

The factors taken into account in formulating this prediction are:-

- The costs of introducing, maintaining and enhancing IPCS.
- The effort saved in configuration preparation.
- The effort saved in installation and commissioning of the delivered product.
- Avoidance of losses due to acceptance of under quotations.
- Avoidance of loss of business due to over quotations.
- Profit from new business arising from IPCS.
- Savings due to shorter cash recovery times.
- Savings due to accurate and timely procurement predictions.
- Savings due to reduction in Adaptive Engineering.

Of these factors the major benefit arises from shorter cash recovery times due to being able to invoice earlier, since the equipment delivered to site is correct first time and every time. (In fact, in the cost/benefit model used, an optimistic view of the cost of money was employed – the interest rates assumed were probably too low for the foreseeable future. Hence the savings are likely to be somewhat greater than those indicated above.)

5.2 Staff Savings

Introduction of IPCS has led to immediate gains in individual productivity. During 1992, the number of people employed in Technical Support was reduced from 25 to 11 whilst the work throughput actually increased.

5.3 *Technology Transfer*

Apart from the planned and achieved levels of cash savings the other important objective of the project was the introduction of KBS technology into GPT-NSG's culture. Whilst there is still a long way to go, it is already obvious that this has been a success, as evidenced by the speed and low cost with which new configurers are being produced. GPT-NSG is now in an excellent position to introduce KBS solutions into its development and manufacturing systems.

6 Indirect Benefits

6.1 *Lessons Learned*

There are a number of lessons of general import:-

- The technologies used and required for the low risk, predictable production of configuring systems are available and well understood;
- There are considerable benefits in the use of hybrid object-oriented techniques;
- For telecommunications systems, care is required in closely defining the boundary of the domain for each configurer. Without this care, there is a tendency for configurers to "grow" uncontrollably;
- KBS development systems which support the generation of plain-language source-files allow lexical level checks to be applied to the consistency of the Knowledge Base; this brings considerable benefits;
- The formal *marketing to design methodology* is of great benefit when capturing the details of complex requirements;
- When developing configuring systems it is possible to convince engineering staff of the benefits and effectiveness of knowledge-based configuring systems and to educate them to deliver configuring information in (near) rule format;
- The large-scale introduction of KBS systems into the normal working systems of a large company both requires and induces major cultural changes. With careful management, particularly with regard to expectations, these changes can be beneficial in areas other than those directly effected by the KBS systems.

6.2 *Applicability across UK industry*

Any supplier of equipment constructed from standard parts to meet varying and largely unpredictable customer requirements would benefit from use of configuring systems such as are described in this paper. The gains primarily accrue from the advancement of the invoicing date made possible by the equipment delivered being right first time and every time. Break-even can be achieved in the first year of operation.

It is to be expected that such suppliers would get a good initial return from individual configurers whilst integration of a full suite of configurers linked by an intelligent Product Selection System would be expected to give significant strategic gains in respect of their ability to respond quickly and accurately to large requirements involving a mixture of their products.

7 Future Plans for the Application

Following the strategy of going for the biggest immediate benefit, the initial push was to get configurers in place for all product ranges. This involved writing a further 7 configurers and bringing the existing configurers up to full facility. Following this, configurers were constructed to enable the configuring of racks containing multiple product types. This will be followed by a station configurer where the user will be able to specify the physical disposition of the equipment within a station and be provided with schedules of cables, cable ducts and other installation materials and equipment.

Alongside this expansion, the system will be moved from its current loose coupling with the MRPII system to a situation where it is closely integrated – to a point where it will not be possible to enter orders that have not been generated by a configurer.

Beyond this, GPT-NSG aims to get into a position where the configurer is available at the same time as the product. An important side-effect of this will be that GPT-NSG's products will then be designed with configuring in mind.

The ultimate aim is to integrate access to the configurers into the strategic product selection system. This will raise the level of dialogue to that of the more abstract expressions of a customer requirement. For example, a requirement described as a cable TV system covering a designated area, rather than as a system capable of distributing signals of a designated character from a set of designated points by means of designated types of equipment. By this time the configurer suite will have become an element in a total marketing and sales environment.

Acknowledgements

As is usual with projects of the complexity of the work described in this paper, many people made a critical contribution. In particular, acknowledgements are due at GPT to Liz Dodds for championing the whole idea of the IPCS during a difficult period of re-adjustment and to Wlodek Rutkowski for sympathetic Project Management.

Within the Decision Support Product Centre, mention is particularly due to Richard Townsend who as well as acting as Business Manager actually coded two of the initial deliveries himself. Additionally, Mark Reeder deserves credit for the success of the CBAM concept which was implemented

against a background of changing requirements and non-stable tools. Mick Smith was responsible for the difficult task of preparing the various reports.

References

BARTLETT, C.W. "S39XC – a configurer for series 39 mainframe systems", *ICLTech. J.* Vol. 5, Iss. 3, pp. 385–403, May 1987.

HUTT, A.T.F., FLOWER, F. "The development of Marketing to Design: the incorporation of human factors into product specification and design" *ICLTech. J.* Vol. 7, Iss. 2, pp. 253–269, November 1990.

JOHNSON, L., JOHNSON, N.E. "Knowledge elicitation involving teachback interviewing" In: "*Knowledge acquisition for expert systems: a practical handbook*", ed. KIDD, A.L. New York: Plenum Press, 1987.

McDERMOTT, J. "R1: a rule-based configurer of computer systems", *Artificial Intelligence*, 19 (1) 1982, pp. 39–88.

STORRS, G.E., BURTON, C.P. "KANT – a knowledge analysis tool", *ICLTech. J.* Vol. 6, Iss. 3, pp. 572–584, May 1989.

WIIG, K. "Expert systems – A manager's guide", (p. 148) Management Development Series No. 28 International Labour Office, Geneva 1990.

Biography

C. W. Bartlett

Clive Bartlett has spent over 30 years in the computing industry. In 1982, he became a founder member of ICL's new Knowledge Engineering Group (then only four strong). Since then, apart from writing the first ICL Configurer, he has worked on a wide range of KBS topics at all levels.

The Knowledge Engineering Group became, by the time of the project described, the Decision Support Product Centre. As a member of this team he was responsible for the overall design of the Integrated Product Configuring System.

He left ICL in December 1992 and is now giving independent consultancy as well as pursuing his interest in antique pressed moulded glass.

CGS - The ICL Configurer Graphics Service

S. M. Sharman
ICL Local Government and Health IT Unit, Bracknell

Abstract

Since 1986, ICL staff have been able to use CONFIGURER to aid them in producing hardware sales proposals. The system allows a user to specify exactly what components are needed to satisfy a particular customer requirement by way of an interactive question and answer session. It then applies a set of complex rules to this specification in order to satisfy two particular goals - firstly to determine whether the configuration as entered is technically correct, and secondly to automatically generate the list of ancillary items such as cables, which are required to ensure that hardware packages delivered to customers are complete. This paper is concerned with a relatively recent addition to this system, known as the Configurer Graphics Service. Described herein are the origins of the Service, its development history, the functionality it provides, and the techniques employed to provide this functionality.

1 Introduction

The general introduction in July 1986 of the CONFIGURER service (then called S39XC, as described in [Bartlett, 1987], marked a sea-change in the manner in which the ICL sales force approached the task of specifying system configurations for customers. By providing a tool for the mechanisation of the configuring process, a number of the problems traditionally associated with the configuring of computer systems are avoided. Such problems typically include overcoming the bottleneck associated with the shortage of staff skilled in system configuration, and ensuring the completeness and quality of configurations (as the supplier is likely to be held responsible for making good any inadequacies in the equipment supplied). In addition to overcoming these problems, the use of a mechanised tool allows the sales force to make great savings in time and effort (savings of up to 1600% are not atypical); perhaps even more importantly in the current climate, they are provided with the additional degree of flexibility required to be able to answer quickly and easily the kind of "what if" questions posed by a customer who requires a system capable of meeting business and technical requirements while satisfying economic constraints.

The range of configuring services offered by CONFIGURER has been greatly increased over the years, until at the time of writing the system may be used to produce configurations for the whole range of Corporate (Series 39) and Mid-Range (DRS 6000, DRS 3000) Systems products. CONFIGURER has also become the access and delivery vehicle for a whole range of associated value-added services, such as performance assessment of a wide range of ICL software products and distribution of up to date Product Descriptions for inclusion in customer proposals.

The CONFIGURER Graphics Service is one such innovation, and was developed in order to:

- reduce the amount of time spent by sales staff producing the diagrammatic representations of systems customary in sales proposals
- increase the quality and consistency of those diagrammatic representations, which typically were either hand-drawn or produced on some kind of desktop publishing package.

The need for such a tool was seen as pressing, both because of the greater demands on the time of the sales force as a consequence of the increasingly dynamic nature of the marketplace, and because the increasingly complex nature of systems made it more difficult for customers to visualise what they would receive. The key issue that the Configurer Graphics Service was intended to address was the customer need to see for what he was paying.

2 History

In late 1989 a member of the CONFIGURER team started work on a prototype graphics system. This was intended initially to produce line-diagrams of Series 39 configurations, and was written in PostScript [Adobe, 1990]. This proved the soundness of the basic concept and the desirability of developing a full solution, although it quickly became clear that to continue with an approach based purely on hand-coded PostScript would lead to a software package which would be extremely hard to update and maintain – thus making such a package unworkable for the highly dynamic ICL product range. It was clear also that successful development would, at least initially, require more resource than was currently available within the CONFIGURER team itself. Thus, at this time an approach was made to the Advanced Technology Group (ATG) of ICL, then a small development unit within Corporate Information Systems, specialising in development of systems based on advanced software technologies such as Artificial Intelligence (Winston, 1984).

A further period of prototyping followed, in order to find a technique which would produce output as good as or better than that produced by the

PostScript-based demonstrator, without compromising the maintenance and future development requirements of the system. This phase took about a month, during which a number of possible approaches were considered, using a rapid prototyping technique. The approach finally decided upon actually turned out to be a hybrid of a number of the approaches prototyped, thus proving the usefulness of this technique in development. The chosen technique integrated a bespoke knowledge-based system using a workstation-based PROLOG [Bratko, 1990] system, together with a desktop-publishing package known as FrameMaker, sourced from Frame Technology Incorporated, in the US.

Having decided upon the approach, development work on the system began around April 1990. By the end of September 1990, the initial version of the system, which produced high level overview diagrams of Series 39 configurations was available for testing by users in the CONFIGURER team. Following this trial, a number of small changes were made to the system in light of comments, and the system was finally made available for general use by all CONFIGURER users in mid-November 1990. Ongoing development work increased the functionality of the system, with deliverables emerging at regular intervals during the year following this general release. By early September 1991, CONFIGURER users could request and receive diagrams for Series 39 Overview Level, Series 39 Technical Vet Level (which shows in detail how the devices in a configuration are interconnected), DRS 3000 (including DRS M55/NX and M75/NX) and DRS 6000.

3 Functionality

3.1 Operation

CGS is invoked in response to a query at the end of a CONFIGURER session. The user is asked whether he or she would like to produce a diagram. If the answer is yes, then the user is given the option to have the resulting diagram delivered as a PostScript file via electronic mail.

The actions leading to the production and delivery of a diagram are as follows. Once the user has completed his or her interaction with the CONFIGURER, requesting a diagram in the process, the information describing that particular configuration (contained within an appropriate ADVISER (ICL, 1985) model) is extracted to a simple text file, formatted in the manner expected by CGS. Using the IPA (Information Processing Architecture) File Transfer Facility, this file is then transmitted from the mainframe upon which the CONFIGURER service is hosted (at the time of writing, the ‘K’ Service at Hitchin) to a Sun workstation in West Gorton, running CGS (see figure 1).

Files which are received by the Sun are processed in strict order, under the control of a cyclic UNIX process (known as *cgsbatch*) which operates continually. When started, this main control process checks for the presence

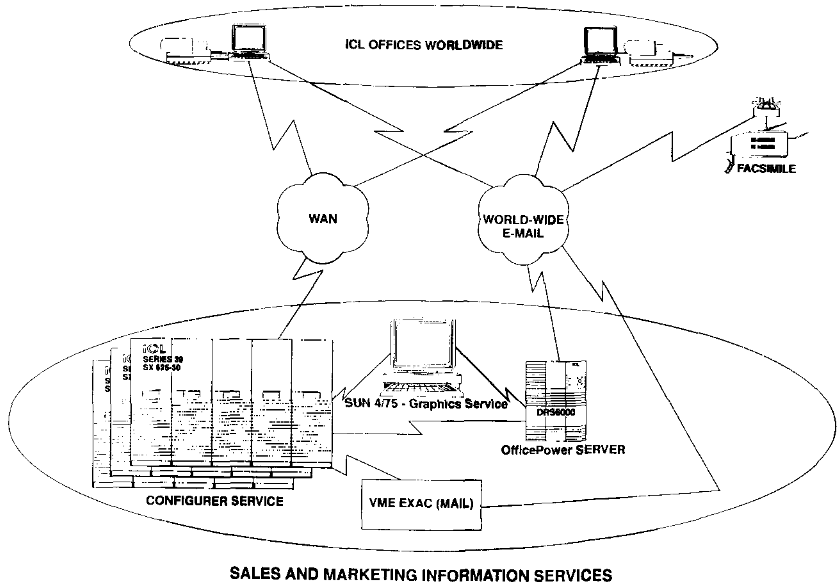

Fig. 1 Connectivity of CGS Systems

of files to be processed, and if none are waiting, returns to an idle state for a period of approximately five minutes. If however files are found, *cgsbatch* then calls further processes which actually generate diagrams, one at a time, for all those files awaiting processing. How each output file is processed depends upon flags set in the input file by CONFIGURER – each completed diagram is sent either to a local printer for printing and dispatch through the post, or alternatively transferred to a local DRS6000 running OFFICEPOWER and sent automatically to the requester via electronic mail. A report file, which may be referenced by the system operator if errors arise during processing, is also generated.

Depending upon the type of files processed, advisory mail messages may also be generated for dispatch via OFFICEPOWER – for example to inform the Series 39 Marketing Manager that there is a new SX prospect, so that he may then contact the appropriate account team to offer support. Finally, a database of previous users is checked, and if this check reveals that a previous user at the same site as the current requester has been receiving diagrams electronically, both names are flagged in a report to the CGS system operator. This allows a mail message to be sent to the current requester to the effect that other users on his or her site are currently requesting diagrams to be sent via email, and providing a contact name on that site.

Once all the requests in a particular batch have been processed, *cgsbatch* then restarts its cycle.

Many different types of diagram may be generated using CGS. They are:

3.2 Series 39 Overview

The S39 *Overview* software provides the customer with a high-level picture of the system which is being proposed. The other diagram types produced by the system are very much in the 'lines and boxes' style, whereas the individual objects on the page in S39 Overview pictures are drawn to resemble faithfully a front view of whatever physical device (eg. a node or disc unit) they represent.

The diagram represents the devices in the configuration by drawing a backdrop which shows the Macrolan and OSLAN networks present in the system as simple lines and superimposing the pictures or 'icons' representing the different node and peripheral devices on to these lines. This approach is illustrated by figure 2.

CGS incorporates a number of mechanisms to ensure that Overview diagrams remain as clear and uncluttered as possible. Scaling rules are incorporated, so that if a large number of devices need to be represented, the size of each individual icon in the picture is reduced in order to make the diagram fit on one side of A4. Similarly, if the configuration is a small one, then the pictures are made larger to reduce the amount of 'white space' on the page. In addition where more than one device of a particular type is configured, (for example multiple banks of a particular disc), labels are added to the diagram to indicate the number present.

3.3 Series 39 Technical Vet

The *Technical Vet* software is used to produce diagrams showing the Macrolan [Stevens, 1983] connections between the various parts of a Series 39 configuration, principally the nodes and storage devices, the purpose of this document being to provide initial guidance to Customer Services engineers on how the machine should be configured when installed on site. In generating this diagram, the following two factors are taken into account:

- peripheral access resilience. This means ensuring that the failure of a Macrolan component, such as an SXMC (SX Macrolan Connector) on a node, or one of the Macrolan Switch Units does not cause total loss of service due to one of the storage devices being inaccessible. In practice this is achieved by ensuring that the most important data is stored on a disc unit equipped with a dual-access controller, and connecting this controller to two completely different Macrolans. Thus if one Macrolan fails, the data can be accessed via the other until the fault is fixed.

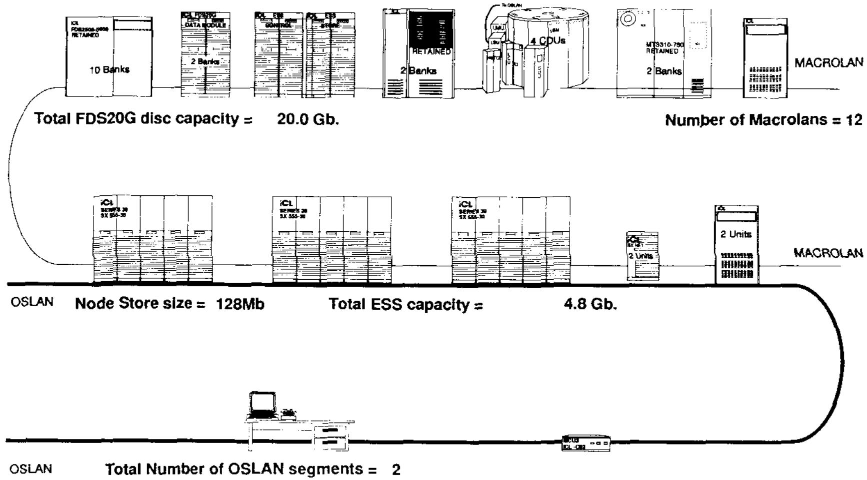

Fig. 2 Sample configuration of SX 555-30 development system (reduced)

- Macrolan loading, that is distributing the macrolan-connected peripherals evenly amongst all the macrolans in the system so as to ensure, where possible, an equal balance of macrolan traffic.

The diagram produced by the Tech Vet software is one in the 'lines and boxes' style. For each device in the configuration, eg. node, disc unit or Macrolan Port Switch Unit (MPSU), a box is drawn on the page, and a number of connector icons, representing links to either Macrolan or OSLAN are superimposed on it. Each one of these connector icons is labelled, connections between devices being shown by duplicating a particular connector icon label on both of the devices. For example, where a particular disc unit is connected to an MPSU one might see a connector labelled *MPO6* present on both devices, indicating a link between the two. Figure 3 shows how this technique looks in practice.

3.4 DRS 3000 and DRS 6000

The diagrams produced for these two machine types are very similar, giving a symbolic representation of the internal bus of the machine, and showing the device controllers attached to this bus. The device controllers are labelled with relevant text, which is derived from the CONFIGURER session so, for example, where an OSLAN controller is present in the configuration it has text associated with it indicating the number of user terminals connected to the system via. OSLAN.

There are some differences between the diagrams produced for DRS6000 machines and those for DRS3000, which are due to the different internal architectures of the types; for example a DRS6000 diagram will, as a rule, contain two busses (the high-speed private bus and VME-bus) where the DRS3000 will show only one; again while current DRS3000 diagrams can be produced on one page, DRS6000 diagrams may extend over a number of pages due to the greater capability for expansion of that system.

Because of the relatively less complex nature of DRS 3000/6000 hardware only one level of diagram is produced, which corresponds very much to the Series 39 Overview type, already described. Figure 4 shows a typical DRS 6000 diagram.

4 Implementation

CGS is a hybrid system, consisting of three major integrated components (figure 5). These are:

4.1 Image Library

The image library consists of a collection of pictorial icons used to form a diagram. The icons, along with any associated text items, are individually

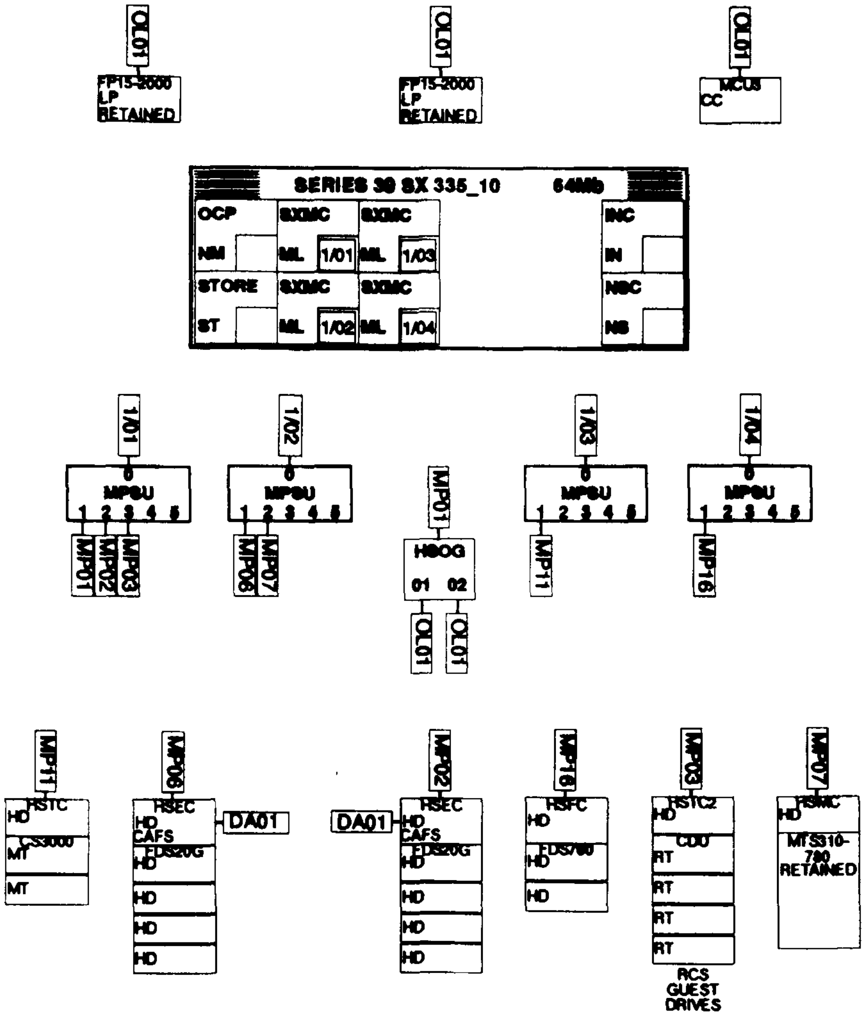

Fig. 3 Series 39 Technical Vet diagram (reduced)

constructed using the desktop publishing package FrameMaker, and once formed are saved in a format known as Maker Interchange Format (MIF). Maker Interchange Format is a declarative page description language, proprietary to Frame Technology, the developers of FrameMaker. It allows the description of any given document in programmatic form, including all the objects within it, whether graphics or text. The reason for this approach is that once the image is in MIF, it becomes an ordinary text file, which can be processed by the standard UNIX text manipulation utilities.

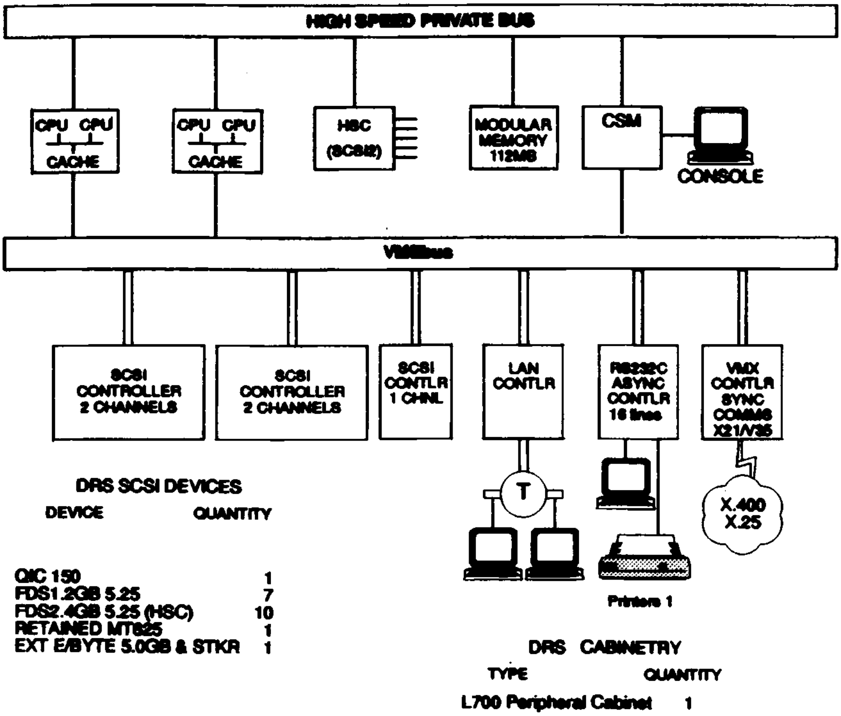

Fig. 4 DRS 6000 Configuration diagram (simplified and reduced)

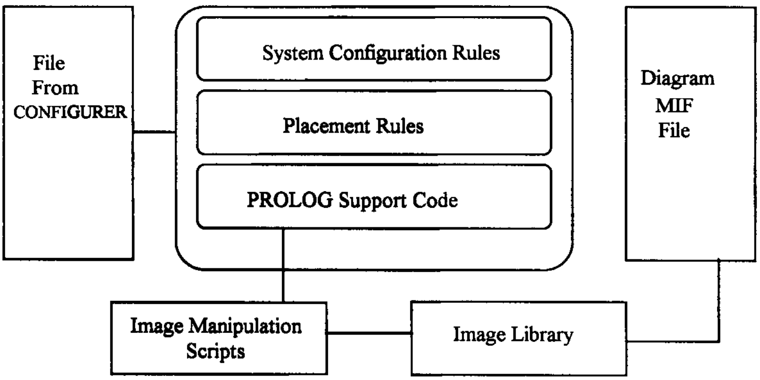

Fig. 5 CGS Components

Each icon is stored in two different forms. The first is as a full MIF file which allows it to be edited using FrameMaker if this becomes necessary. The second form is as a smaller file referred to as the *'details' file* for that icon. This details file contains only the MIF which describes the picture icon and the frame which bounds it – all other information, for example document style attributes, is removed, being irrelevant due to the way that that CGS generates a diagram.

For the whole system, there are approximately 2000 images; 30MB is required in all to store all the MIF and details files. For convenience, the image library is actually subdivided into three sections – one each for Series 39 Overview, Series 39 Technical Vet and the various DRS systems. The image library is constantly being added to as the various product ranges expand.

4.2 Image Manipulation Scripts

The image manipulation scripts are, used to process the images stored as 'details' files in the image Library. These scripts are either in the UNIX C-Shell (SunOS, 1990) macro language, or alternatively in AWK (Aho, Weinberger and Kernighan – they being the authors) (SunOS, 1990). They generally operate by searching for generic tag strings inserted by the person who generated the original image, and replacing them with an appropriate context-sensitive string. As an example, the Series 39 disc controllers generally contain a text string which reads *%CAFS*. This string is searched for, and replaced either by a blank space or the word CAFS, depending upon how that controller is been allocated by CONFIGURER. A CGS diagram consists of a MIF file which has been built up from image 'details' files.

The system includes other scripts that can: alter the position of images on the page; add text to images; produce the effect of a page throw, or add text in headers and footers.

4.3 Diagram Construction Module

The Diagram Construction module is actually a rule-based expert system, or rather a collection of three rule-based expert systems (one each for Series 39 Overview, Series 39 Technical Vet and DRS), which in total contain an expert knowledgebase of approximately 300 rule constructs. These rules fall generally into two categories, either dealing with placement issues, for example deciding over how many pages a diagram should be laid out; or with system-configuration issues, for example to which macrolan a particular storage device should be connected.

Placement rules were derived either from the general wishes of the users, or by making use of the common sense of the developers! An example rule is given below.

```
rule_80:    if     found_depth_of_Oslan_elements
            and    found_depth_of_Node_elements
            and    ((   no_mpsu_present=>no
                   and found_depth_of_MPSU_elements)
                   or
```

```
              (    height_of_MPSU⇐0))
        and   found_depth_of_Macrolan_elements
        and   add_up_depths_to_give_total

        then  know_overall_depth_ml.
```

This rule is used to calculate the total height of all the elements of a Series 39 Technical Vet diagram. The rule invokes various other rules to calculate the depths of each of the device types in the system (ie. nodes, MPSUs, Oslan and Macrolan devices). Each one of the *if...and* parts of the rule must succeed, or the whole rule will fail. Note the use of the 'or' construct within the rule (emboldened), which checks whether there are any MPSU elements within the configuration, and sets the total height of this element type to zero if there are none. There are also two operators within this construct which are worthy of mention, being non-standard PROLOG:

⇒ which means 'withdraw or test the nominated fact', and
⇐ which means 'add the nominated fact to the database'.

These operators were added to make the rather arcane database manipulation functions in PROLOG easier to use and to make the rules easier to read by non-PROLOG experts.

System-configuration rules were generated both by consulting the appropriate product literature and conducting knowledge elicitation sessions with product experts. These sessions proved to be doubly useful, as in addition to explaining current product status and best practice, they helped to forsee future developments, so that the system could be designed to accommodate changes. The example rule below is taken from the Series 39 Technical Vet software, and sets the method of internode connection; it being a rule that if the system is a Series 39 SX with more than two nodes, the nodes are coupled together using the SXnet method.

```
rule_30_4  :   if     system_type⇒sx_system
               and    node_quantity⇒NodeQty
               and    NodeQty>2
               and    internode_coupler⇐sx_net

               then   decided_sx_net_or_mru_for_sx_systems.
```

The Diagram Construction Module is written in ICL DECISIONPOWER PROLOG [Blacoe, 1992]. Within the module, the processing is split, so that some tasks (usually the lower-level, more computationally intensive ones) are performed by calls to PROLOG routines, whereas high level control is embodied within the rule constructs mentioned previously. These rule constructs are of the classic IF-THEN production rule syntax, and are interpreted using a simple backward chaining rule shell written in PROLOG. It was decided at an early stage to develop the system in this way, as this allowed new product knowledge to be added during the life of the system without having to employ an expert PROLOG programmer to maintain it. In practice, the method works very well, although it has been found that when adding a new peripheral *type* (as opposed to adding a new member of a particular peripheral *family*), it has sometimes proved necessary to make changes to the low-level PROLOG code in order to accommodate the particular characteristics of certain device types.

The diagram construction module works in two stages. Firstly, the system configuration rules are run, which generate any information required about the technical detail of the system which is additional that that supplied by CONFIGURER. For example, for Series 39 style systems this includes assigning storage devices to Macrolans; and for DRS systems, determining in which cabinets peripheral boards are to be shown.

Once this has been done, control passes to the placement rules, which carry out the placement of the individual images in the diagram; this is done by invoking individual image processing scripts, which alter the appearance of the images before copying them into the diagram MIF file.

Once a diagram has been completed, one of the FrameMaker utilities is called, in order to generate a PostScript file from the input MIF. This is then either printed on a laser printer, or sent out via electronic mail, as described previously.

At the current time the system contains in total approximately 3MB of code (including all rules, PROLOG, manipulation scripts and general support routines). Production of a diagram may take anything between one and twelve minutes, depending upon the complexity of the configuration to be represented.

5 Future Development

The Configurer Graphics Service is a continually evolving system – indeed, by its very nature it must evolve with the products which ICL supplies, or become obsolescent. Since the initial prototype of the system was made available for general use in November 1989, its scope and functionality have gradually been expanded until it now covers the entire current range of Corporate and Mid-Range Systems products, from the DRS M55/NX models right up to heavily loaded four-node and six-node Series 39 SX configurations.

In addition, improvements in the internal IT infrastructure of the Company have allowed diagrams, reports and so on, to be delivered over the Corporate electronic mail network. In the current business climate the implications of this are far reaching – typically within an hour of entering the required customer system details into the CONFIGURER, a member of the sales staff can have on his or her desk, all the components required to produce a complete proposal, including a priced system configuration report, configuration diagrams and full product descriptions.

There is little doubt that CGS will continue to evolve with the facilities offered by the CONFIGURER. The system has shown itself to be a success, consistently achieving both business and technical objectives laid down for it. Usage of the system over the two and a half years of its life has been extremely heavy – over thirteen and a half thousand diagrams to date – and the comment from users has been excellent.

6 Acknowledgements

Many people have contributed to the success of the Configurer Graphics Service project at various stages during its development, not least amongst these being the original CGS development team (Phil Boardman, Chris Clewes and Sue Hurst), who provided a firm foundation on which the present system has been built. Thanks are also due to the many people in ICL sales and marketing world-wide, who have provided helpful suggestions and comments on the service, and to the various product experts who have freely given of their time to ensure the underlying knowledge bases were correct.

Finally, but not least, thanks must go to the Configurer Graphics Service Manager, Dick Higgs, who from the very beginning has been instrumental in its development. Without his remarkable commitment and enthusiasm, CGS would never have existed at all.

References

BARTLETT, C.W.: 'S39XC – A configurer for Series 39 mainframe systems', *ICL Tech J*, Vol 5 no 3 pp 385–403, 1987.
ADOBE SYSTEMS Incorporated: '*PostScript Language Reference Manual*', Addison-Wesley, 1990.
WINSTON, P.H. and HORN, B.K.: '*Artificial Intelligence*', 2nd Ed., Addison-Wesley, 1984.
BRATKO, I.: '*PROLOG Programming for Artificial Intelligence*', 2nd Ed., Addison-Wesley, 1990.
'*ADVISER User Manual*', ICL publication R30008/02, 1985.
STEVENS, R.W.: 'Macrolan – A High Performance Network', *ICL Tech J*, Vol 3 No 3, pp 289–296, 1983.
csh(1), awk(1): '*SunOS Reference Manual*' Sun Microsystems Inc, 1990.
BLACOE, P.H.: '*DECISIONPOWER Development Toolkit for Sun*', Product Description KE-04, ICL 1992.

Biography

Steve Sharman

Steve Sharman graduated from Portsmouth Polytechnic with a degree in Computer Science. He joined ICL in 1991, working initially on knowledge-based system developments for the Advanced Technology Group in Kidsgrove. He is currently part of the Local Government and Health IT Unit, responsible for the development and implementation of Personnel Systems for LG&H and other ICL divisions.

Location Transparency in Heterogeneous Networks

A. Drahota[1] P. H. Ford[2] and I. G. Vincent[2]
[1]ICL, [2]University of Nottingham, UK

Abstract

Distribution Transparency is a key concept for developing a new manageable generation of distributed systems. A particular aspect of distribution transparency is Location Transparency. The ICL product DAIS implements an extensive set of transparency mechanisms, including one for Location. This paper describes the implementation of the DAIS Location transparency mechanism as achieved by the DTI and SERC sponsored collaborative project OASIS, undertaken by ICL, Hydro-Electric Plc, The University of Nottingham and Gid Ltd. A particular feature of the implementation is the ability to maintain Location Transparency in a heterogeneous environment of host computers and networks.

1 Introduction

The rapid growth of distributed computing has created a need for a framework for the standardisation of *Open Distributed Processing (ODP). The ISO Reference Model of ODP (ISO 10746, CCITT Rec. X.900)* provides such a framework. It creates an architecture containing the concepts, rules and standards which can be used to describe and implement distributed systems. The ODP model covers distribution of system components, their interworking and interoperability, and how portability of software components can be achieved [ISO '92].

In parallel with the definition of ISO ODP, the *Object Management Group (OMG)* have started defining their architecture for distributed object based computing – the *Common Object Request Broker Architecture (CORBA).* OMG is a very large industrial consortium with a worldwide membership of information technology suppliers and users. OMG's mission is to establish an architecture and a set of specifications, based on commercially available object technology, to enable creation of distributed integrated applications. The primary goals of OMG are reusability, portability and interoperability of object-based software components in distributed heterogeneous environments [OMG '93].

Both ODP and CORBA are raising significantly the level at which applications are programmed above the networked machine. For example in CORBA, the Object Request Broker (ORB) component of the system enables application objects to make requests and receive messages transparently, that is without being aware of or needing to accommodate the distributed environment in which they are executing. This will considerably simplify the developer's task and improve development productivity for distributed systems, because developers will not need to be concerned with the mechanisms for communicating between distributed application components.

An important difference between the approaches mentioned above and many of those adopted in the past is that distribution options are not restricted to specific support environments (e.g. to a proprietary networking protocol) on which initial development has been carried out, but it is possible to relocate the developed application system to other environments without re-design or re-coding.

ICL has participated actively in the definition of both ODP and CORBA. This is reflected in the early availability of an ODP and CORBA intercept product – the DAIS platform. Much of ICL's early activity in developing the new technology was in collaboration with other organisations, particularly in the following projects: ANSA/ISA (Alvey and Esprit II project), RICHE and Bank'92 (Esprit II projects) [Drahota '90], Sysman (Esprit III), OASIS (a project within the DTI/SERC sponsored Open Distributed Systems Architecture (ODSA) programme).

The experience gained during the projects is reflected in DAIS, which extensively exploits the work of the researchers. Many of the significant benefits, gained by users of the new technology, can be attributed to the availability of advanced transparency mechanisms in the support environment.

A transparency mechanism is a system feature which conceals some aspect of the physical system from the application developer or user. So, for example, the application programmer can write communicating programs without knowledge of any network through which the programs may need to send messages to each other; to the programmer it appears as if the communicating programs were always resident in the same system process or virtual machine. There are many transparencies which can be provided as part of a support environment for distributed processing, section 2 outlines several of these.

This paper concentrates on the description of one of the transparencies – the one concealing location of system components – and outlines the mechanism which delivers it as implemented in the OASIS project and now incorporated into DAIS.

2 Transparencies

The ODP model recognises that complexity of distributed systems must be tackled by provision of mechanisms which conceal particular distribution issues from application designers and developers. These mechanisms are known as *transparencies*, and the ODP model recognises several, including:

Access transparency which enables interworking between heterogeneous computer architectures by masking differences in data representation and invocation mechanisms (including the distinction between local and remote invocations) of operations at interfaces

Concurrency transparency which masks scheduling of recoverable operations that act on shared state at an interface, to achieve consistency (i.e. during transactional operations the ACID properties are maintained – Atomicity, Consistency, Isolation and Durability)

Failure transparency which masks the failure and possible recovery of objects, to enhance failure tolerance and system availability

Federation transparency which masks interworking boundaries between separate administration and technology domains

Location transparency which masks interface location, to enable location-independent interface identification and continued use of interfaces after their relocation

Migration transparency which is an extension to location transparency and which masks relocation of an object from the object being relocated and ensures that its clients are not affected by the move; migration is often used to achieve load balancing and reduce latency

Replication transparency which masks replication of objects; replication can be used to enhance performance and availability

Resource transparency (sometimes called *liveness transparency*) which masks variations in the ability of a system to provide necessary resources for an object to engage in actions; [ISO '92].

Transparencies are a fundamental concept in the ISO ODP standard. They are sometimes grouped under the convenient heading of *Distribution Transparency*. Their availability can dramatically simplify design and development of distributable applications, and at the same time preserves reconfiguration and enhancement options for the system. Exploitation of the transparencies also reduces the system management burden, thus releasing scarce expertise to concentrate on applications rather than on complex systems issues.

The important point about adopting the ODP approach to development is that the application thus implemented can be deployed in a single host (centralised) but distribution options are preserved and can be taken up if later required.

Aspects of distribution transparency should be selective, except for Access transparency, which is required in all cases. This means that it must be possible for a developer to make selectively visible some aspect of the distributed system (e.g. components' location, or intermittent failure of a component). Only with selective transparencies will it be possible to utilise the same system infrastructure for building vastly different applications, ranging from those which are completely unconcerned with how their business logic is mechanised in the host environment, to those which need to control precisely some physical aspects of the environment in which they exist.

The ODP model can be applied in many different implementations.

ANSAware is an exemplar implementation of ODP. ANSAware was developed by the Advanced Networked Systems Architecture/Integrated Systems Architecture (ANSA/ISA) project (1983–), mainly as a proof of concepts embodied in the ANSA architecture, which itself is the basis from which the ODP standards are evolving. The software is in use by numerous research groups around the world and it was exploited in the US to develop the NASA Astro-physical database system, one of the world's largest distributed applications.

The ICL *DAIS* platform, available since December 1992, is an ODP conformant product adopting many of the ANSAware implementation approaches. DAIS also contains the functionality of an Object Request Broker (ORB) as defined in the OMG CORBA.

In CORBA, object interworking is supported by the Object Request Broker (ORB) and a set of Object Services. The ORB enables transparent communications between objects, mediating client-server interactions. Object Services are a collection of services (interfaces and objects) that support basic functions for using and implementing application objects. Typically, an Object Service provides some generic group of functions and, at the same time, some aspect of transparency to its clients. For example, the Name Service provides a function for translating an Object Name into an Object Reference, which can be employed by its user to bind to the named object. At the same time, the Name Service is used in conjunction with the ORB to conceal Objects' Location. Objects exploiting Objects Services are written to specified Application Programming Interfaces (APIs), defined in IDL and implemented by the Object Services.

Apart from the Object Request Broker, OMG have begun standardisation for services covering Naming, Events, Lifecycle (creation and destruction), Persistence (i.e. persistent storage manager), Concurrency, Transaction Management, Externalisation, Time and Relationships. Several other Object Services are expected to be standardised, including Security, Trading, Change Management, Query, Archive, and Interface and Implementation Repositories [OSA '92].

In ANSAware and in DAIS, the infrastructure supporting objects is divided into a basic *Nucleus* (a collection of functions providing basic distribution capabilities and linkage to communications interfaces), and a set of transparency mechanisms (themselves implemented as objects). Architecturally, the transparency mechanisms are Object Services used by a "non-distribution transparent" ORB to achieve distribution transparency [Herbert et al '93].

Interworking between objects supported by different ORBs is possible by exploiting *Interceptor* objects, which perform any necessary mapping between the different environments (see section 3).

With ANSAware and with DAIS, as is the case in CORBA, an application developer describes a service in terms of its interface using an Interface Definition Language (IDL). Interface Definitions are the shared specifications between client and server objects. Clients are not aware of server implementation details, only of the interfaces made externally visible via IDL, and interactions between objects are only possible via interfaces so defined. With DAIS and ANSAware the developer also uses a declarative statement of the required guarantees (i.e. implied transparency needs) upon operation invocation within that interface. The ANSAware or DAIS tools to compile and link the application build an ORB for the application that includes appropriate transparency mechanisms. This approach emphasizes the selective nature of many transparencies and allows "plug in" transparencies to add capability to a basic ORB [Herbert et al '93]. Interoperability between application objects running on top of different ORB implementations can be achieved via Interceptor objects, which are discussed in section 3.2. The relationship between the components of both CORBA and DAIS support environments is shown in Fig. 1.

The ANSA and DAIS approach is very flexible. With DAIS, an ORB is created tailored to the needs of particular application systems. Different systems require different guarantees in areas such as security, reliability, performance, availability and so on. Many such requirements may conflict e.g. security vs convenience, autonomy vs uniformity, abstraction vs performance. The ORB will need to reflect the various trade-offs which the application developers need. Therefore, an ORB which is in part generated specifically for each application system is the most flexible and adaptable approach to meet the variety of needs that will be encountered.

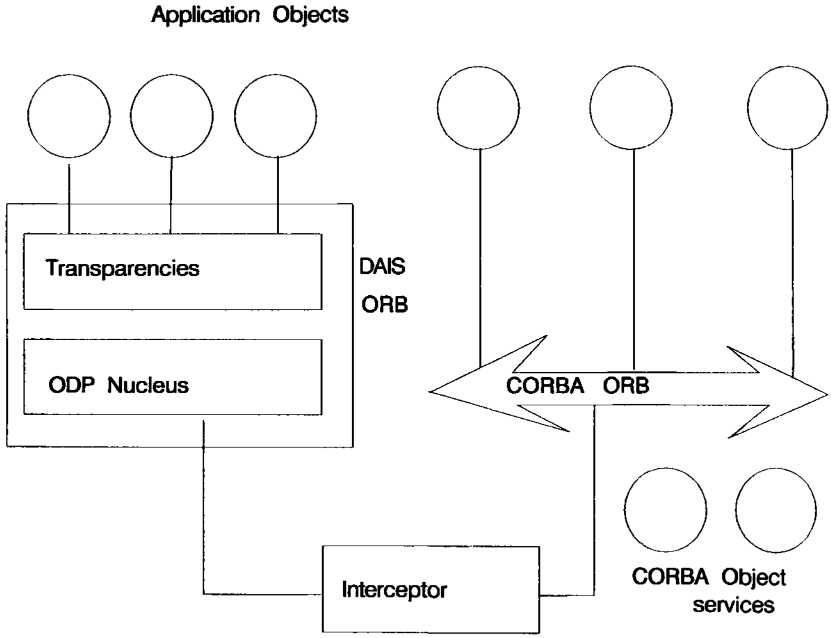

Fig. 1 DAIS Object Request Broker

3 Location Transparency

Location is an important concept in distributed systems. To be able to use the services of some Server Object, a Client Object must identify the point in space where the Server is able to receive a message on which it is to act. Usually, Location is represented by some label which a programmer uses, for example an Address which identifies to the network layer the end-point at which the server object is listening for requests. If this location is stored in the program as a pre-assigned address then the Server location is fixed,

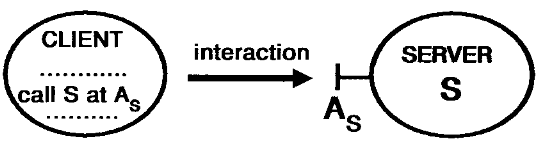

Fig. 2

as far as the Client Object is concerned (see Fig. 2). In such circumstances, when it is required that the Server Object be relocated, as will be the case, for example, when its host system is to be replaced, the Client will need to be regenerated with a new value of the address. This will increase system maintenance/management cost and hence should be avoided.

To achieve *Location Transparency*, programs, or their supporting infrastructure, need to be written in such a way that the binding between a client and a server occurs (i.e. client achieves awareness of the server's location) near the instant in time when the client wishes to use the server, and, should the Server Object subsequently migrate to another location, the client program's view of the server location can be automatically readjusted. The first capability is often termed *late binding*, the second is also known as *migration transparency*. The migration transparency scheme relies on use of *Relocators*, which are services maintaining a directory of locations for Objects that have migrated (e.g. due to recovery from checkpoint). The relocating service or its supporting infrastructure informs the Relocator of changes in location, and the Client Objects or their supporting infrastructure can request the Relocator to provide the location of the required service.

Various implementations and techniques for achieving at least limited location transparency exist and, depending on how comprehensive a support for the transparency they provide, they deliver varying degrees of benefit to system providers.

1. To avoid the need to alter application programs, many suppliers provide configuration files which allow customers to supply addressing information required for distributed access. Such a scheme eliminates the need for re-coding when server locations change, but it does require system management activity (i.e. changing the configuration files). The cost of maintaining configuration files can become quite significant when the population of objects reaches a certain level, or when location changes are frequent.
2. More advanced approaches use the concept of a *directory*, sometimes global in scope, and an associated programming interface. A client Object is programmed to interrogate a directory service, from which it

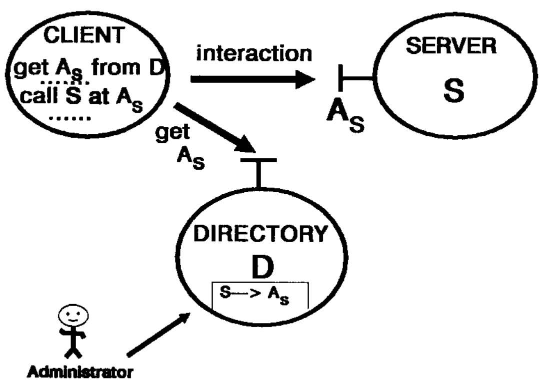

Fig. 3

obtains location details for the required service (see Fig. 3). Binding between the client and server is thus established, based on a centrally administered repository of location information. Sometimes the directory is incorporated in a service with which clients interact, and which itself conceals distribution of its parts. For example, relational database management systems adopt this approach, where a client interacts with a database server using a 'view of data' which conceals possible distribution of the data elements in other database servers. The server to which the client connects uses its directory to access transparently the information on behalf of the client and presents a unified view to the server. However, the drawback with this approach is that a client must access information via the directory bearing service, rather than directly, potentially introducing delays and a single point of failure. The directory information is usually static and requires manual updating when objects move.

3. It is possible to implement a dynamically maintained directory which becomes aware of locations of services when they are activated. This is the approach prototyped in *ANSAware* [ANSA '92], and adopted by the ICL DAIS product. In this scheme, Server Objects can "advertise" their services in the directory (known as the Trader) and Client Objects can subsequently obtain the location information from the Trader for

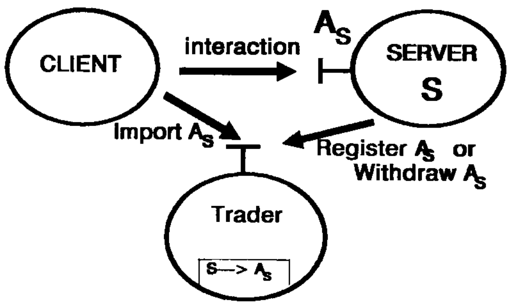

Fig. 4 Trading

the services they require (see Fig. 4). Migration of services is handled by advertising the new location and maintaining relocation information in a Relocator service. The infrastructure automatically uses the Relocator to locate migrated services. When a service is to close down it can withdraw its offers from the Trader, and thus the directory content reflects only those services that are available. The trading system can itself be distributed. It is possible to structure the system so that different application domains are serviced by separate directories which can be federated to form cross-domain associations within which objects can interwork.

The Trading scheme offers significant advantages; in particular it removes the task of managing each service's location information from the system administrator, ensuring that the systems maintain location data automatically (self administration), and hence it reduces the effort required to reconfigure distributed systems.

The scheme, as implemented in the ANSAware exemplar, functions as follows:-

- A server object, once instantiated, *Exports* an offer of service. This Export is carried out for every Service Interface which the server wishes to make publicly visible.
- The server can qualify its offer with a set of properties which the prospective client objects can use to constrain their selection to the most appropriate service on offer.
- The *Server Offer* manifests itself in the form of an *Interface Reference*, which the server object manufactures to contain necessary addressing information and protocol hints, and which client objects use to bind to the service. An Interface Reference is essentially a name for the service, together with information enabling the client to bind to the service safely. As such it contains, apart from addresses, some end-to-end check information, which can be used by communicating objects to assure correct connection.
- The use of Interface References containing addressing information is attractive, in that once a client has obtained the reference (from Trader or any other object in the system) it can bind to the service without the need for a look-up in any centralised repository of addressing data.
- Client objects *Import* the Interface Reference and use the addressing data contained within to establish communications with the server.
- It should be noted that all the manipulation of Interface References, establishment of communications links and the actual transmission of data are handled by the ANSAware platform. The application programs are totally unaware of the complex processing associated with networks, they merely issue procedure calls on the remote services. It is possible to introduce transparently *Relocator Objects*, which will transform Interface References in cases where the targeted interface has migrated in the system (e.g. when a service is moved to another host).

The ANSAware implementation manufactures Interface References which contain a network address of the host computer and communications port identification on which the server object (behind the Interface Reference) will be listening for incoming messages. The platform supports communications via several protocols, namely UDP/IP (User Datagram Protocol/Internet Protocol), TCP/IP (Transmission Control Protocol/Internet Protocol) or, in the case of intra-machine communications, IPC (Inter-Process Communications). However, whilst allowing an Object to communicate with others using any of the supported protocols, the implementation supports only bindings between Objects which reside on the same physical network segment and share at least one communications protocol. The implementation provided in ANSAware is a subset of the ANSA and ODP architectural approach to providing Location Transparency.

Architecturally, it is assumed that the information contained within an Interface Reference, which is effectively a name for the service interface (or an object), is only valid within the context in which it has been created (e.g. a particular homogeneous network). Such name is said to be *context relative*. If the Interface Reference is passed outside the context in which it has been created, it must be transformed in the gateway between the networks, such that the addressing information is valid in the new domain. The mechanism for transforming the Interface Reference within the gateway is sometimes

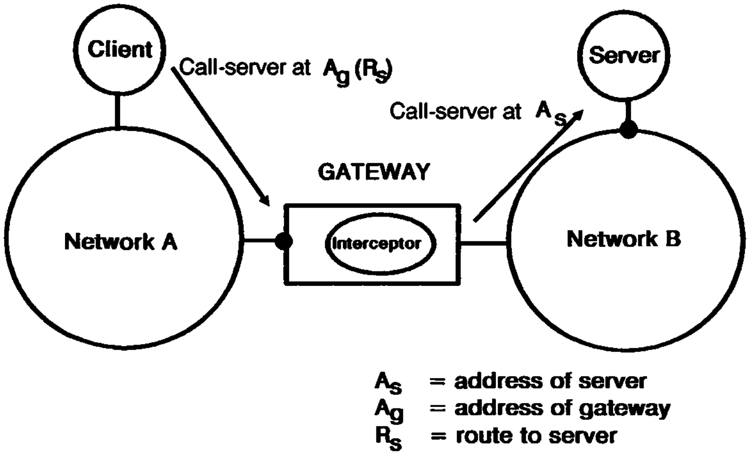

Fig. 5 Interceptors between networks

also called an *Interceptor*. Fig. 5 illustrates an interceptor responsible for handling addressing information between two separate networks. A client object in network A perceives the server object in network B as co-resident in its own network, because it is addressing the gateway (which resides in both networks) which itself conceals the exact location of the server.

3.2 Interceptors

Interceptors mask boundaries between domains from applications. In general, administrative boundaries require controls to be imposed and records to be kept. Technology boundaries require transformations to be made. ANSA identifies two types of Interceptors – Fixed and Nomadic [Herbert et al, '93].

Fixed interceptors provide transparent access to interfaces and object migration across boundaries. They intercept invocations and transform arguments and results (e.g. addresses). They are part of the communications path between the interacting objects. Interception can occur at the source, at the destination or at an intermediate location. Where an administrative boundary is being crossed the interceptor applies administration-specific access controls and re-assigns responsibilities for security and dependability [Herbert et al, '93].

Technology interceptors carry out protocol and data translation over technical boundaries; they are sometimes known as "gateways" or "protocol converters".

Nomadic interceptors provide a batched store-and-forward migration service. Objects may be relocated in a distributed system via removable media (e.g. magnetic tape or optical disc) on which they may be stored, or arising from the disconnection and subsequent reconnection elsewhere of mobile computers. The objects so moved may need to be accessed in their new location from their previous client, and they may also retain (context relative) references to other objects they may require to access. Nomadic interceptors are responsible for repairing references after an off-line move has occurred.

ANSAware, which is essentially a 'proof of concept' implementation of an ODP support environment, does not currently implement a full range of Interceptors and hence it does not cater for the transformation of the Interface References passing between different networking domains.

In the OMG model [OMG '91], the responsibility for binding a client object with a server object rests with the Object Request Broker and the Object Services. An object identity, which as in ANSA, can be a context relative name, is translated into a reference pointing at the required service. OMG does not specify how such a translation should be implemented.

Clients and Servers running on the ICL DAIS platform, which extensively exploits ANSA results, can communicate via ANSAware supported protocols, as well as via OSI protocols. Additionally, ICL determined that a requirement exists for communications over complex networks, where protocol converters (gateways) are used and where the Client Object's and Server Object's hosts may not support a common protocol or addressing scheme.

The following sections elaborate the requirement and describe the solution implemented to meet it.

4 Communications Requirement

The DAIS platform can run in a number of host environments, including:

ICL systems: DRS 3000/6000 with SVR4 Unix, Series 39 with VME,
Hewlett Packard's HP 9000 HP-UX,
Sun's Sun 4/SparcStation with SunOS or Solaris 2,
IBM compatible PCs with MS-DOS or Windows 3.1 or OS/2 or SCO-Unix,
Digital's VAX with VMS.

The initial DAIS release supported communications via TCP, UDP and IPC.

Through analysis of customer requirements, it was identified that support for heterogeneous networks had to be provided, whereby Client Objects could communicate with Servers residing on different networks and possibly using different addressing schemes. It was required that full Location Transparency was to be maintained in such circumstances. This requirement can be viewed as one for implementing *Federation Transparency*, where two network domains are being federated into a single environment for a distributed application system.

In common with ANSAware, DAIS uses Interface References and Trading. A problem arises when a server manufactures a Reference containing an address which is not recognised by the Client importing the Reference (e.g. DTE/TSAP (Data Terminal Equipment/Transport Service Access Point) versus NSAP/TSAP (Network Service Access Point/Transport Service Access Point) formats), or which appears valid but is not reachable from the client (as would occur when compatible networks are separated by another type of network and gateways e.g. an X.25 Wide Area Network (WAN) between two Local Area Networks (LANs)).

Further complications are introduced by different addressing restrictions demanded within the various supported host systems. For example, some systems do not support dynamic TSAP allocation or connections to unknown addresses. Within OSI communications, Sun systems can use only pre-configured "paths" to known locations, and VME systems require that all TSAPs be catalogued before use. Such restrictions would dramatically inhibit the flexibility and dynamic nature of an application system based on DAIS, unless tackled within the DAIS platform.

Inter-network communications can be achieved via intelligent or dumb gateways and/or routers. Dumb gateways are preconfigured with all paths through them in the form of TSAP-to-TSAP mappings. The DAIS platform needs to be aware of these mappings if it is to be able to handle addresses in network domains separated by such gateways. An object sending an Interface Reference to another across a dumb gateway would not normally be aware that the message will leave its network domain and enter another; the addressing scheme implemented within gateways makes the remote location appear to be located on the same network as the sender. Furthermore, it is impractical to consider inserting DAIS Interceptor code into all dumb gateways, especially as some are not readily programmable.

Regardless of the above problems, it is required that DAIS servers can Export offers of services and that DAIS clients can Import these and bind to the servers. Therefore, means are required for clients to obtain addressing information which is valid in their network, because only local network addresses can be used in an environment of multiple networks connected via "dumb" gateways. Similarly, means must be provided for servers to Export addresses which any potential clients can use, i.e. addresses valid in remote as well as the local networks.

5 DAIS Solution

The OASIS project has adopted a pragmatic approach reflecting the limitations imposed by various networking technologies and host operating systems. This approach is now included in the DAIS product.

A global network is decomposed into network domains, which can be identified by name. A network domain name is included as part of the addressing information contained within Interface References, and permits a client to recognise if an address is resident within its own local network.

In conventional networks, to make use of "dumb" gateways, network administrators must predetermine which services will need to be accessed from which remote networks and they must provide address mappings within the gateways. Furthermore, the administrators need to configure the potential clients of such services with gateway addresses and appropriate route identifiers (typically TSAPs), so that the clients can connect to the required remote services. In DAIS, the clients should not need to be configured to contain location data for their servers; it should be sufficient for servers to advertise their availability via the DAIS Trading service.

It is possible to utilise the administrator-created address mappings and make these visible to the DAIS server objects via a *DAIS infrastructure library*. Once this is achieved, the servers can Export offers containing addressing data valid in all the network domains from which clients are capable of access to the servers. The DAIS infrastructure library operates on network management information and the application object remains by the library isolated from any configuration-specific details.

The DAIS Interface Reference (IRef) is therefore an enlarged version of the ANSAware IRef, containing addressing hints for each protocol supported by the server (host) for each network from which it is reachable. Hence, the DAIS IRef contains not only the server host addresses, but also the addresses of various gateways in remote networks, which recognise routes to the server exporting the IRef. The exporting of a "fistful of addresses" caters for all possible clients in any of the network domains from which access may occur. The onus of selecting the correct address is on the client (or rather the DAIS infrastructure supporting the client), which can determine which address is appropriate within its domain.

The OASIS implementation satisfies the needs of the vast majority of operational distributed systems, where the number of different networks linked together into a large heterogeneous net is, normally, relatively low. Typically, a large organisation will operate a single backbone network to which smaller Local Area Networks are connected.

Architecturally, the solution does not scale up, because the effort in managing a very large number of remote addresses and the size of the Interface References become excessive. Such large heterogeneous networks will be supported by full implementation of inter-domain interceptors in programmable communications gateways. Nevertheless, the current implementation will support the majority of customer needs. In most situations there will be many more client objects than servers and it is the latter for which network paths need to be configured in gateways.

In some systems, a server can only listen for incoming messages on a pre-catalogued port (TSAP). This is obviously also the case where a dumb gateway is used which maps remotely generated messages onto a pre-configured local TSAP. Furthermore, some implementations only allow clients to address known (pre-catalogued) addresses. These limitations have been tackled in OASIS by implementing *TSAP Managers.*

The role of a TSAP manager is to maintain the pool of 'TSAP to TSAP mappings' and, on request, to allocate one to a DAIS Object. It will also release a mapping to a free pool once instructed by a DAIS object to do so. Where a dumb gateway is present, the TSAP Manager will provide the "address on the other side of the gateway" to a server wanting to perform an Export.

Some host systems demand that client objects can "call" only on known (i.e. pre-catalogued) services, and this would mean that all possible locations for all server objects would have to be pre-catalogued to support object migration. DAIS will remove this limitation by providing dynamic cataloguing (or registration) of services. Once a client receives an offer of service, it can request that this is registered within its host system, before it attempts to bind to the service. In this fashion, the host's registry of remote services is built up dynamically, and system administrators need not pre-configure accesses to the whole distributed system.

6 OASIS Exploitation of DAIS

The implementation of the Location Transparency mechanisms described earlier was a by-product of the OASIS project. OASIS aim is the confirmation of ODP applicability in commercial data processing. This is to be achieved by the development, live usage and detailed evaluation of two pilot distributed applications at Hydro Electric in Scotland.

The two applications were specified to tackle particularly important business requirements of Hydro Electric:

- the need for system capability to identify customers, based on possibly limited and/or inaccurate data
- provision of an integrated view of customer information, based on exploitation of existing Hydro Electric's application systems and databases
- the need to make available for marketing purposes a variety of views of information maintained in the various systems by Hydro Electric

The OASIS consortium developed two applications based on the DAIS platform:

Customer Identification and Information System (CIIS)
and
Marketing Application.

The marketing application provides a unified view of the Hydro Electric databases, allowing users to select information using a variety of enquiry mechanisms. The application logic is fronted by a sophisticated graphical user interface, developed using X-Designer, and running on either a PC or a Sun workstation. The access to databases is managed by the DAIS Information Service, which makes the totality of data appear resident in a single homogeneous database. The pilot system accesses data stored in an IDMSX database on a VME mainframe and in an Oracle database running on a DRS6000 Unix system.

The CIIS application uses an object database holding summary information about customers; this is implemented in Ingres on a DRS6000 host. The application contains powerful fuzzy matching logic which can cope with incorrectly captured data. Once a customer is identified, the application can connect the user transparently to Hydro Electric's core operational systems (e.g. accounting) running on S39 VME mainframes under TPMS. Components of the CIIS application run on Sun Classic workstations (Solaris 2) or on personal computers (Windows 3.1), and on DRS6000 (Unix SVR4). The CIIS pilot is deployed in the Hydro Electric customer call centres at Perth and Inverness, the VME services are situated in Aberdeen. The systems are connected by a complex network including Local Area Networks and a Wide Area Network, with a variety of gateways and bridges used.

Both applications have been developed with full location transparency between their components. The applications are even unaware from which databases the information they are processing originates.

The OASIS pilot is due to complete in December '93 and a detailed evaluation report will then be available. Early experience shows that developing applications in an environment containing a range of transparency mechanisms reduces application complexity whilst preserving deployment options. In the OASIS case, changes in network topology, reorganisation of the Call Centres and a switch from PCs to Sun workstations, all of which occurred during the project's lifetime, had no impact on the application development.

7 Summary

Location transparency is one of the most important aspects of Distribution Transparency. Its availability, in conjunction with Access Transparency, contributes significantly to the desired flexibility of distributed systems enabling them to be reconfigured dynamically when circumstances demand it. With Location Transparency, application components can be started on new host computers when their original hosts fail or are replaced, and without the need for reprogramming or for system management activity to modify addressing information.

Location transparency provides a basis for building sophisticated load-balancing mechanisms, where application components can be relocated from over-utilised resources onto less loaded ones. Load-balancing could be performed continuously or periodically. For example, it could be exploited in distributing workloads from a few production machines onto both production and development machines at peak load periods. Organisations with heavy seasonal processing peaks (e.g. retailers during sales periods) could reduce the amount of spare processing capacity and hence reduce their IT expenditure.

When Location Transparency is delivered automatically by the supporting infrastructure and the development tools, as is the case with DAIS, it results in application systems which are inherently easier to manage, which have a potential for change built into them and which, most significantly, are quicker to design and build than conventional programs attempting to retain a similar degree of flexibility.

The experience in OASIS shows that Location Transparency can be maintained by the application support infrastructure even in environments of inflexible networking technologies and practices where implementation of full interceptor objects is not practicable.

Acknowledgements

The authors would like to thank a number of people for their assistance with this paper and for implementing many of the ideas for providing a comprehensive Location Transparency in the DAIS product. In particular, the work of Martin Walsh and Walter Eaves from the ICL GMC OASIS team, and assistance from Nic Holt of ICL CSD are much appreciated. Of course, the excellent work of the ANSA team at Cambridge has provided the foundation and inspiration from which the OASIS results grew.

References

[Drahota '93] DRAHOTA A, RICHE-Reseau d'Information et de Communication Hospitalier European, *ICL Tech. J.* Vol 7 No 2, 1990;

[ISO '92] *ISO/IEC 10746–3: Information Technology – Information Retrieval, Transfer and Management for OSI* – Basic Reference Model of Open Distributed Processing (Prescriptive Model), 1992 (CD).

[ANSA '92] *ANSA Reference Manual: – An Overview of ANSAware 4.0* (Doc. RM.099.00); *Application Programmer's Manual (Doc.* RM 102.00); System Manager's Guide (Doc. RM 101.00): APM Ltd, March 1992.

[Herbert et al '93] *ANSA, ORB Interoperability:* Herbert A, Otway D, Van der Linden R, Watson A, Domville I (BNR); APM Ltd, May 14, 1993;

[OMG '91] *The Common Object Request Broker: Architecture and Specification;* OMG Document 91.12.1, December 1991.

[OMG '93] *Object Services Request for Proposal 2,* OMG TC Document 93.6.1, June '93.

[OSA '92] *Object Services Architecture (revision 6),* OMG TC Document 92.8.4, October '92.

Biography

P.H. Ford

Prof. Peter Ford is currently Pro-vice-Chancellor at the University of Nottingham with overall responsibility for IT development, and research and consultancy links with Industry and Commerce. He is a Chartered Engineer holding a chair in Information Technology in the Department of Computer Science, and is also Director of the University's Cripps Computing Centre and IT Institute. In addition, he is on the Board of NCC and is a Director of a number of small companies specialising in consultancy and software development. He has previously worked in the aircraft and computer industries as well as in local government. His current research interests include network-transparent editing systems and development of Management Information Systems. He also acts as a consultant on networking issues and information systems development, both in the UK and abroad.

I.G. Vincent

Ian Vincent has been System Manager within the University of Nottingham Computing Centre for the past 10 years. He has been responsible for implementation and development of major computing systems, initially working with VME operating system and lately with Unix on a variety of hardware platforms. His early interests were in the area of filestore and job scheduling, but these have widened to include communications, particularly file transfer and terminal handling. He has worked closely with ICL development staff for a number of years, during which time he designed and developed a variety of software, including code for reinstatement of files from category archives and the low level code to handle X.29 connected terminals. These developments were adopted by ICL and were included in the VME product line.

A. Drahota

Tony Drahota is a member of the Technical Directorate of the ICL Government and Major Companies division. He has over 15 years of computing experience gained in various technical and project management roles within ICL. For the past 2 years he was the ICL technical manager for the OASIS collaborative project sponsored by the DTI and SERC under the Open Distributed Systems Architecture (ODSA) programme. Previously, he managed two ESPRIT projects, RICHE and Bank '92, and development of Management Support products for use within Government Departments.

Future Office Interconnection Architectures for LAN and Wide Area Access

Uri Baran
ICL Retail, Bracknell, UK

Abstract

This paper is about the interconnection issues involved in designing for future office environments. It introduces an architecture for all communications requirements and focuses on ISDN as the service best suited to satisfy most requirements. Practical issues and future technologies are also considered with the aim of providing such an architecture with a means for implementation and to ensure its longevity.

1 Introduction

This paper covers some of the issues involved in the design of a flexible office communications architecture for voice, data and video applications focusing on ISDN.

It considers current and future technologies, compatibility, applications, practical implementation issues and relevant standards.

Although the paper touches on Computer/Telephone Integration and future WAN/LAN (Wide Area Network/Local Area Network) technologies, its main aim is to outline an architecture to satisfy emerging requirements rather than longer term issues although compatibility of the architecture with these issues is considered.

The paper is concerned only with the physical and topological requirements of the protocols and networks to enable access to the service by the equipment, so the platform and user equipment are outside its scope.

It is anticipated that the content will be of concern to Network and Telecommunications managers interested in a universal infrastructure to satisfy both requirements in the future, despite the corresponding alignment of their responsibilities.

2 Future Office communications topology

Medium to large offices are moving from the provision of a LAN connected PC and a telephone on each desk, towards the integration of the two. The incorporation of new technologies and services enables higher speed and more easily accessible and controllable wide area connection for data, voice and video.

Much useful background information is given in the November 1992 issue of this journal about networking and related technologies, while the May 1991 issue contained several papers on ISDN (Integrated Services Digital Network).

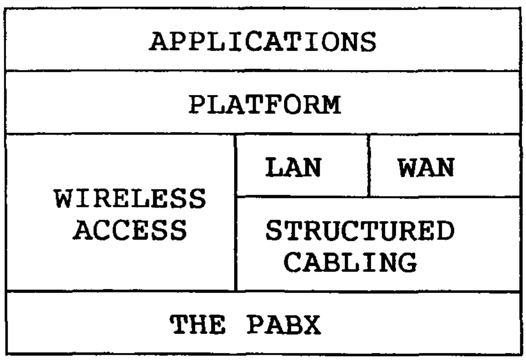

Fig. 1 Communications Architecture Model

Fig. 1 above shows a model of a communications architecture suited to the office environment. As in the case of the famous ISO 7-level OSI model, services are provided by each layer to those above it. In this case the method of service provision is explicitly stated to be either wired or wireless.

The application resides on the "platform"; the platform communicates with the outside world by means of a LAN and/or WAN interface provided from the 'PABX' (Private Automatic Branch Exchange) via a Structured Cabling scheme. The alternative is direct wireless access to the 'PABX', once again for LAN and/or WAN access.

The 'PABX' is in inverted commas as it is an open question as to whether the central communication device will be a switch or use repeater technology (which is associated with *hubs* see Sections 2.4 and 3.3).

The architecture allows LAN communications via the PABX (which occur within the model of Fig. 1) as well as WAN communications outside the model through the 'PABX' to similar models in remote offices.

A brief description follows of some of the elements of the model and the way they are expected to evolve.

2.1 Applications

Applications are the driving force for blurring of the LAN/WAN boundary. Emerging applications that demand an integration of LAN and WAN services are primarily those involving Multimedia (MM). The term covers the "input, output, processing, transmission and storage of audio, still images, motion video and animation, all in digital form" (R. McCann). MM is primarily a PC-based technology used for presentations, training and the provision of mixed information. It is finding various applications in retailing. The natural progression of MM technology is to provide wide area connections, enabling updating of information, interactive communications and data retrieval. The MM approach is becoming increasingly important in actually making sales.

Two subsets of MM technology are Video- and Desktop-conferencing. Videoconferencing technology is moving from proprietary platforms to PC platforms using international standards (CCITT Recommendation H320). Desktop conferencing is described in [Fuller, 1991]. Both of these applications and their increasing ubiquity are stimulating the need for high performance WAN access for PC's (usually via ISDN).

Other application areas include communications with other companies, computer aided design (CAD) and database access – see [Meers, 1992].

Some new products, such as Lotus Notes, embody 'Groupware', a concept that enables a group of individuals to work together on common business problems irrespective of place or time. The idea of the '*agile*' corporation, coming from the Iacocca Institute report "21st Century Manufacturing Enterprise Strategy", underlines the application areas suited to this architecture. An agile organisation uses technology aimed at getting the right information to the right person at the right time.

Another similar concept, is that of the '*virtual*' *corporation* [Byrne, 1993] namely one that can form alliances/joint ventures and exchange information with other companies by dint of a suitable compatible information architecture. Thus dispersed project team members can remain at their desks yet partake collectively in major joint activities.

2.2 Platforms

Discussion of the platform is outside the scope of this paper: it can be regarded as the processing unit that runs the application. For most office applications it is a PC/Workstation.

2.3 Wireless Access

Fig. 1 illustrates wireless access to the PABX for the provision of WAN and LAN services. To date, wireless WAN access via a PABX is provided by one of two technologies see [Flatman AV & Weiser PG, 1992].

2.3.1 *CT2 - Cordless Telephony 2.* This technology was developed in the UK and forms the basis for a European non-mandatory standard. CT2 has the capacity for data applications up to 32kbps per channel as well as its voice applications.

2.3.2 *DECT.* Digital European Cordless Telecommunications (DECT) is a higher performance technology which is a mandatory EC standard. It provides usable data rates up to about 1Mbps.

Wireless LAN access most often uses Spread Spectrum technologies [Flatman & Weiser, 1992]; many products are available with performances up to 2 Mbps. Most products use the 900 MHz or 2.4 Ghz bands. Higher performance products exist (Motorola's Altair) and will emerge from the recommendations of the ETSI RES10 and the IEEE 802.11 committees. Mainly due to the lack of standardisation none of these products currently support WAN access via the PABX. PABX manufacturers are hampered from entering the market which is currently very fragmented. Again, the CDMA architecture of these systems is not well suited to large numbers of voice channels.

Of the PABX access technologies, only DECT has been used for LAN products (in Olivetti's NET3 product). PABX access via DECT is expected before the end of 1993.

2.4 LAN access

Considering the variety of types of LAN in use, LAN access provision to the platform should at least cope with the most popular types. Most popular LAN standards originate from the IEEE 802 Committee, apart from the FDDI standard developed by the American National Standards Institute (ANSI) X3T9.5 task group and the Arcnet standard developed by Datapoint Corporation.

Types of LAN standards produced by the IEEE 802 committee include:

IEEE 802 Committee	*LAN Standard*
802.3	CSMA/CD – usually known as Ethernet
802.5	Token ring
802.11	Wireless LANs

The principle of the 'hub' is often associated with LAN access. It embodies the idea of a central point with multiple connections fanning out in all directions. All LAN topologies (i.e bus, ring, star etc) can be implemented in the form of a hub with the connections for each user coming from and going to the hub. The hub itself is an active device and it is ideally suited to starwired structured cabling (see section 2.6).

LAN access is usually from intelligent hubs (or concentrators); modern hubs provide resiliency, access to common LAN types, inbuilt bridging, routing, management, fault isolation and network reconfiguration all within a single modular chassis. Hubs tend to be found at the point where the structured cabling from each user's desk can be accessed (the communications closet); they serve the local users.

2.5 *WAN access*

Providing the platform with access to the WAN must cope with the wide range of WAN services provided by the local TO (Telecommunications Operator). Wide Area Network services can be considered under the following categories (with some regional variations).

2.5.1 *Circuit Switched.* This method involves setting up an exclusive connection between two points providing real time service for as long as required. Two types of circuit switched services are provided:-

Analogue circuits. Suitable for voice and data transmission with data rates currently (non-compressed) up to 19200 bps (V.32terbo).
Digital Circuits. ISDN provides the commonest international service in this category with voice and data service bandwidths between 64 kbps and 2 Mbps. ISDN follows the CCITT I-series of Recommendations (I.100 - I.600).

2.5.2 *Leased Lines.* Both analogue and digital facilities are available for connecting two sites permanently together to provide voice and/or data connection. The service levels are highly dependent on the local TO.

2.5.3 *Packet Switched.* Packet switched networks enable the sharing of transmission facilities for multiple users, unlike circuit switching which dedicates individual transmission facilities to link two points. This is achieved via a network of switching nodes providing multiple paths for the transmission of data packets. The international standard governing direct access to the switching nodes is CCITT Recommendation X.25.

2.6 *Structured Cabling*

[Flatman, 1987] foresaw standardisation of building communications cabling. Subsequently, standards have emerged namely the US EIA/TIA-568 Commercial Building Telecommunications Wiring Standard and a European version ISO/IEC JTC 1/SC25 Generic Cabling for Customer Premises, currently in draft. These standards define a generic communications wiring system supporting multiproduct, multivendor environments that do not distinguish between computing or telecommunications cabling i.e a single cable type and topology supports both computer and telecommunications cabling.

Structured cabling is starwired with the cables (for types see section 3.2) fanning out from central points, usually called communications closets, to users and other communications closets.

2.7 *The 'PABX'*

The Private Automatic Branch Exchange (or Switch as it is sometimes known) is found in medium to large office type environments. It enables telephone users to make both internal calls and to dial out directly without need for an operator. A PABX differs from the keyphone systems found in small offices; a keyphone has no central switch and all exchange lines are available to all attached telephones; thus some of the equivalent 'intelligence' of the switch is distributed to the phones.

The switching functionality required for internal and external voice communications maps well onto the data communications environment for LAN and WAN access. Its LAN equivalent, the 'Hub', is described in the LAN access section. Whether the central communications device is a hub or a PABX is immaterial. The term PABX has been chosen because hubs tend to provide access to shared bandwidth LANs (acting as repeaters) with the trend being towards switched (PABX-like) LANs in the future.

There have been many attempts to add data facilities to the PABX. An early attempt called DOV (Data Over Voice) used the starwired PABX cabling to provide low speed inter-office data connections for PC and Terminal type equipment. This was achieved by splitting the voice and data signals and routing them to the PABX or computing equipment with switching equipment.

A further evolution of the PABX known as an ISPBX (Integrated Services PBX) acts as: "The link between the user and the ISDN" [Lane, 1987]. ISPBXs are compatible with both the PSTN (Public Switched Telephone Network) and the ISDN (both private and public).

ISPBXs handle voice as well as data in all its forms i.e text, fax and video.

[Lane, 1987] identifies the benefits of mixing voice and data on a single network based on PABXs as: reduced costs; increased availability and flexibility. He also says there is logically a need for a single communications link to each desk since the predominant costs of such links are those of their installation.

3 Integrating the components

The ideal solution to full integration would provide the following:-

3.0.1 Integration of voice and data via a single network service meeting all requirements

3.0.2 A single communications connection to each desk

3.0.3 A single cabling infrastructure

3.0.4 A central PABX (or hub) giving both LAN and WAN access

3.0.5 The ability to interwork with existing equipment, protocols, LANs, WANs and applications.

Ideally, all the above should be based on international standards in each area. Each will now be discussed separately.

3.1 Integrated voice and data services/Single communications connection to each desk.

Dealing with both 3.0.1 and 3.0.2 together, ISDN is the only widely available service to provide such facilities, although several emerging alternatives such as Fast Packet, ATM, MANs and FDDI II see [Meers, 1992] may do so in the future. However, there are several factors affecting the choice of ISDN. These include: Tariffing; Cost and availability of equipment; Alternatives (currently available): Compatibility with emerging applications; Interworking capability (with existing services and equipment).

ISDN can provide both LAN and WAN services for both voice and data and can provide the single connection to the desk. However, realistically ISDN is unlikely to be used for all of these for a variety of reasons.

3.1.1 *Bandwidth.* Although ISDN could provide up to 2 Mbps (Primary rate ISDN), to each desk, this would prove too expensive and its division into 30 separate 64 Kbps B-channels makes it unsuited to LAN applications. Typically then, the ISDN service to the desk would be the basic rate service with its two 64 Kbps channels. With the current cost of a PC interface card for this service at around £700 compared with about £60 for a 10 Mbps 10BaseT connection (Ethernet on starwired twisted pair cable), ISDN is unattractive for local data communications.

3.1.2 *Voice.* Voice communications also are not expected to move quickly to ISDN. Because of the technology involved, an ISDN telephone can be expected always to be more expensive than a standard analogue one; current prices are about £200 for ISDN compared with £15 for the analogue. Although the sound quality, connection time and various ISDN supplementary services add value over standard PSTN connection, it is difficult to justify the added cost except in specialised situations.

3.2 Single cabling infrastructure

The US EIA/TIA 568 Standard for Commercial Building Wiring is the only option within this category; the forthcoming European standard is based on it. However, under this standard there are several options for the type of cable:-

3.2.1 Four pair 100-ohm unshielded twisted pair (UTP)

3.2.2 Two pair 150-ohm shielded twisted pair (STP)

3.2.3 50-ohm coaxial cable

3.2.4 62.5/125 micrometre optical fibre cable

In practice, there are few requirements for coaxial cable, which does not support the multiplicity of LAN and WAN types required of such a system.

Optical fibre can be compatible with most types of LAN and WAN but usually via conversion from electrical to optical interfaces. This is expensive and, usually, unecessary except in cases requiring electrical isolation, high security and high bandwidths.

Two pair 150-ohm cable (or IBM Type 1 as it is also referred to) is a relatively expensive, large diameter but high performance cable. Its main limitation is its two- rather than four-pair construction. This results in limited ability to cope with RS232 communications when more than 4 signals are required, and in minor incompatibilities with 10BaseT because its impedance is outside the range specified in the standard. It provides an expensive but relatively reliable medium and one that is relatively inflexible and difficult to install. It is also not compatible with the AT&T/HP proposal for a 100BaseVG 100 Mbps LAN standard (which requires 4 pairs).

Although EIA/TIA 568 calls for 4 × 100-ohm unshielded twisted pairs, most manufacturers are providing shielded variants that meet all the performance requirements of the standard. Addendum TSB-36 introduces two higher performance variants called Category 4 and 5 that will enable operation up to 16 and 100 Mbps respectively compared with the 10 Mbps rates specified for the standard Category 3 cable referred to in EIA/TIA 568. In general, this type of cable meets all the requirements and forms the cable of choice if the criteria are cost, flexibility, compatibility with existing LANs/WANs and futureproofing. The Category 5 variant is also tending to be the first choice for future LAN developments such as 100 Mbps Ethernet as well as CDDI.

The European Generic Cabling draft standard specifies 100 Ohm cable as preferred with a 120-ohm option as an alternative.

3.3 Central PABX for LAN and WAN access

As described in the section on LAN access, at medium to large sites LANs tend to be accessed via intelligent hubs. Higher-end ranges of hub also provide WAN access and internetwork bridging and routing. Few of these products currently provide access to voice networks. Yet the technology employed in the high end hubs is changing from shared bandwidth to dedicated (switched) bandwidth. An example is the incorporation of switched Ethernet techniques (from companies like Kalpana) into the hubs. This is illustrated in Fig. 2.

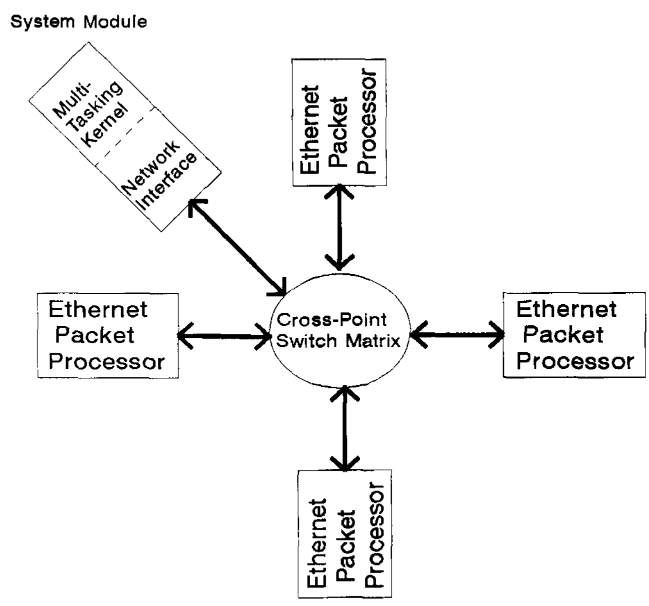

Fig. 2 Kalpana Etherswitch architecture

Figure 2 shows the Kalpana Etherswitch Architecture made up of the system module that learns the location of each host address and sets up the address table on each ethernet packet processor. This enables the configuration of the cross-point switch matrix on a packet-by-packet basis.

The method of switching Ethernet is similar to that used by PBX's, in that parallel connections are created between nodes on Ethernet LANs. This effectively increases the Ethernet bandwidth from 10 Mbps to multiples of it depending on the number of ports on the Ethernet switch.

Another development is of hubs based on ATM (Aysnchronous Transfer Mode). ATM is a new high speed LAN/WAN technique that is also based on a switching architecture. ATM represents a set of CCITT switch interface standards developed for WANs but adapted for LANs. ATM is still some way from being generally available for WAN use and is currently being promoted by hub manufacturers, initially for backbone LAN use with future uses for the workgroup LAN and, eventually, in the WAN. ATM switches will provide ports compatible with current LAN types. It is anticipated that its starting data rate of 155 Mbps will be compatible with Category 5 UTP although this aspect is still under development.

No Hubs yet provide ISDN ports for direct connection to the user but they can have ISDN connection as one of the WAN service options. However, ISDN PABXs are available that give both voice and data access for multiple user ports but, at present, no other LAN or WAN facilities.

Consequently, there is currently no single hub or PABX that sufficiently integrates LAN and WAN access to enable its exclusive use for all communications requirements. There are however differing architectures available to meet varying requirements; these are described in the following sections.

3.4 *Interworking capacity*

ISDN really scores in connection to WANs; it replaces the existing voice network but its uptake in this application depends on tariffing and equipment costs. In the UK, an ISDN phone call costs the same as a normal PSTN call but the line rental per quarter is about £84 compared with £60 for two business lines. The installation cost of ISDN is £400 compared to about £300 for two business lines to which it is equivalent.

For data applications, ISDN can absorb most if not all other existing WAN services, providing a universal WAN service for all applications. It is circuit switched and enables faster data rates than analogue lines; a fair comparison with digital leased lines depends on the data rates and rental costs offered by local TO's and the line utilisation and ISDN call costs.

ISDN also offers compatibility with packet switching (X.25 access compatibility) via a low speed D-channel capability (ideal for Electronic Funds

Transfer and card authorisation) and 64 Kbps via each B channel. Although ISDN is able to do so, not all TO's have made these facilities available. Even when they do, tariff comparisons with existing methods will determine the uptake.

ISDN is integrated into the existing voice (PSTN based) network to the extent that an ISDN telephone can communicate with an analogue telephone. The conversion is done in the TO's exchange or in the local PABX if it provides both analogue and ISDN ports. Interworking with existing equipment is thus assured.

In the UK, ISDN is being used as backup for leased lines, for terminal adaptors (adaptors that convert from existing network interfaces to ISDN) and, increasingly, for LAN bridging. At present it is the only cost-effective and practical service to use for video conferencing, wide area desktop conferencing and multimedia applications.

4 ISDN and LAN Distribution

Having said that the ideal single integrated solution is not yet available, what are the options and issues relating to the provision of ISDN/high speed LAN connection to the desk?

4.1 ISDN over structured cabling

Details of how to deliver ISDN to the Terminal Equipment are to be found in the ISDN Blue Book standards (Volume III – Fascicle III.8 ISDN Overall Network Aspects and Functions, ISDN User-Network Interfaces Rec. I430 Annex A). This describes the 3 primary configurations.

4.1.1 Point to multipoint.

Short passive bus

This configuration allows for the connection of up to 8 TE's (Terminal Equipment) anywhere along the bus. The bus is passive inasmuch as it contains no active components. Although spurs of up to 1M are allowed, their use is not recommended because of the distorting effect of these stubs

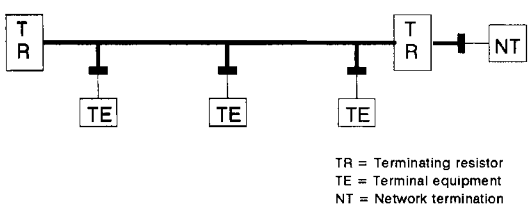

Fig. 3 Short Passive bus

on the signals. The short passive bus is not compatible with structured schemes owing to their point-to-point nature.

Extended passive bus

This configuration takes advantage of the grouping of terminal connection points at the end of the cable far from the NT (Network Terminator). In this configuration the maximum number of TE's is restricted to 4. In this case the TE's are grouped at one end of the cable which can be of up to five times the length of a passive bus.

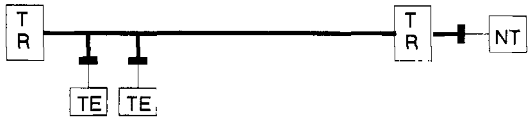

Fig. 4 Extended Passive bus

The arrangement is compatible with structured cabling, if one considers it to consist of lengths of point-to-point cabling with the user end presented as a communications outlet and the other end as a socket in a communications closet. The extended passive bus is then compatible with this configuration if the TE's are considered to be a grouping of ISDN equipment (i.e fax, PC and telephone) on one user's desk. All the items of equipment are connected together at the desk by a short cable bus plugged into the communications outlet. The ISDN Generic Cabling Adaptor (see later in this section) is ideal for this purpose.

The Structured cable limit of 100M is well within that allowed by the relevant ISDN cabling standard (see later this section) for the extended passive bus.

4.1.2 Point-to-Point. This configuration allows a single TE at the end of the cable as in Fig. 5. Once again, this is compatible with structured schemes. In general, the other rules for ISDN wiring are compatible with structured cabling in such respects as:-

- Cable type – EIA/TIA 568 and Addendum 36 cable types are compatible
- TE cord length – EIA/TIA 568 specifies a maximum of 3M which is the maximum allowed for ISDN TE cords. A cord is the length of flylead cable attached to the TE for the purpose of plugging it into the cabling system communications outlet
- Connector types – Both use ISO 8877 (RJ45) types.

Fig. 5 Point to Point

The European standard *'prEN50098-1: Customer premises cabling for information technology Part 1: The design and configuration for ISDN basic access user network interface'* contains useful information as to how to implement these configurations over generic (structured) cabling. This standard introduces the idea of the ISDN Generic Cabling Adaptor.

The main difference between point-to-point based structured schemes and passive bus-based ISDN is the requirement for the termination resistors to be external to the TE (unlike 10BaseT). Terminating resistors must be located at each end of the bus. The NT usually has the option of acting as a terminating end of the bus with its configurable termination resistors. Some means of terminating the other ends outside of the structured cabling is needed. This may take the form of the ISDN Generic Cabling Adaptor (GCA) (see Fig. 6).

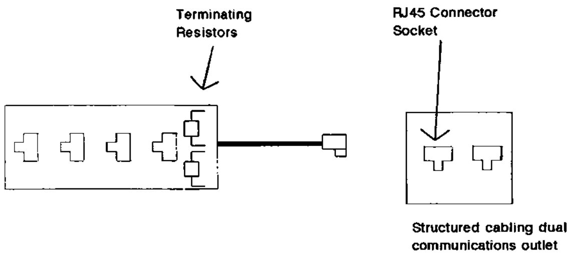

Fig. 6 Generic Cabling Adaptor

The GCA, shown on the left in Fig. 6, can be a single or multi-port device pluggable directly or via a cord into the communications outlet shown on the right in Fig. 6. The GCA itself contains the necessary terminating resistors, thus enabling connection of single or multiple TE devices to the outlet. It can be thought of as analogous to the 4-way distribution boards used for connecting up to 4 domestic appliances to one 13 amp power socket. Unfortunately, few ISDN GCA's have been commercially available, leaving using organisations to make up their own.

4.2 The ISDN hub

The point-to-multipoint and point-to-point configurations described in sections 4.1.1 and 4.1.2 require a separate ISDN network connection for each cable. This means a dedicated connection to each desk which is wasteful and expensive if the required capacity can be shared amongst users. The usual solution to this problem is to instal an ISDN PABX (see section 4.3), which allows users to share ISDN network connections supplied by a TO.

If no local PABX can be financially justified how is it possible to provide a large number of users with access to ISDN over a structured cabling scheme? The structured scheme ostensibly requires a single ISDN point-to-point/extended passive bus per desk; if there is only a single ISDN network connection, then only a single desk can be supplied. This intermediate requirement cannot be solved at present. It calls for an ISDN hub (equivalent to a 10BaseT multiport repeater) that would convert the bus to a star (in a similar manner to the move from thin or thick Ethernet to Ethernet over twisted pair), enabling almost unlimited point to point access to a single Network Terminator.

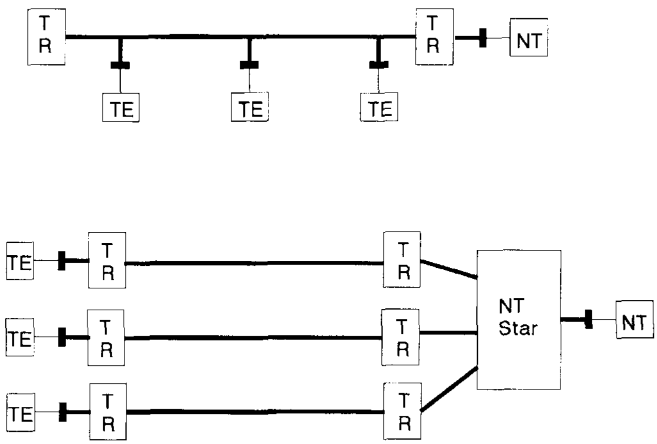

Fig. 7 Passive bus to Starwired conversion via NT Star

Figure 7 shows how the passive bus can be converted to a configuration compatible with Structured Cabling by means of an ISDN hub.

In fact there is a reference to such a device in the ISDN standards (Rec.I430 Annex a). It is called an NT star and enables point-to-multipoint operation using only point-to-point wiring.

There may be a market for such a device but it must be remembered that for all users attached to it, only 2 B channels (and one low speed D channel) can be accessed at any one time. Thus concurrent access would be limited; also, if devices connected to the same hub need to talk to each other they must do so via the local exchange and the call will be charged.

To date the author has seen no evidence of any NT star device commercially available. Realistically, the best solution is to use an ISDN PABX although its likely cost would exceed that of an ISDN hub by a factor of perhaps 5 to 10.

4.3 The ISDN PABX

Once the cabling is ISDN compatible there is the issue of enabling internal and external ISDN access. The ISDN PABX (an ISPBX) is one option. ISPBXs are configurable allowing a mix of ISDN basic and primary rate trunks (as well as other digital services) for wide area access and basic rate and standard analogue ports to the desk as required. The ISDN PABX is shown in Fig. 8.

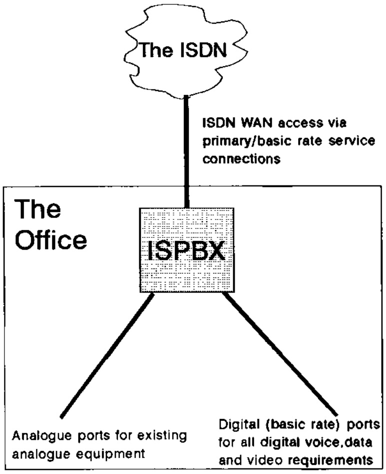

Fig. 8 The ISDN PABX

The pros of this method are:-

- Multiple Basic rate ports available
- Compatibility with existing analogue telephone and datacommunications equipment
- Simple upgrade path to ISDN access for non ISDN equipment

The cons are:

- Incompatibility with industry standard LANs
- Not all offices can justify the purchase of a PABX (key systems can cope with office populations up to about 120 extensions)
- Incompatibilities between voice and data requirements of the PABX i.e difficulties in tuning capacity to meet both requirements in terms of service availability.

4.4 ISDN via LAN

A hybrid solution is possible that maintains the single desk connection but provides the full bandwidth of the existing LAN. This is illustrated in Fig. 9 which shows a terminal enabling combined voice and data communications via the LAN with an ISDN bridge linking to the WAN. Theoretically, the LAN provides ample bandwidth for relatively low speed ISDN connections. In practice, LANs such as Ethernet, which is a contention-based packet network, do not provide good voice service due to variable delays and

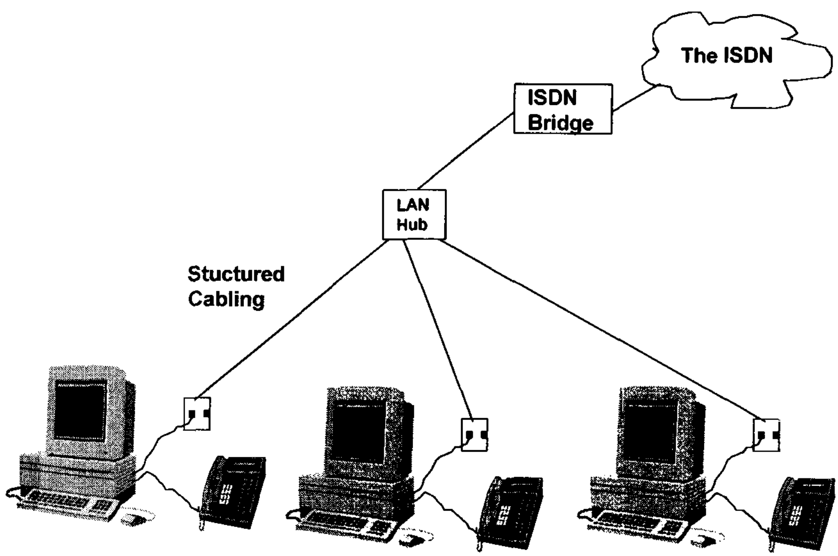

Fig. 9 Voice and data access via LAN

throughput, especially as loading increases. Token ring and FDDI, which are deterministic, are more suited to, but are currently not widely used for these applications. Emerging integrated service LANs such as IEEE 802.6 (Metropolitan Area Networks) or IEEE 802.9 (Integrated Voice and Data LANs) may provide solutions in the long term.

4.5 PABX/LAN hybrids

A variant on the integration of voice and data communications is computer and telephone integration (a term focusing more on the functional integration of computers and telephones) which can take the form of a hybrid network using a different approach. Recently AT&T and Novell linked up to produce the 'Telephony Services for NetWare White Paper'. This describes a proposal to link the capabilities of NetWare LANs with the features and functionality of PABXs. When so linked, the Netware LAN and PABX enable PC users to control phone calls from within an application. Microsoft and Intel have together issued a Windows TAPI (Telephone Applications Programming Interface) that will allow PABX suppliers to enable their PABX's to communicate with PC's in a Windows environment.

A new type of PABX is appearing to support the AT&T/Novell standard and new concepts such as 'Open Telephony' (the telecommunications variant of open systems) are emerging. An example of such a system is in Fig. 10; it shows a typical arrangement retaining both the existing analogue telephone network as well as the LAN connection to the desk. Control of the telephone

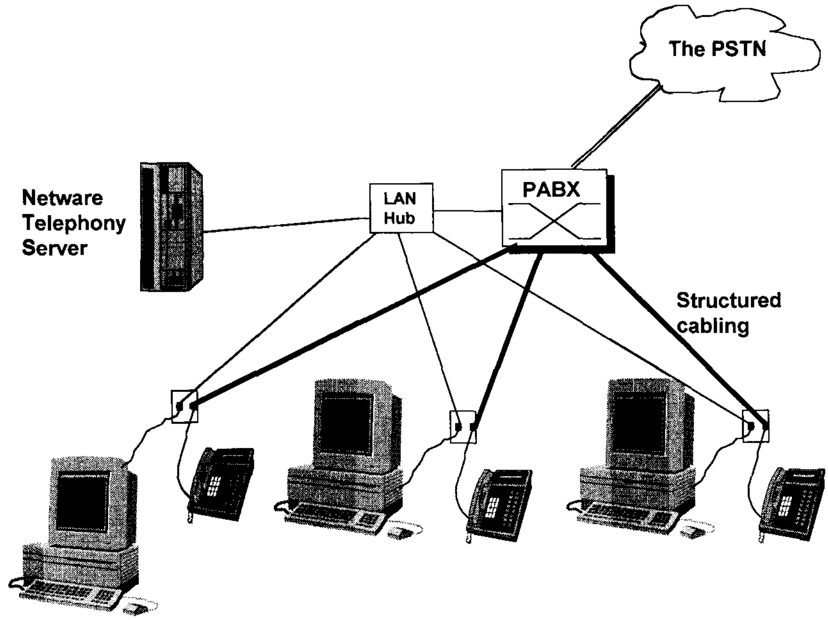

Fig. 10 PABX and LAN hybrid

is passed to the PC application, which in turn controls the telephone over the LAN connection via the Telephony Server to the PABX.

ISDN can be included simply by installing an ISDN bridge between the LAN and the PABX (as in ISDN via LAN) for data applications. Although not a full integration of voice and data services, this scheme is based on a PABX for WAN access, with separate voice and data networks joined at the PABX. If ISDN telephony takes off, the dual network is unnecessary as the telephone control and integration functions can be carried out under direct platform control (ISDN allows this) rather than by this indirect method; the ISDN PABX will then make a re-appearance.

Early implementations of the Telephony Services, such as are offered by Interconnects 3000 series of PABXs, are aimed at voice handling, call handling and desktop productivity applications rather than at integration of voice and data.

This topic is usually referred to as Computer/Telephone Integration and is on the periphery of the scope of this paper.

5 Implementation Issues

Once the important decisions have been made in such areas as topology, technology, flexibility, "futureproofing" and performance, a host of further issues are raised by the very nature of what is being attempted. In fact, it may well be impossible to produce an identical solution globally even if the requirements are the same everywhere. There follows an incomplete list of points/questions to be answered when trying to implement an integrated interconnection architecture.

- ISDN has not been universally implemented; even where it has there are differences in the way that has been done, although initiatives such as EuroISDN are attempting to standardise it.
- Many TO's have not been de-regulated and consequently access to their cabling is not available, nor will the services be provided over cabling owned by customers.
- Safety and regulatory standards, which differ from country to country, must be adhered to when routing different services through a generic cabling system. Even the TO's safety status definition of identical services may differ from one TO to another. (The safety status defines the potential level of hazard from the service).
- Do the cables themselves meet the electrical isolation requirements of all the relevant safety and LAN standards?
- How is the cabling related to the Electromagnetic Compatibility of its environment in the areas of emissions and susceptibility?
- Are all of the services planned to run over structured cabling benign with respect to each other?
- Has the safety status of a service been invalidated by routing it over cabling that presents it in a different form (e.g. an RJ45 connector) to that it was provided in?
- As the boundary between what is a piece of data communications or of telecommunications equipment becomes less distinct, are the associated approvals requirements interfering with each other?

Some of these issues have been resolved, some are being dealt with in various international committees and some are simply not on the agenda.

6 Conclusions

This paper has highlighted the key variables that can determine the implementation of the model in Fig. 1.

These are: ISDN/LAN partitioning; PABX/Hub functionality; Wireless facilities and compatibility.

6.1 ISDN/LAN partitioning

ISDN via LAN seems to meet all requirements insofar as expensive ISDN access is shared over existing LAN connections with the addition of a simple ISDN LAN bridge. This is a good solution for data, providing the LAN does not become choked with ISDN traffic e.g. multiple video conferences;

it is a poor choice for voice until higher speed deterministic LANs are common, or perhaps it may form a good application for switched Ethernet?

A mix of ISDN PABX and an ISDN bridge on the LAN should satisfy most requirements with the choice between each connection depending on the traffic profile for each use. It will require two connections to each desk; this is currently standard for structured cabling systems but what of existing voice requirements? These can usually be met by employing the existing (non-structured) telephone cabling and the old analogue equipment, that is until ISDN telephones become cheap enough to plug straight into an ISDN extended passive bus available at the desk.

6.2 *PABX/Hub functionality*

Confusion reigns while manufacturers of PABXs offer data facilities, hub suppliers offer WAN access and TOs tout Virtual Private Networks (VPN) – that employ public networks to provide the capabilities of private networks.

Add to this the requirements you may have for Computer/Telephone Integration and the simple solution is: "do nothing now".

Realistically, VPN's cannot yet provide LAN services nor do they yet offer basic rate ISDN. Consequently, they cannot support the architecture of Fig. 1. PABX manufacturers are moving towards interworking with the data requirements of their customers as well as enabling some options for Computer/Telephone Integration.

The simple answer is there is no simple solution; it depends on one's requirements. As stated earlier, no single offering provides comprehensive LAN and WAN services although the ISDN PABX coupled to a LAN via a bridge has the best potential. The Intelligent Hub cannot be beaten for high speed LAN access although it may be implemented in the form of several interconnected units rather than a single standalone unit. The author anticipates a major role for both for some time to come, with the size of the role purely application dependent.

6.3 *Wireless facilities and compatibility.*

The requirement for wireless access is still limited and is restricted to specialist environments described in [Flatman/Weiser, 1992]. With PABX manufacturers already offering wireless access and Hub suppliers likely to follow, it is likely that the two sides will converge slowly. In this connection it may be noted that existing wireless LAN products generally come from companies with some military/radio expertise which is not commonly found among hub manufacturers.

Another possibility might be a facility for wireless access to ISDN; it would offer a means of access without clogging up the LAN or the need to buy an

ISDN switch. It would permit continued use of the relatively cheap old analogue telephone system. Hub manufacturers could offer it as a means of supporting enhanced WAN data requirements and PABX manufacturers might see it as a means to upgrade data facilities while retaining the older analogue equipment.

Despite any decisions made on the functionality of PABX and Hub, it is likely that wireless access solutions will come in flavours that suit most applications provided the additional costs can be born.

One final remark: even if the ideal architecture cannot yet be realised, the most important point is to consider communications as a single subject encompassing all forms of data (including voice and video) and to look towards integration as an opportunity to enhance the efficiency and competitiveness of the organisation.

Acknowledgements

The author's thanks are due to A. Flatman, E. Turner and J. Pinkerton for technical assistance, clear thought and supreme patience and understanding.

References

BYRNE, J.A. 1993, The Virtual Corporation. *International Business Week article 8/2/93.*

FLATMAN, A.V. & WEISER, P.G. The Evolution of Wireless Networks *ICL Tech. J.*, Vol. 8, no. 2, pp. 242–271, Nov. 1992.

FLATMAN, A.V. Universal Communications Cabling – A Building Utility. *ICL Tech. J.*, Vol. 6, No. 1, pp. 1117–136, May 1988.

FULLER, A.R. Future Applications of ISDN to Information Technology. *ICL Tech. J.* Vol. 7, No. 3, pp. 501–511, May 1991.

LANE, J. *Exploiting Integrated Voice and Data Networks.* NCC Publications. 1987.

MEERS, J. Broadband Networking. *ICL Tech. J.* Vol. 8, No. 2, pp. 210–224, Nov. 1992.

Biography

U. Baran

After obtaining a degree in Electrical and Electronic Engineering, Uri Baran worked as a hardware design engineer in diverse areas encompassing voice processing, video distribution and communications processing.

He joined ICL Retail Systems in July 1990 specialising in interconnection. Subsequent work has involved the development of a retail specific Structured Cabling scheme and various interconnection projects and consultancy. He is currently a project manager with responsibilities for ISDN/Multimedia technology exploitation and interconnection while also completing a part time MBA.

Parallel Lisp and the Text Translation System METAL on the European Declarative System

Carsten Hammer
Siemens AG, Corporate Research and Development, Dept. ZFE BT SE 43, Otto-Hahn-Ring 6, D-81739 Munich, Germany

Abstract

The paper describes the Lisp programming language subsystem of the European Declarative System EDS. The machine independent model of parallelism which allows easy port of existing sequential programs into parallel form is explained. A major application, the natural language translation system METAL, is characterised and used to demonstrate how large Lisp programs can benefit from the parallelism concepts.

1 Introduction

The European Declarative System EDS is being developed to support large knowledge based systems [Haworth et al. 1990; Skelton et al. 1992]. It comprises a parallel machine [Ward et al. 1990], a parallel relational database, a parallel logic language [Reeve et al. 1993], and a parallel functional language, EDS Lisp.

The parallel Lisp system is described in this paper. Lisp is a powerful general purpose programming language used mostly for AI-based applications. Lisp is a highly interactive language that allows easy prototyping and fast developments. In particular, the arbitrary combination of compiled and interpreted code and the excellent programming environments make up the attraction of Lisp. In EDS Lisp, the advantages of Lisp are combined with the computational power of parallelism.

The next section starts with an overview of the parallel language constructs of EDS Lisp, then deals with the corresponding process-and-store model and shows how the concepts of the EDS operating system EMEX and EDS Lisp are related. Parallelisation of large applications is demonstrated with the knowledge-based language translation system METAL. The section on

METAL explains how METAL works using sample translations from German to English. Several parallelisation strategies for METAL are discussed.

The hardware architecture and the operating system EMEX, in particular the virtually shared memory[1] concept of EDS, are described in [Ward et al. 1990; Borrmann et al. 1990]; therefore we end this introduction with only a short sketch of the necessary concepts.

The EDS prototypes have up to 32 processing elements (PEs) connected via a fast Delta network. The prototypes are extendible to 256 PEs, where the design is open to still larger numbers of PEs. Each PE consists of a 40 MHz SPARC processor with 64 MByte local memory. In addition, there is a second SPARC on each PE supporting system tasks like access to the memory of remote PEs (the so-called System Support Unit SSU). The operating system EMEX (EDS Machine Executive) provides a virtually shared memory such that the distributed store of the EDS machine can be viewed as a single common address space by the applications. EMEX also supports a fast inter-process communication mechanism (IPC).

2 The Parallel Language EDS Lisp

EDS Lisp is a parallel extension of the language Common Lisp [Steele 1984]. Among the many existing Lisp dialects, Common Lisp has been chosen as base language as it constitutes a de-facto Lisp standard. Most existing Lisp applications can thus easily be ported to EDS Lisp. Easy port of existing applications was one of the design goals of EDS Lisp. A major design decision was to add language extensions for explicit parallelism instead of automatic detection of parallelism, because the latter results in finer-grain pieces of parallel execution, whereas the intended target architectures are more suited to larger grain sizes. For the explicit parallelism, we wanted to add only a few constructs which should be powerful enough to cover all programming situations that may arise. This distinguishes EDS Lisp from other parallel Lisp dialects like Spur Lisp [Zorn et al. 1989] that have a philosophy aiming just into the opposite direction: add as many constructs as you like, for instance to experiment with parallelism.

The next sub-section deals with the language extensions chosen for EDS Lisp. The parallel process-and-store model of EDS Lisp is then afterwards described in a separate sub-section. A more extensive discussion of EDS Lisp is given in [Hammer, Henties 1990]. The parallel debugging and visualisation facilities of EDS Lisp are described in [Ilmberger, Wiedemann 1993].

[1] The term 'virtually shared memory' implies virtual memory created and managed by software; no hardware feature is used—Ed.

Parallelism in EDS Lisp is provided by the *future* construct. This construct can also be found in other parallel versions of Lisp, the first one being [Halstead 85]. The *future* is a high-level parallelism construct which involves spawning of parallel processes as well as automatic synchronisation with the result returned from the parallel processes. Furthermore, it fits nicely into the functional programming style of Lisp. *Future* is a function that specifies parallel evaluation of another function *f* with the arguments *arg-1 ... arg-n*[1]:

(*future f arg-1 ... arg-n*)

The function *future* immediately returns an (initially empty) placeholder for the result of *f* and then starts the evaluation of *f* as a separate process (also called child process, *future* process, or *future* for short). The parent process and the child process can now continue in parallel. Many operations in Lisp can be performed with the empty placeholder, like assignment to variables, parameter passing, or construction of lists. Only if the parent process needs to access the result of the *future* process does it wait until the result is available and then continue operation. Both the placeholder mechanism and the implicit waiting (i.e. process synchronisation) are invisible to the programmer.

The following example highlights the placeholder mechanism. In the standard Common Lisp assignment

(*setq* x (*translate-to-french "This is an example"*))

the result of *translate-to-french* is assigned to the variable *x* (*setq* means *assign-to* in Lisp). The assignment has to wait until the result is computed. In EDS Lisp, the assignment

(*setq x* (*future #'translate-to-french "This is an example"*))[2]

spawns a parallel process and assigns an empty placeholder for the result of *translate-to-french* to *x*. The program that executed *setq* can now continue (e.g. call other functions or spawn other translations) while the function *translate-to-french* is being computed in parallel. *x* can immediately be used in further assignments like

(*setq z x*)

without accessing the actual contents of *x*; only the (reference to the) placeholder of the parallel process executing *translate-to-french* is copied to *z*.

The implicit result synchronisation comes into effect if the actual result of a future call is accessed, as in

(*print x*)

---

[1] In Lisp, the left parenthesis is written *before* the function name, in contrast to conventional (procedural) languages where it is written *after* the function name.
[2] #' is needed here to ensure that translate-to-french is passed as function.

Here, EDS Lisp tests if x contains a placeholder, and if the corresponding process is finished. If the process is not finished, the process executing the *print* is suspended until x (respectively the result of the associated process) becomes available. It should be noted that an indefinite number of processes can be waiting for the placeholder (in this example all those processes accessing x and z).

Around the *future*, there are a few supporting constructs like one to test if a *future* is finished (to avoid implicit waiting if further work can be done in the meantime) and killing of *futures* (i.e. abandonment of processes).

In addition to the implicit result synchronisation, EDS Lisp also provides an explicit synchronisation mechanism: (*wait x*) waits for the completion of x and then returns the result of x as the result of the function *wait*. For example, in

(*setq y* (*wait x*))

one can be sure that y does *not* contain a placeholder of an unfinished process.

Another synchronisation construct are exclusive functions (so-called x-functions in EDS Lisp). X-functions implement the concept of **Critical Regions**. X-functions can only be executed by one other function at a time. If there are more functions calling an X-function, then they have to wait until the first one has left the x-function; then the next one can start to execute the x-function.

Finally, mailboxes for explicit inter-process communication are provided in EDS Lisp. The programmer can create empty mailboxes with *make-mailbox* (and assign the created object for later use to a variable) like in

(*setq m make-mailbox*)

and then send and receive messages with

(*send message m*)
(*receive m*)

where *receive* returns the first message from the mailbox *m*. *Receive* is a blocking operation, which means that it is suspended if there is no message; as soon as a message arrives, *receive* is resumed. Mailboxes are a many-to-many communication mechanism: many processes may send to one mailbox, and many – probably others – may receive from this mailbox. Mailboxes are a communication construct that can be implemented very efficiently on most parallel architectures.

2.2 Process and store model of EDS Lisp

The *process model* of EDS Lisp assumes an indefinite number of parallel processes; in particular, the number of processes can be bigger than the number of available processors on the machine. Each *future* can be viewed as a separate process on language level, so the terms *future* and *process* can be used synonymously at language level. The distribution of parallel processes to physical processors (PEs) on the EDS machine is done by the EDS Lisp run time system and is completely transparent to the Lisp programmer. If there are more processes than processors, more than one process will be mapped onto one PE.

The *store model* of EDS Lisp supports data sharing between parent and child processes (inheritance) as well as local variables proprietary to individual processes. A child process can access data from its parent via parameter passing. Moreover, it can access the lexical and dynamic variables defined in its parents' environment. This implies that a process (i.e. a *future*) can access its parent's data even if they execute on different processing elements. If shared variables are used, the programmer is responsible for the correct parallel access to these variables, and the above-mentioned language constructs can be taken for synchronisation. Of course, there is only a need for explicit synchronisation if the functional style of Lisp is left.

In general, a language has to define which situations lead to defined results and which do not, if shared variables are accessed from parallel processes. Trying to get defined results where it may not be necessary leads to over-synchronisation and thus to poor performance. The store model of EDS Lisp is thus defined in a way that allows a weak coherent implementation [Borrmann et al. 1990] of the virtually shared memory used by EDS Lisp. The weak coherency mechanism allows storage inconsistencies between synchronisation points and yields better performance than a strong coherency mechanism that always has to keep the storage coherent.

An inconsistency situation that is only possible in a weak coherency environment is depicted in fig. 1. In process 1, the variables x and y are set to 1

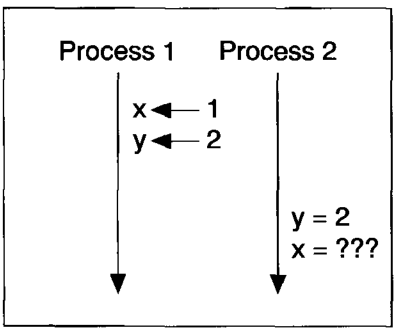

Fig. 1 Weak Coherency Effect

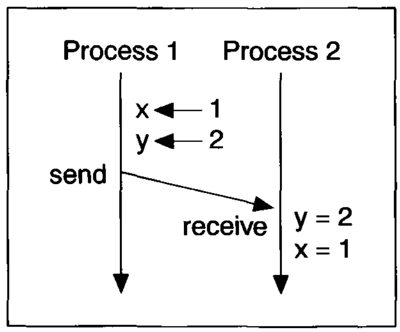

Fig. 2 Synchronised Situation

and 2 respectively, in this order. In process 2, there is no statement possible about the value of the variable x, even if the later assigned value of y has been tested to be 2. The user has to synchronise between the events to get deterministic results. In figure 2 the synchronisation is done via a message;

synchronisation makes the storage of both processes coherent, and both x and y have their correct values. All implicit or explicit synchronisations of EDS Lisp make the store between the synchronised processes coherent, like spawning a process, getting the result from a process, using *wait*, an x-function, or messages. The above-mentioned inheritance of data between parent and child process is in fact a consequence that spawning is seen as a synchronisation: the parents' and the child's storage are coherent at the moment of spawning.

The above example is quite strange, and it is unique to weak coherence. The "normal" undefined case is that two processes write unsynchronously to the same global variable. Here the value of the global variable is undefined, whatever the model of parallelism is. So synchronisation is needed in any case if shared variables are allowed. In an EDS Lisp program where accesses to shared variables are properly synchronised, there is absolutely no difference in behaviour between a shared store, a virtually shared memory with strong coherency, and a virtually shared memory with weak coherency; except that weak coherency results in better performance than strong coherency. A more detailed discussion about the storage model of EDS Lisp and the weak coherency mechanism can be found in [Hammer, Henties 1991].

2.3 *Interaction of the concepts of EDS Lisp and the EDS machine*

The concepts of EDS Lisp and of the EMEX operating system and the EDS machine have influenced each other. The EDS Lisp system runs as an EMEX task that has a team on each PE belonging to the EDS Lisp program. In each team (i.e. on each processor) there is a managing thread which does run time management like process distribution and scheduling; in addition there is a set of threads that execute the *futures*.

Communication and synchronisation in the EDS Lisp run-time system is done via the Inter-Process Communication (IPC) of EMEX, this holds for tasks internal to the run-time system as well as for execution of EDS Lisp's mailboxes and the x-functions. The virtually shared memory of EMEX supports the shared store model of EDS Lisp.

The virtually shared memory mechanism is also used to move code and data to newly created processes on other processors. The Lisp run-time system need not care about the actual copying of code and data to other processors; only the start address and some managing information of a process are sent via IPC to another processor which then tries to start the code at this address. If the code is not present, the virtually shared memory mechanism gets an interrupt and fetches a copy of the corresponding code from the node where it resides. The same holds for data and other code accessed by the Lisp program. Later accesses to this code/these data are then fast local accesses. Fetching code from other PEs is done by the second processor on each PE, the System Support Unit SSU. The Lisp program can continue executing another *future* in the meantime.

The language definition of EDS Lisp has been so chosen that the efficiency of the weak coherency scheme can be fully utilised. This means that in most cases the EDS Lisp system need not wait for acknowledgement of coherency messages. The coherency messages can again be processed by the SSUs in parallel to the Lisp program execution. For debugging purposes, EDS Lisp can be switched to a strong coherency model which is also supported by EMEX.

3 Parallel Text Translation: a Commercial Application

3.1 The METAL system

To test the EDS Lisp system, we needed a large commercial application written in Common Lisp. We have chosen the METAL[3] system [Gajek 91; Thurmair 91] which is a Common Lisp application developed at Siemens Nixdorf Information Systems (SNI). METAL targets the translation of technical documents, in particular where large amounts of text have to be translated. The language pairs German/English, English/German, German/Spanish, French/Dutch, and Dutch/French are already commercially available; five other language pairs are under development. For EDS Lisp, the language pair German/English has been selected as a test application. The basic language dictionaries contain about 20.000 word stems for each language; in addition there are application-specific lexicons, such as lexicons for computer science or chemistry, which can easily be extended by the user to satisfy his needs exactly.

The importance of such an application is underlined by the Common Market in Europe where a further and rising demand for translations of technical documents into a variety of European languages is expected. Consider that in a typical technical translation application, the entire documentation for a complete technical product line often amounts to several tens of thousands of pages. For an export-oriented industry, support of an automatic translation system is absolutely necessary. The METAL translation system will not replace highly qualified human translators, but it will relieve them of routine work and improve their productivity.

The quality of the METAL translation depends strongly on the complexity of the input sentences. As the targeted area of METAL is technical documentation, the input sentences are normally comparatively simple, and the resulting translations are quite understandable and need only little postprocessing. Examples of METAL translations will be given in the next section.

METAL is computationally extremely intensive and needs large amounts of main memory for the extensive dictionaries. Depending on the complexity of the sentences, METAL translates with a speed from about one word

[3] Machine Translation and Analysis of Natural Language.

per second to several seconds per word on a 40MHz SPARC processor. METAL is thus ideally suited to exploit the parallelism of the EDS machine to improve performance.

3.2 How METAL works

For a complete translation METAL has a series of tasks to solve. First of all, METAL has to determine the word stems of single words, i.e. to do the morphological analysis. Part of this is the segmentation of compounds into their constituents (see fig. 3). The morphological analysis involves intensive searching in the source language dictionaries, evaluation of different hits and selection of the most probable solution.

German word	**Constituents**
Vektorcodegeneratoren	Vektor (noun sterm), Code (noun sterm), en (plural ending)

Fig. 3 Morphological Analysis

The next step is to construct the derivation tree of the processed sentence. Tree construction starts with small subtrees which will then be combined to larger derivation trees until the complete sentence structure is finished. Each of the words from the dictionary has attached to it a series of syntactic and semantic attributes that are used to guide and evaluate the construction of different derivations, and, finally, to select the most likely one. A part of this process is the anaphora resolution which deals with words that refer back

German sentence	**English sentence**
Die *Personen, die* Daten erfassen	the *persons who* gather data
Die *Maschinen, die* Daten erfassen	the *machines which* gather data

Fig. 4 Anaphora Resolutions (Context Dependent Translations)

to other previously used words. Figure 4 shows an example, where the German relative pronoun "die" has to be translated differently depending on the noun it refers to. This syntactic and semantic analysis is the most time consuming part of METAL.

The third major part of the translation process is the transformation of the sentence structure from the source language into the target language. This may involve a complete restructuring of the sentence, as figure 5 shows. After this, the word stems of the target language are selected from the target language dictionary, transformed into their correct syntactic form, and

German sentence:	Häufig werden diese Schleifen zur Bearbeitung verketteter Listen eingesetzt.
Word-for-word translation:	Frequently are these loops for processing linked lists used.
Correct English word order:	These loops are frequently used for processing of linked lists.
METAL translation:	These loops are used frequently for processing of linked list.

Fig. 5 Structure Transformation

output according to the transformed derivation tree. Figure 6 shows the translation of some sentences taken from operating instructions for an intermediate EDS Lisp version.

METAL works on a sentence-by-sentence basis, that is, it does not take context between different sentences into account. The context resolution inside a sentence is already complex enough to lead to extensive computations, and taking wider context into account would increase the computational demand even further without significantly improving the result.

3.3 The parallel METAL version

Talking about parallelisation of METAL means, on the one hand parallel execution of code for better performance and, on the other hand, the distribution of the large dictionaries to the different PEs.

For the parallel execution of the METAL code, there is a series of possible strategies. The main differences between these strategies lie in the granularity of the parallel processes and in the possible parallelism. Several of the following approaches can of course be combined.

Parallelisation via pipelining of translation phases. Each of the translation phases (i.e. morphological analysis, syntactic analysis, structural transformation, generation of target sentence) can be generated as a separate process. These processes can then be organised as a pipeline where the result of one phase is fed as input into the next phase. This approach leads only to low parallelism; in addition, the time behaviour of the pipeline is dominated by the slowest phase.

Parallelisation of morphological analysis. There can be several alternatives for the segmentation of a given word into its components. Each segmentation in turn can lead to further possible segmentations. All these alternatives can be evaluated in parallel similar to most divide-and-conquer algorithms.

Dieses Dokument beschreibt die wichtigsten Eigenschaften und die Bedienung des EDS-Lisp-Systems. Das auf einer Sun3 mit dem Motorola-Prozessor 68030 ablaufende EDS-Lisp V0 1R04 besteht aus zwei Teilen:

- Das Speicherabbild VM enthält die Funktionen des Lisp-Systems, die für diese Version integriert worden sind (siehe Kapitel 5).
- Das Ladeprogramm 'lisp' dient zur Aktivierung des EDS-Lisp-Speicherabbildes.

Des Aufruf von EDS-Lisp erfolgt auf der Sun3 durch Aktivierung des Ladeprogramms aus dem Kommando-Interpreter. Als Parameter erwartet das Ladeprogramm die Angabe einer VM-Datei.
Nach dem Laden und Initialisieren des VMs wird vom EDS-Lisp-System 'EDS>' als Prompt ausgegeben. Das EDS-Lisp-System kann durch Aufruf der Lisp-Funktion 'quit' verlassen werden.

Original German Text

This document describes the most important characteristics{quality} and the operation{service} of the EDS-Lisp system. The EDS-Lisp V0.1R04 running on a Sun3 with a Motorola processor 68030 consists of two parts:

- The VM storage limit contains the functions of the LISP system which have been integrated for this version (see chapter 5).
- The loader 'LISP' is used for activation of the EDS-Lisp storage image.

The call of EDS-Lisp occurs through activation of the loader from the command interpreter on the Sun3. The loader expects the information {specification} of an VM file as parameters.
As a prompt,' EDS>' is output after loading {shop} and initializing the VMs by the EDS-Lisp system. The EDS-Lisp system can be left by call of the LISP function 'quit'.

METAL Translation

This document describes the most important characteristics and how to operate the EDS-Lisp system. EDS-Lisp V0.1R04 runs on a Sun3 with a Motorola processor 68030 consists of two parts:

- The storage image VM contains the functions of the Lisp system which have been integrated for this version (see chapter 5).
- The loader 'lisp' is used for activation of the EDS-Lisp storage image.

EDS-Lisp is called by activation of the loader from the command interpreter on the Sun3. The loader expects a VM file as a parameter.
As a prompt, 'EDS>' is output by the EDS-Lisp sytem after loading and initializing of the VM.
The EDS-Lisp system can be left by a call of the Lisp function 'quit'.

Translation After Minimal Human Postprocessing

Fig. 6 Excerpt From Operating Instructions for an Intermediate EDS Lisp Version

Parallelisation of syntactic analysis. As mentioned above, the parser constructs small pieces of the sentence subtrees which in turn will be combined to form larger subtrees. Many of these subtrees can be generated independently or partly independently of each other leading to possible parallel processes. The parallelism can be introduced at a fine grain level where the small subtrees are constructed, it can also be introduced at a higher grain size where different interpretations at the sentence level have to be evaluated. The major disadvantage of this approach is that the parser needs to be rewritten significantly for this kind of parallelism which contrasts with the easy port to EDS demanded.

Parallelisation at the sentence level. METAL operates by translating each sentence independently. This leads to a straightforward parallelisation: The translation of each sentence is spawned as parallel process. The principle can be driven even further, if a whole paragraph or a complete page or chapter is spawned as separate process. This approach does not speed up the translation of a single sentence, but it does for the translation of large amounts of text.

The last approach has been chosen for parallel METAL on EDS Lisp because of its simplicity and performance. The METAL system needs to be modified only slightly, and the large grain size of this approach fits in nicely with the distributed architecture of the EDS machine thus leading to a good performance.

As already indicated, the parallelisation of METAL has also to deal with the distribution of dictionaries. It would be a bad choice to have the complete dictionary on each processing element (PE). The available main memory on each PE is not big enough to hold entire lexicons and the working space for the translations. Thus the lexicon needed to be paged on disk and the remaining working space for the translation algorithms would be small, resulting in frequent garbage collections. These effects slow down translation; in fact this is a large part of the computation time needed on sequential workstations, which normally do not have enough main memory to hold the entire METAL system.

To avoid this situation it is necessary to hold only a part of the dictionary on each PE. Here we can benefit from an effect that only a small part of the dictionaries is in fact needed for a specific translation: about 100 structural words already cover 50% of an average text, a basic vocabulary of 2000 words covers 85% of an average text, and with an extended basic vocabulary of further 2500 words 95% of an average text are covered [Klett 77]. The situation may be even better for technical texts, where the vocabulary is more limited than in the quoted "average texts".

Following this effect, one approach could be to distribute the lexicons evenly over the processors such that each processor holds only a fraction of the complete dictionary. The shared memory model of EDS Lisp assures that all words can be accessed, even if they are not locally available. In this case, the virtually shared memory model of the EDS operating system EMEX copies the missing words to the PEs where they are needed; all further accesses to these words are then fast local accesses.

For the current version of parallel METAL, we have chosen an even more straightforward approach. The METAL system, including the dictionaries, is loaded on one PE only. When parallel execution starts, the necessary parts of the dictionaries are copied by the virtually shared memory mechanism to the PEs where they are needed. This implies a comparatively high network traffic from one PE to the others at the beginning of the translations. This situation will only persist for a short time; as soon as the most frequently used words have accumulated on the different PEs, the network traffic will become very low.

Taking now the above-mentioned ideas into account, the parallelisation of METAL is quite simple. The translation of each sentence is spawned as a parallel process by a *future*. The load-balancing mechanism of EDS Lisp distributes the execution of the *futures* to different PEs of the EDS machine. The virtually shared memory mechanism of the EDS machine moves the necessary (i.e. the accessed) parts of the translation algorithms and dictionaries to the PEs where they are needed.

In addition, the *future* is used in a way that ensures that the sentences translated will be collected in the correct order; there is no need to label the sentences and sort the independently translated sentences after they are returned from the different processing elements.

The way to port METAL to the EDS machine given above gives an idealised picture. The real situation shows that there is more to do. The main point was to convert the METAL system to standard Common Lisp. This was in fact the most expensive part, because the original METAL implementation uses machine dependent extensions of Symbolics Common Lisp for performance reasons. Another problem that must not be underestimated is the possible use of global variables. A use of a global variable may be quite normal in the sequential case; in a parallel environment this may lead to unpredictable effects. These variables need to be encapsulated in the translate-function in such a way that they become private variables of the different processes.

The actual port is then done as described in the last section: Only a sentence distribution loop needs to be added; neither the actual distribution of processes nor the distribution of the dictionaries need concern the application programmer.

Because the communication overhead becomes very low (after a short initial phase) and the parallelism is large grain, the speedup of parallel METAL is expected to scale almost linearly with the number of the processors.

Performance measurements to prove (or disprove) this are underway. The performance of the parallel METAL system need not be used only for raw speedup; instead more advanced and resource-intensive algorithms can be supported by EDS. As a result, better translation quality is envisaged.

4 Summary

The paper has described the parallelism concepts of EDS Lisp, the parallel Lisp system of the European Declarative System. The concepts comprise only a few parallel language constructs that are sufficient to express any parallel problem. EDS Lisp assumes an indefinite number of processes and uses a shared store view of the distributed EDS machine.

As major test application, the text translation system METAL, is being ported to the EDS machine. The process-and-store model of EDS Lisp, the virtually shared memory model of the operating system EMEX and a straightforward parallel model of METAL work excellently together to allow an easy port of METAL to the EDS machine and to result in high performance.

In general, computationally intensive Lisp programs can easily be converted to parallel programs. The programmer inserts *future*-calls for program parts to be executed in parallel. Load distribution, data movement, and result synchronisation are done by the underlying Lisp system. Additional complexity is only necessary if parallel accesses to global variables need to be synchronised. As bugs resulting from bad synchronisations are difficult to spot, EDS Lisp contains an elaborate set of parallel debugging and visualisation tools.

References

BORRMANN, L., ISTAVRINOS, P. "Store Coherency in a Parallel Distributed Memory Machine." *Distributed Memory Computing, 2nd European Conference, EDMCC2, Munich, Germany, April 1991, Proceedings*, A. Bode (ed), LNCS 487, Springer-Verlag 1991.

GAJEK O. "The Metal System." *Comm. of the ACM, Vol. 34, No. 9, September* 1991.

HAMMER, C., HENTIES, T. "Parallel Lisp for a Distributed Memory Machine." Ashok Gupta (ed.), *Proceedings of Lisp Europal Workshop "High Performance and Parallel Computing in Lisp", November 1990*, Twickenham, England, published by EUROPAL, Surrey, UK, 1990.

HAMMER, C., HENTIES, T. "Using a Weak Coherency Model for a Parallel Lisp." *Distributed Memory Computing, 2nd European Conference, EDMCC2, Munich, Germany, April 1991, Proceedings*, A. Bode (ed), LNCS 487, Springer-Verlag 1991, 42–51.

HALSTEAD, R., "Multilisp: A Language for Concurrent Symbolic Computation." *ACM Transactions on Programming Languages and Systems*, October 1985.

HAWORTH, G., LEUNIG, S., HAMMER, C., REEVE, M. "The European Declarative System, Database, and Languages." *IEEE Micro*, December 1990.

ILMBERGER, H., WIEDEMANN, C.-P. "A Visualization and Control Environment for Parallel Program Debugging." *Parallel and Distributed Systems: From Theory to Practice, 26th Hawaii International Conference on System Sciences*, HICSS-26, Hawaii, 5–8 Jan 1993.

Grund- und Aufbauwortschatz Englisch. Ernst Klett Verlag, Stuttgart, 1977.

SKELTON, C.J., HAMMER, C., LOPEZ, M., REEVE, M.J., TOWNSEND, P., WONG, K.F. EDS: "A Parallel Computer System for Advanced Information Processing." "*Conference on Parallel Architectures and* Languages Europe, Parl 92, Proceedings, Paris, June 1992.
STEELE, G. "Common LISP: The Language." *Digital Press*, 1984.
THURMAIR, G. "Metal: Computer Integrated Translation." *Proceedings of the SALT Workshop* 1991, Manchester.
WARD, M., TOWNSEND, P., WATZLAWIK, G. "EDS Hardware Architecture." *Joint Conference on Vector and Parallel Processing, Zürich*, September 1990.
ZORN, B., HO, K., LARUS, J., SEMENZATO, L., HILFINGER, P. "Multiprocessing Extensions in Spur Lisp." *IEEE Software, July* 1989, pp. 41–49.

Biography

Carsten Hammer

Carsten Hammer received a Diploma in Computer Science and his Doctorate from the Technical University of Braunschweig in Germany. He works in Siemens Corporate Research and Development in Munich where he has project responsibilities for the parallel EDS Lisp system. He has been manager of the Parallel Software Group which involved projects on automatic vectorisation and parallelisation, parallel debugging and visualisation tools, and integration of parallel and object oriented techniques. He is now head of the Simulation Technology Group.

Detecting Latent Sector Faults in SCSI Disks[1,2]

Hannu H. Kari and Heikki Saikkonen
Department of Computer Science, Helsinki University of Technology Espoo, SF-02150, Finland
Fabrizio Lombardi
Department of Computer Science, Texas A&M University College Station, TX 77843, USA

Abstract

This paper presents new improved methods for detecting latent sector faults in a disk subsystem as caused by media deterioration of the disk magnetic storage material. Usually, sectors in a disk are accessed using uneven patterns causing some of the sectors to be accessed only seldom. In case of media deterioration on the rarely accessed sectors, a latent disk fault may remain undetected for a long time. To detect latent sector faults, a disk is scanned through periodically. In this paper, an adaptive algorithm is proposed to utilize the idle time of the disk for scanning commonly used disks that comply with SCSI-II interface standards.

1 Introduction

In a storage system, faults in disk units are usually divided into two categories: sector and disk faults. A sector fault is mainly due to media deterioration of the disk unit and the probability of a sector fault in a disk is significantly larger than a disk fault.

Generally, the storage system can handle faults by using redundant data storage, such as *redundant arrays of inexpensive disks* (RAID) [5, 8, 11, 14]. The redundant disk storage is then used to preserve the integrity of the system after a sector or a disk fault [8, 12, 14]. The disadvantage of this approach is that the fault is generally encountered only when the faulty area

[1]This research is supported by the Finnish Cultural Fund and the Foundation of Jenny and Antti Wihuri.

is requested by a user. Besides an extended response time for a user request, this will increase the probability of data loss due to latent media faults [6].

In this paper, a novel approach for detecting latent sector faults of commonly used SCSI disks is proposed. Detection is accomplished using a separate process that utilizes the idle time of the disk subsystem. Hence, it is possible to detect and repair sector faults before they are encountered by user requests. The scanning process utilizes features of modern disk units (statistics information and sector fault recovery with a reassign block command) [1, 2].

The organization of this paper is as follows. Chapter 2 introduces briefly previous studies. Chapter 3 discusses methods for repairing deteriorated magnetic media. The proposed adaptive scanning algorithms are discussed in Chapter 4. In Chapter 5, the effects of the proposed methods on the disk subsystem performance are analyzed. Finally, conclusions are presented in Chapter 6.

2 Previous Studies

Disk arrays are widely studied in the technical literature [5, 8, 10, 11]. Several disk array organizations have been introduced to improve either system performance or reliability (or both) [3, 9, 11]. A disk subsystem is mainly optimized for storing and retrieving data during the normal operation of the system and not for obtaining the best performance with the degraded level disk subsystem.

Typically, modern disk arrays are not prepared to detect media deterioration in advance, but they rely on the redundant disk subsystem [11, 12, 14]. When a disk request encounters a sector or a disk fault, data is recovered using the redundant information and then the faulty storage is repaired. This may cause a significant delay on user requests as the retry and repair processes may take a long time.

Latent faults can be detected using algorithms like *memory scrubbing* [13]. An example of disk scanning algorithms has been presented in [7]. The main problem with these types of algorithm is that it is unable to adapt to the user load. This results in difficulties for highly loaded systems if the system load exceeds the estimated maximum. A further disadvantage of the non-adaptive method is that it is not fully able to utilize the uneven load of the system (especially the low system load periods).

3 Media Deterioration and Recovery

A sector fault (denoted usually as a *bad sector*) occurs when the magnetic medium fails to keep the appropriate storage due to medium degradation. Usually, the minimum area affected by the fault is a sector [1, 2, 4].

A modern disk unit is already arranged for handling sector faults [1, 2]. The current SCSI standards specify only the *reassign block* command format (to initiate a sector repair process) but not the actual repair process itself (i.e., it is up to the disk manufacturer to specify how the spare sectors are used).

A modern disk unit gathers various information about the disk behavior during its normal operation [2]. Typical statistics collected in these systems are the number of read and write operations. Also, the disk collects statistics for problems (events) that have occurred.

The new SCSI-II standard specifies more statistics information than the earlier SCSI standard [1, 2]. Especially, the detailed information about the last encountered event is useful for preventing and detecting faults in a disk subsystem.

The main problem with the enhanced statistics is the masking effect. As the standard specifies the detailed information only for the latest event (masking any previous event), some of the faults may remain undetected. In practical implementations, the mean time between events is significantly longer than the average interval of polling the statistics. Hence, it is possible to catch most of the faults with a detailed information and therefore other counters are not needed (as proposed in [7]).

4 Adaptive Scanning Algorithms

The proposed scanning methods are based on the previous study [7] of its enhanced Algorithm 2. This algorithm can be improved by adjusting the scanning interval based on the disk activity. The size of the request block is usually kept constant as it is optimized with respect to rotation delays. Both methods proposed in this paper use the same scanning algorithm together with different scanning intervals.

The main procedure for scanning a disk is performed using relatively small check regions. After scanning a check region, the statistics information is read. A smaller check region allows a better fault detection in case of multiple disk faults. A check region is divided into smaller blocks (*scanning requests*) that are read sequentially one by one. The size of the scanning requests is kept to the same order as the user requests to minimize delays for user disk requests.

Beside the scanning process, the proposed approaches utilize also a timer-based process. This process checks the disk subsystem activity at a regular interval. Based on the current instantaneous disk activity and the activity history, the system activity is calculated. The system activity is then used for determining proper parameters in the scanning algorithm.

The proposed scanning algorithm is as follows.

Algorithm 1:

1. Reset the initial parameters (waiting time *wt*, request size *rs*, start address *sa*, and size of check region *cr*).
2. Reset the start address offset, $sao = 0$.
3. Wait for a fixed amount of time given by *wt*.
4. If there are no user requests in the disk (in process or waiting in a queue), go to step 7.
5. Call function *adjust (wt)*.
6. Go to step 3.
7. Call function *adjust (wt)*.
8. Issue a disk read request at location $sa + sao$ with size *rs*.
9. When the read request has been completed, increment *sao* by *rs*.
10. If $sao < cr$, go to step 3.
11. Read the extended disk statistics information.
12. If no faults were encountered, increment *sa* by *sao*, and go to step 2.
13. Test the potentially faulty address (*pfa*) that was reported by the statistics.
14. If the sector in *pfa* is faulty (or the test reads cause more retries), start the *reassign block* command to repair that sector.
15. If $pfa < sa$ or $pfa > (sa + sao)$, go to step 2.
16. Increment *sa* by *sao*, and go to step 2.

The scanning parameters are adjusted using the function *adjust (wt)* and the calculation of the system activity. The system activity in Method 1 is given by

$$A(t_i) = \sum_{k=i-N+1}^{i} \frac{a_k}{N} = A(t_{i-1}) + \frac{a_i}{N} - \frac{a_{i-N}}{N} \tag{1}$$

where a_i is the instantaneous activity (either 0 or 1) at time i and N is the number of samples used in calculating the activity. For Method 2, the system activity is given by

$$A(t_i) = A(t_{i-1}) \times (1 - p) + p \times a_i \tag{2}$$

where p is the history factor. The history factor p specifies how strongly the most recent measurements affect the calculation of the activity. The benefits of the second method are a faster adjustment on changes and no need for storing $N + 1$ samples of instantaneous activity. Based on the activity calculation, the *adjust* function is given by

$$wt = A(t_i)^2 \times (wt_{\max} - wt_{\min}) + wt_{\min} \tag{3}$$

where $wt_{\min}$ is the minimum wait time for the scanning algorithm (to limit the maximum system load increase caused by the scanning process) and $wt_{\max}$ is the maximum wait time for the scanning algorithm (to limit the maximum scanning time).

In equation (3), the quadratic term of the activity is used to compensate the increase of the queue length with a higher system load. If a linear function ($wt = A(t_i) \times (wt_{\max} - wt_{\min}) + wt_{\min}$) were used, the wait time would increase too much even with a moderate disk load ($0.25 < \rho < 0.75$) and a significant part of the scanning capacity is wasted.

An important feature of the proposed methods is to react more quickly for an increase in load than for a decrease. If it is slow to react to a decrease in load, the only effect is that some idle time is lost (that could have been utilized better in scanning the disk). This increases slightly the time to scan the entire disk while decreasing the capability of utilizing very short idle intervals efficiently. On the other hand, if the scanning algorithm is not fast enough by decreasing its activity when the disk load increases, user disk requests may be significantly affected (user disk requests can suffer longer response times than neccessary). In this case, the effect of the proposed scanning method can be worse than in [7].

5 Effects of the Algorithms

The effects of the adaptive scanning algorithm can be measured using two alternative approaches. First, the effect on the user requests (increase in queue length) can be studied as a function of the alternative scanning methods and their parameters. Second, the length of the scanning process can be estimated based on the system load and proposed methods.

The proposed scanning methods are analyzed using the system load illustrated in Figure 1. This represents two scenarios of system behavior in which the activity of user disk requests changes significantly. The first step corresponds to the case when the system load is increased radically. The second step corresponds to the case when the load drops back to a low level.

5.1 Effect on the queue length

In [7], a previous scanning algorithm has shown to increase the disk load moderately even in a heavily loaded system when the queue length of a disk is studied (the increase in queue length due to the scanning algorithm is less than 0.5 even when the disk is utilized by a user for 90%). However, the highest increase is found under a heavy load when the system performance is the most critical. On the other hand, when the system load is low (and the disk could perform more scanning activity without endangering the response time requirements) the activity of the scanning algorithm is also low.

As presented in [7], the average increase in queue length is given by

$$\rho_{\text{inc}} = \frac{n \times t_{alg} + t_{chk}}{n \times t_{alg} + t_{chk} + \dfrac{n \times t_{\text{wait}}}{1 - \rho}} \tag{4}$$

where ρ is the utilization caused by user requests, t_{alg} and t_{chk} are the average processing times for a disk and a statistics read request made by Algorithm 1, respectively, n is the number of iterations before checking the statistics information (equal to cr/rs), and t_{wait} is the wait time. Because the proposed scanning algorithm issues a new disk request only when the disk is idle, the maximum increase in queue length is one. In these calculations, it is assumed that $t_{alg} = t_{chk} = 0.1\text{s}$. n is 20. $wt_{\min} = 2\text{s}$, and $wt_{\max} = 64\text{s}$.

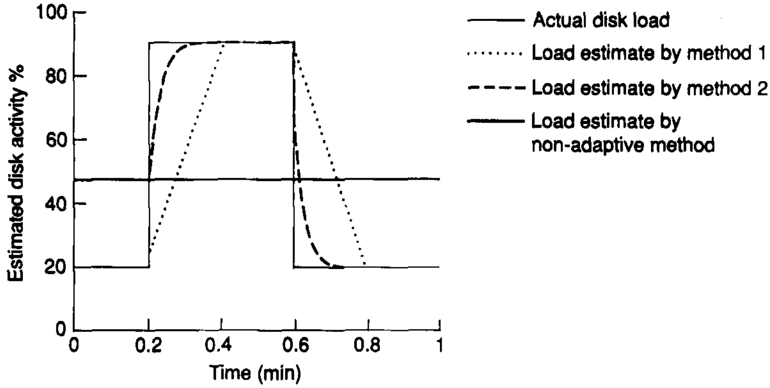

Fig. 1 System load estimate as function of time

Figure 1 depicts the estimate of system load according to Equations (1) and (2). Methods 1 and 2 are also compared with the algorithm proposed in [7] (where the scanning algorithm is adjusted based on the average load). Method 1 significantly lags in detecting the increase in system load. Respectively, the decrease of system load is not recognized very quickly. On the contrary, Method 2 provides a faster response for the change in disk load.

Both proposed methods are analyzed by comparing the results with the non-adaptive scanning algorithm [7]. Method 1 is analyzed by varying the number of samples (N) while Method 2 is analyzed by varying the history factor (p).

Figure 2 presents a comparison between the non-adaptive [7] and the adaptive (Method 1 for different values of N) scanning algorithms as measured by the average increase in queue length. In general with Method 1, the increase in queue length is higher (lower) when the system load is low (high) than with non-adaptive scanning algorithm. As the number of samples (N) increases, the peak of the longer queue length grows because the change of the disk load is detected too slowly. Similarly, with larger N, it takes a longer time before the scanning algorithm starts to expedite the scanning process when the system load has dropped.

Figure 3 presents a comparison between the non-adaptive [7] and the adaptive (Method 2 for different values of p) scanning algorithms as meas-

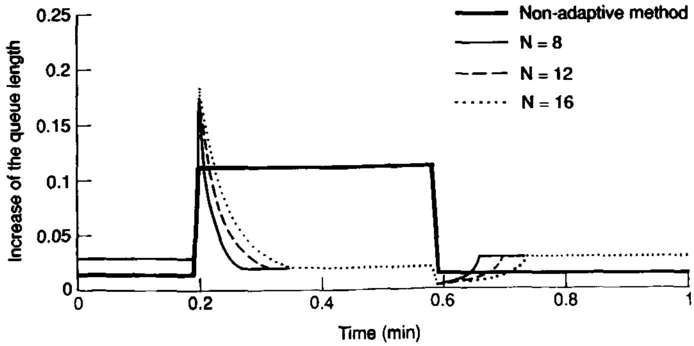

Fig. 2 Effect of the scanning method 1

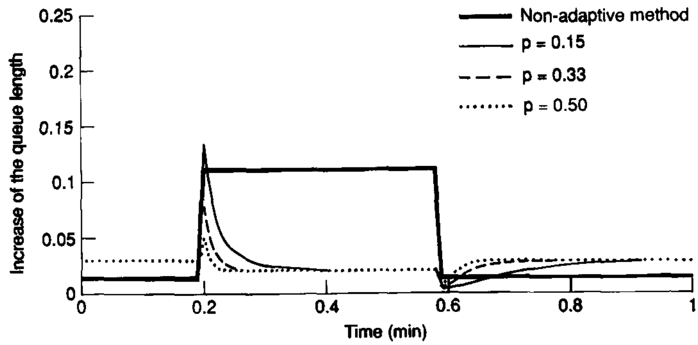

Fig. 3 Effect of the scanning method 2

ured by the average increase in queue length. In general with Method 2, the increase in queue length is higher (lower) when the system load is low (high) than with non-adaptive scanning algorithm. The larger the value of p is, the faster the adaptive behavior to the changing system load is. When the system load increases rapidly, the peak in queue length is as high as with the non-adaptive method but the peak decreases quickly as p increases. When the system load drops, the scanning algorithm reactivates itself more quickly for a larger value of p.

5.2 *Effect on the scanning time*

Based on the system load profile of Figure 1, relative and absolute scanning times of the alternative scanning methods are presented in Table 1. Both proposed methods perform better than the original, non-adaptive scanning algorithm presented in [7]. More than 37% of the scanning time can be saved and in the best case the scanning time can be reduced almost by half.

Table 1 Relative and absolute scanning times of the alternative scanning algorithms for a 256 MB disk

Method	Non-adaptive method (7)	Method 1			Method 2		
Parameters	wt = 10s	$N=8$	$N=12$	$N=16$	$p=0.15$	$p=0.33$	$p=0.50$
Relative scanning time	100%	57%	60%	63%	62%	56%	54%
Absolute scanning time	40.7 h	23.1 h	24.3 h	25.5 h	25.6 h	22.8 h	22.1 h

6 Conclusions

In this paper, methods to detect latent sector faults and deteriorated storage media have been studied. The basic principle for better fault detection consists of scanning the entire disk sequentially by using the idle time of the disk unit. This process utilizes the statistics information that SCSI disks gather. By initiating a disk request only when the disk is otherwise idle, the effect on user disk requests is minimized.

When a non-adaptive scanning algorithm is used, the disk is scanned using the same parameters regardless of the system load. This may lead to an extended response time especially when the disk load is high. Despite only a minor increase of response time, this may not be suitable for environments with strict response time requirements. Also, the non-adaptive scanning algorithms are not able to fully utilize the uneven disk utilization and especially long idle periods.

In this paper, the scanning algorithm is improved by adjusting the scanning parameters based on the disk activity. The higher disk activity is generated by a user, the longer wait time is used for the scanning algorithm. Thus, the effect of the scanning algorithm in a heavily loaded system is even further decreased while allowing the full utilization of the longer idle periods of the disk (such as at night time).

Two different methods for adjusting the scanning parameters have been discussed. They are based on calculating the average load of the disk from the recent activity. Both proposed methods decrease the scanning time of the disk (in the presented example, the scanning time is decreased by 37% to 46%). When the system load rapidly increases, the first proposed scanning method results in a higher peak for the increase of queue length than the non-adaptive scanning algorithm. On the other hand, the second method has a much smaller peak and, by setting properly the parameters, it provides a lower peak than the original non-adaptive scanning algorithm.

The maximum number of additional disk requests, that any user request may need to wait, is only one. Hence, the delay for a user request does not increase significantly. Especially, the maximum response time increases only slightly. Thus, this method can be used also in systems that have strict response time requirements (e.g., in real-time systems or with multi-media applications) to improve reliability and decrease the extended response times due to sector faults.

References

1 *"ANSI Standard X3,131. Small Computers Systems Interface (SCSI)"*. American National Standards Institute (ANSI), New York, 1986.

2 *"ANSI Standard X3,131. Small Computers Systems Interface-2 (SCSI-2)"*. American National Standards Institute (ANSI), New York, 199x. Draft.

3 CHERVENAK, A.L., *"Performance Measurements of the First RAID Prototype"*. University of California TR UCB/CSD 90/574. Berkeley, Master's Thesis. May 1990.

4 *"HP 97556/58/60 5.25-inch SCSI Disk Drives. Technical Reference Manual"*. Hewlett-Packard, manual part No. 5960–0115, 1991.

5 GIBSON, G.A., *"Redundant Disk Arrays: Reliable, Parallel Secondary Storage"*. University of California TR UCB/CSD 90/615, Berkeley, PhD Dissertation, March 1991.

6 KARI, H.H., LOMBARDI, F. *"On the Analysis of Repair Algorithms for Mirrored Disk Systems"*. Department of Computer Science, Texas A&M University, p. 27, May 1992.

7 KARI, H.H., SAIKKONEN, H., LOMBARDI, F. "On the Methods to Detect Latent Sector Faults of a Disk Subsystem". *1993 SCS Western Multiconference on Computer Simulation.* MASCOTS '93, January, pp. 317–322, 1993.

8 KATZ, R.H., OUSTERHOUT, J.K., PATTERSON, D.A., CHEN, P., CHERVENAK, A., DREWES, R., GIBSON, G., LEE, E., LUTZ, K., MILLER, E., ROSENBLUM, M.: "A Project on High Performance I/O subsystem". *Computer Architecture News,* pp. 24–31, August 1989.

9 LEE, E.K. *"Software and Performance Issues in the Implementation of a RAID Prototype"*. University of California TR UCB/CSD 90/573, Berkeley, Master's Thesis, May 1990.

10 O'BRIEN, J. "RAID 7 Architecture Features Asynchronous Data Transfers". *Computer Technology Review, Spring* 1992, pp. 43–47.

11 PATTERSON, D.A., GIBSON, G., KATZ, R.H. "Introduction to Redundant Arrays of Inexpensive Disks (RAID)". *IEEE Proc. COMPCON,* Spring 1989.

12 REDDY A.L.N., BANERJEE, P. "A Study of Parallel Disk Organizations". *Computer Architecture News,* August pp. 40–47, 1989.

13 SALEH, A.M., SERRAND, J.J., PATE, J.H.: "Reliability of Scrubbing Recovery-Techniques for Memory Systems", *IEEE Trans. of Reliability, vol. 39, No. 1.* pp. 114–122, April 1990.

14 SCHULZE, M., GIBSON, G., KATZ, R. PATTERSON, D.: "How Reliable is a RAID?". *Proc. IEEE Spring Comp. Conf., San Francisco,* February 1989.

Book Review

BS 7649. Guide to the Design and Preparation of Documentation for Users of Application Software (BSI, 1993, pp. 104 price Members £32.40 Non-members £64.80.)

"If all else fails, read the instructions." We all do it. Somehow we expect to be able to drive cars, or video recorders, or complex software, with minimal intellectual effort. And we all hate manuals.

Now here to help us is British Standard 7649, commissioned by BSI from ICL Enterprises. Perhaps it contains a cure?

Well, there is good news, neutral news, and bad news.

The neutral news – neither good nor bad – falls into two areas.

Firstly, scope. This document is not what I in my innocence would have expected a British Standard to be. It does not prescribe. It does not lay down tests for conformance. It does not even suggest that conformance would guarantee good documentation. It is, as its title clearly says, a compendium of advice, on how the documentation process should be carried out and how the physical appearance of documentation can be optimized. It has rather little to say about documentation content.

Secondly, point of view. It is very much a document by authors for authors. The ICL Press Release proclaims it to be the result of extensive consultation with developers, authors and consultants. It doesn't say anything about consultation with users. The Guide covers paper documentation but not on-screen explanations and help, possibly because these are usually produced by developers rather than authors. There is even an occasional note of authorial plaintiveness ("authors should be ... equal members in the (development) team").

The good news is that within these limitations this is an admirable document. Its praiseworthy features include:

- emphasis on the distinction between awareness documentation, tutorial documentation, and reference documentation – this is fundamental,
- heavy stress on clarity as the primary virtue (and the Guide itself is very clear),

- a recommendation that authors should themselves become users of the software before they start writing,
- some suggestions on how the usability of documentation might be measured.

Any documentation produced in the spirit of this Guide (conforming to the letter may be too much to ask) will be a thoughtful, professional, information-rich job.

The minor bad news is what I would regard as a number of sins of omission:

- as implied above, the manual-lovers have had their comprehensive say – and it is comprehensive, if anything to a fault – but the manual – haters' views are not addressed;
- there is just one, hardly visible, suggestion of involving real users (as distinct from authors as surrogate users) in the documentation process. The Guide does recommend user feedback, but that is a much feebler instrument;
- there is little discussion of the documentation of user organizations' in-house systems. It is hard to deny that these ought to be just as well documented as widely marketed packages, but this is a counsel of perfection. Overambition is self-defeating. Realistic advice for this imperfect world would have been welcome;
- the Guide assumes the standard software life cycle. It has almost nothing to say about documenting "light duty" systems (those which for whatever reason do not justify major documentation effort) or "fuzzy systems" (where the requirement is not clear-cut or not stable, and the system constantly evolves). To be fair, the software world as a whole has not really got its mind round these two issues either.

The major bad news is that BS 7649, however well it may be taken on board by applications software producers, is not going to solve the problem of our hating manuals. At best it will improve it around the edges.

Why do we shy away from manuals? The question is a real and serious one. The word processing package I use comes with what seem to me perfectly adequate tutorial and reference manuals – why then are there 18 different books about this package in my local bookshop, covering (to judge by their titles) the same ground as the manuals, at prices up to £27.45?

The answer is that *we think* we know manuals are boring. People's expectation of books is that they should compel interest (except reference books – but even there the great Dr Johnson's Dictionary includes some celebrated jokes). My word processing package tutorial manual is interesting – provided that the reader has a genuine and lively wish to learn to use the package.

It does little to meet any other reader half-way. Fat books look boring even when they are not.

Compare school text-books: they used to be a by-word for boredom, but in recent times many have been livened up without loss of content. User documentation, particularly in the awareness and tutorial categories, needs the same treatment. Clarity is not enough.

BS 7649 seems not to recognize this. For example it advises against jokes. I can see why (jokes may miscarry, are hard to translate, can annoy), but I think it is wrong. It recommends writing for the most naive user. Again I can see why, but the result will be to bore everyone else. Yes, I know this line of thought leads to the unwelcome conclusion that separate documentation for different levels of user sophistication is needed – but the same is after all true of text-books.

My advice, then, to software managers controlling "heavy duty" projects is: "Read BS 7649. Make your authors read it too. But then tell them to rank reader interest alongside clarity; to get out of the developers' private world from time to time to involve real live users in what they are doing; and, when faced with a choice between comprehensiveness and entertainment value, to go for entertainment every time (leaving the comprehensiveness for the reference book)."

John Aris

Pages contained in each issue

(1) 1-177	(3) 349-566
(2) 179-348	(4) 557-680

Subject index
Volume 8

Author index
Volume 8

ICL TECHNICAL JOURNAL

Guidance for Authors

1. CONTENT
The *ICL Technical Journal* has a large international circulation. It publishes papers of high standard having some relevance to ICL's business, aimed at the general technical community and in particular at ICL's users and customers. It is intended for readers who have an interest in the information technology field in general but who may not be informed on the aspect covered by a particular paper. To be acceptable, papers on more specialised aspects of design or applications must include some suitable introductory material or reference.

The Journal will usually not 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 matter relating to unannounced products of any of the ICL Group companies.

Letters to the Editor and reviews may also be published.

2. AUTHORS
Within the framework defined by §1 the Editor will be happy to consider a paper by any author or group of authors, whether or not an employee of a company in the ICL Group. All papers are judged on their merit, irrespective of origin.

3. LENGTH
There is no fixed upper or lower limit, but a useful working range is 4000–6000 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 to this

4. ABSTRACTS
All papers should have an Abstract of not more than 200 words, suitable for the various abstracting journals to use without alteration. The Editor will arrange for each Abstract to be translated into French and German, for publication together with the English original.

5. PRESENTATION
5.1 *Printed (typed) copy*
Two copies of the manuscript, typed $1\frac{1}{2}/2$ spaced on one side only of A4 paper, with right and left margins of at least 2.5 cms, and 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.

5.2 *Disk version*
Authors are strongly urged to submit a magnetic disk version of their copy in addition to the manuscript. The Editor will be glad to provide detailed advice on the format of the text on the disk.

5.3 *Diagrams*
Line diagrams will if necessary be redrawn and professionally lettered for publication, so it is essential that they 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 have a caption and be numbered for reference in the text, and the text marked to show where each should be placed – e.g. "Figure 5 here". Authors should check that all diagrams are actually referred to in the text and that all diagrams referred to are supplied. Since diagrams are always separated from their text in the production process these should be presented each on a separate sheet and, *most important*, each sheet must carry the author's name and the title of the paper. The diagram captions and numbers should be listed on a separate sheet which also should give the author's name and the title of the paper.

5.4 *Tables*
As with diagrams, these should all be given captions and reference numbers; adequate row and column headings should be given, also the relevant units for all the quantities tabulated. Short tables can be given in the text but long tables are better submitted on separate sheets and these, as for diagrams, must carry the author's name and the title of the paper.

5.5 *Photographs*
Black-and-white photographs can be reproduced provided they are of good enough quality; they should be included only very sparingly. Colour reproduction involves an extra and expensive process and will be agreed to only exceptionally.

5.6 *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, and only text references; references to works not quoted in the text should be listed under a heading such as "Bibliography" or "Further reading".

5.7 *Style*
A note is available from the Editor summarising the main points of style – punctuation, spelling, use of initials and acronyms etc. – preferred for Journal papers.

6. 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, as for example to conform to the Journal's general style for spelling or notation, will be made by the Editor.

7. PROOFS, OFFPRINTS
Printed proofs are sent to authors for correction before publication. Authors receive 25 offprints of their papers, free of charge, and further copies can be purchased; an order form for copies is sent with the proofs.

8. COPYRIGHT
Copyright in papers published in the *ICL Technical Journal* rests with ICL unless specifically agreed otherwise before publication. Publications may be reproduced with the Editor's permission, which will normally be granted, and with due acknowledgement.