Management Summary

BUTLER COX FOUNDATION

# The Future of System Development Tools

BUTLER COX FOUNDATION

# The Future of System Development Tools

Management Summary
Report 74, March 1990

Butler Cox plc

LONDON
AMSTERDAM MUNICH PARIS

Published by Butler Cox plc
Butler Cox House
12 Bloomsbury Square
London WC1A 2LL
England



**Availability of reports**
Members of the Butler Cox Foundation receive three copies of each report upon publication; additional copies and copies of earlier reports may be purchased by members from Butler Cox.

Photoset and printed in Great Britain by Flexiprint Ltd., Lancing, Sussex.

# The Future of System Development Tools

# Management Summary

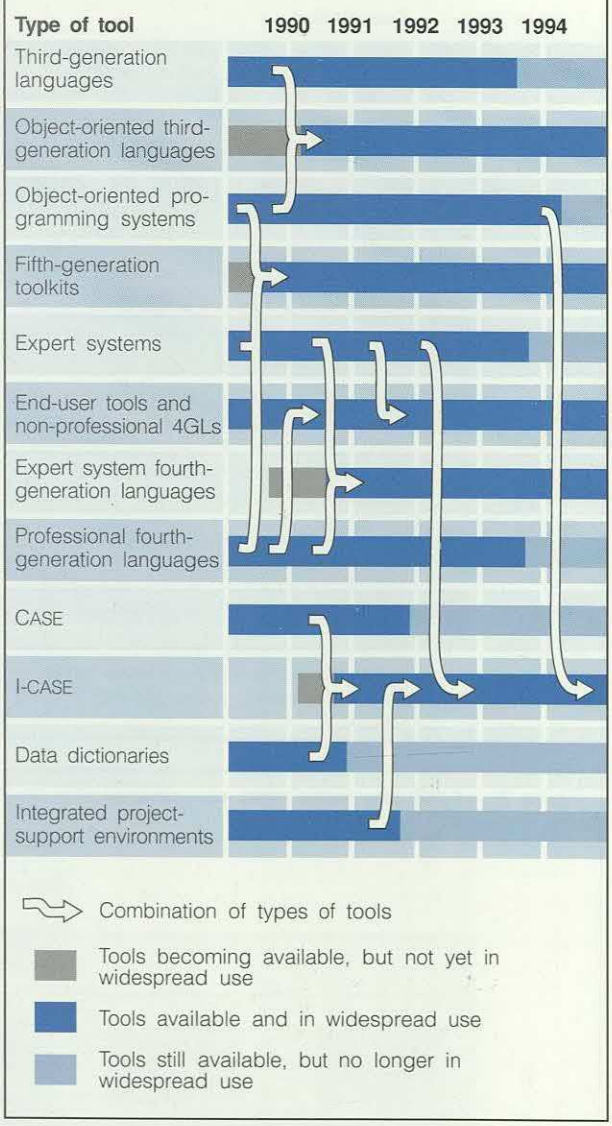
*This document summarises the main management messages from Foundation Report 74, published in March 1990. The full report is available to members of the Butler Cox Foundation.*

Suppliers of the system development tools now beginning to appear in the market promise that their products will finally solve the problems that have beset system development departments for the past few decades. As in the past, not all of these promises will be realised. While the new tools undoubtedly have the potential to make the department's task much easier in the longer term, most of them are still at an early stage of development, and in the short term, the promises should therefore be treated with caution. This does not mean, however, that the tools should be ignored. There are benefits to be gained if they are adopted selectively and with regard to the future.

There is great uncertainty about the form that future tools might take, about their impact on the development process, and about the most appropriate timing for their introduction. As Figure 1 shows, the situation is complex, and will become more so in the mid-1990s. The purpose of the report is to guide systems directors through this maze so that they can exploit the tools to provide the flexible and reliable service that the business demands, without precluding the possibility of adopting others as it becomes feasible and sensible to do so.

Some of the terminology used to describe the tools may be unfamiliar. A selected glossary is given in Figure 2, overleaf.

**Figure 1** Over the next five years, existing types of development tools will combine to form new ones




## Do not commit to I-CASE yet

When fully implemented, an I-CASE (integrated computer-aided software engineering) development environment will provide a compatible and integrated set of tools that covers the entire applications life cycle. At present, no complete I-CASE environment exists. The promise is, however, that I-CASE will improve development productivity and quality, will allow more sophisticated applications to be developed, and will reduce dependence on individual tool suppliers. It will also provide a means of analysing, supporting, and maintaining applications that were originally developed outside the I-CASE development environment.

As Figure 3 overleaf shows, there are three basic elements to an I-CASE development environment. The I-CASE *system development tool*

*set* will provide the tools that are required across the entire applications life cycle. The I-CASE *framework* will provide both a standard interface to which all the tools must conform, and a means of supporting the project team at every stage of development via a common user interface. The I-CASE *data dictionary* will be the store for all the development information, throughout the life cycle of the application.

**Figure 2 A wide variety of development tools will be available in the 1990s**

| Type of tool | Description |
|---|---|
| CASE (computer-aided software engineering) | Tools that automate systems development activities. Examples include analyst work-benches and application and system generators. |
| I-CASE (integrated CASE) | A set of tools that provide integrated and semi-automated support for the full applications life cycle. |
| Reverse-engineering | Tools that allow existing applications to be rebuilt in a structured and consistent manner. |
| Fourth-generation language | Programming tools that use a subset of natural language to specify 'what' the computer is to do, rather than 'how' to do it. |
| Rule-based | Development tools based on expert-system (or knowledge base) techniques, or that allow expert systems to be developed. (An expert system is an application based on knowledge or know-how.) |
| Object-oriented | Tools for constructing applications from software 'objects' — structures that include both data and programs, and which usually represent objects in the real world. |
| Multimedia | Tools for constructing applications where the information is stored in different forms — data, text, graphics, audio, and even moving images. |

Two trends are currently evident. Several of the major hardware suppliers have recognised the importance of I-CASE and have announced their commitment to it. Rather than compete with the established CASE suppliers, however, they have promised to provide a data dictionary and a 'standard' framework in which to integrate the CASE tools that are available today. Software suppliers, on the other hand, are developing tools that will 'fit into' an I-CASE development environment or develop into full I-CASE products.

A complete I-CASE development environment is unlikely to be available until the first quarter of 1992 at the earliest. The major difficulty at present is the lack of consistency in the various standards applicable to I-CASE, and there is no indication of which ones will eventually prevail. What currently exists are fragmented pieces of an I-CASE environment that will be integrated over the next two years or so, and linked with others, such as planning tools, testing tools, and so on, to provide an integrated environment. Furthermore, I-CASE will require a change in the traditional view of the applications life cycle (see Figure 4). Today's development methods are geared to the traditional life cycle. Many of them will need to be enhanced to make them suitable for the I-CASE life cycle.

Our advice is that the time is now right for planning to migrate towards I-CASE — not yet

**Figure 3 I-CASE has three main components**

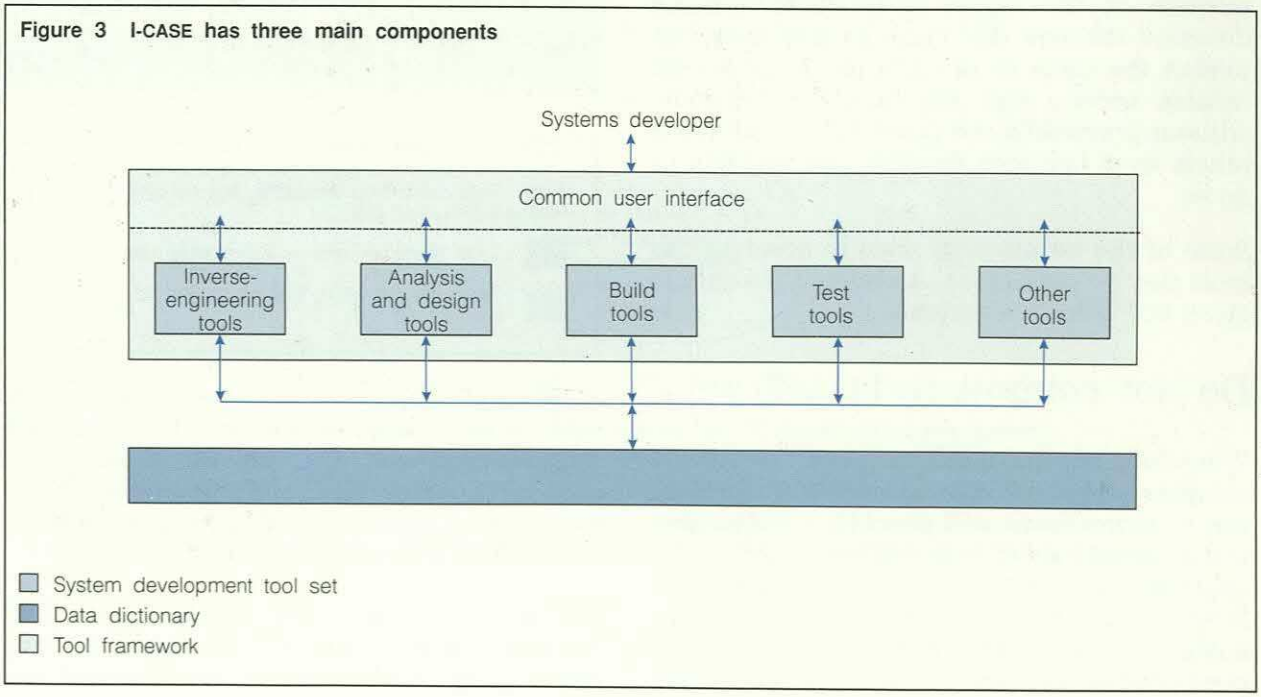
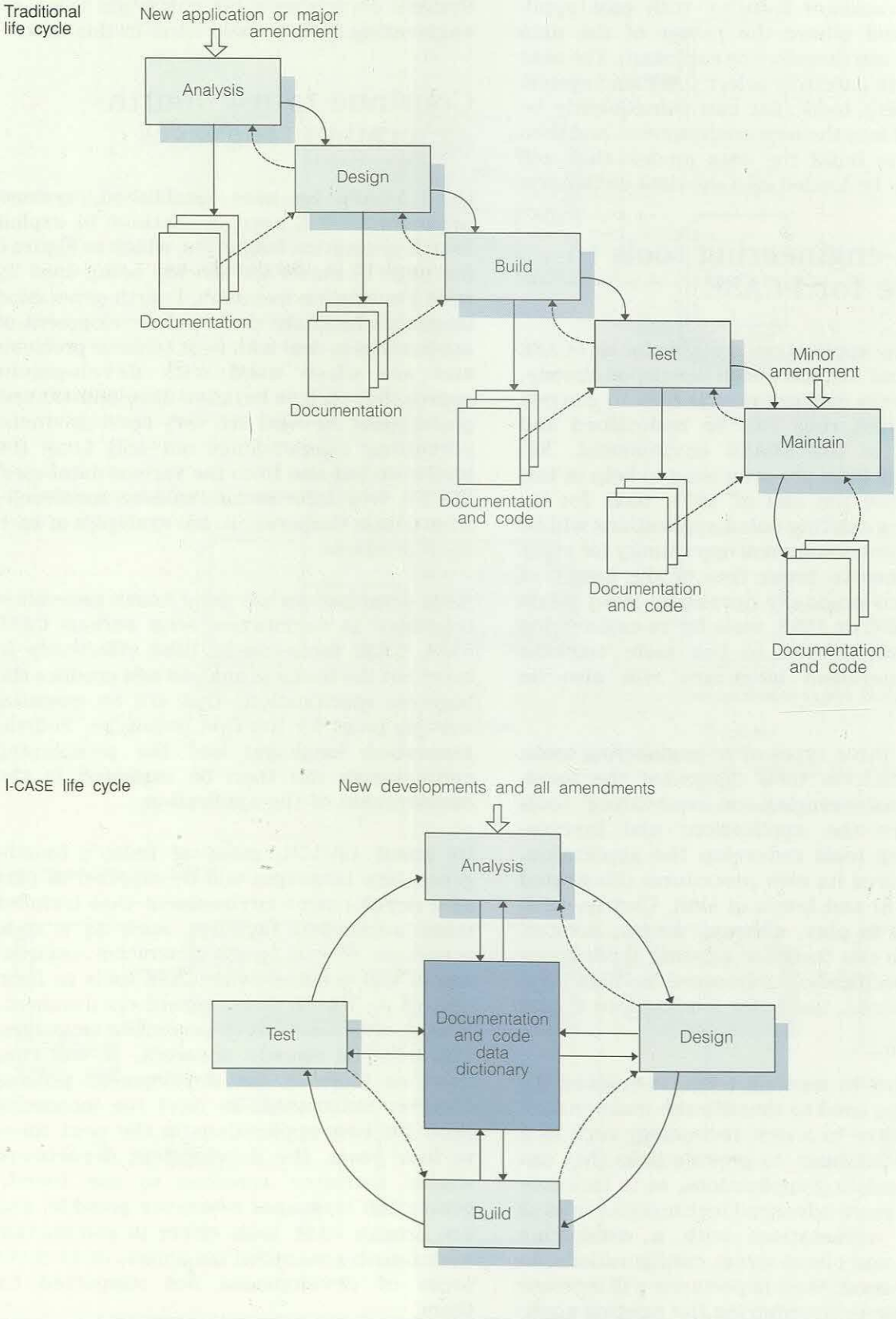


**Figure 4** **The development life cycle in an I-CASE environment differs markedly from the life cycle that is common today**



making a full commitment to it, but taking care not to create barriers to its subsequent adoption. The first step is to identify where I-CASE will be most beneficial (probably for large corporate or departmental applications, which will tend to share common features with past applications, and where the power of the data dictionary can therefore be exploited). The next steps are to carefully select CASE and system development tools that can subsequently be integrated into the new environment, and then to begin to build the data models that will eventually be loaded into the data dictionary.

## Use re-engineering tools to prepare for I-CASE

Many of the applications suitable for an I-CASE environment will have been developed already, and the main concern now is how to convert them so that they can be maintained and enhanced in the I-CASE environment. Re-engineering tools are promising to help in this area. Before the end of 1991, tools for re-engineering existing Cobol applications will be available, providing a real opportunity for many organisations to break free of the legacy of applications originally developed many years ago. By 1992 or 1993, tools for re-engineering applications written in the more common fourth-generation languages will also be available.

There are three types of re-engineering tools: redocumentation tools document the application; restructuring or renovating tools restructure the application; and inverse-engineering tools redevelop the application. Each requires its own procedures (illustrated in Figure 5) and levels of skill. Clearly, each has a role to play, although as yet, none of these tools can translate existing applications into a specification expressed in high-level business terms, ready for loading into a data dictionary.

In the short to medium term, re-engineering tools can be used to simplify the maintenance task, to move to a new technology such as a relational database, to provide links that can integrate existing applications, or to facilitate the use of more advanced technologies such as powerful workstations with a windowing interface, and client-server configurations. In the longer term, their importance will increase as their role in transferring the existing applications portfolio to an I-CASE environment is recognised. The most time-consuming task will be building the data model(s) for the existing (and often fragmented) databases, ready for loading into the I-CASE data dictionary. Systems departments can now begin to use re-engineering tools to help them in this task.

## Continue to use fourth-generation languages

Until I-CASE becomes established, systems departments will need to continue to exploit fourth-generation languages, which as Figure 6 (on page 6) shows are already being used by most Foundation members. Fourth-generation languages facilitate the rapid development of applications to deal with most business problems and are often used with development approaches such as iterative development and prototyping. Several are very open products, permitting independence not only from the hardware but also from the various databases. FOCUS, from Information Builders, and Oracle, from Oracle Corporation, are examples of such open products.

Some organisations are using fourth-generation languages in conjunction with various CASE tools. CASE tools can be used effectively to carry out the business analysis and produce the business specifications that are an essential starting point for the first prototype. Fourth-generation languages and the prototyping methodology can then be exploited in the development of the application.

By about 1993/94, many of today's fourth-generation languages will be supplied as part of a development environment that includes other automated facilities, such as a code generator. These fourth-generation environments will combine with CASE tools to form part of an I-CASE development environment. Indeed, with some fourth-generation languages, this trend is already apparent. It will continue as more of the development process becomes automated. To meet the increasing need for new applications in the next three to four years, the development department should therefore continue to use fourth-generation languages whenever possible, and use proven CASE tools either in conjunction with fourth-generation languages, or to cover areas of development not supported by them.

**Figure 5 The different types of re-engineering tools require different procedures, different levels of support, and different technical knowledge**

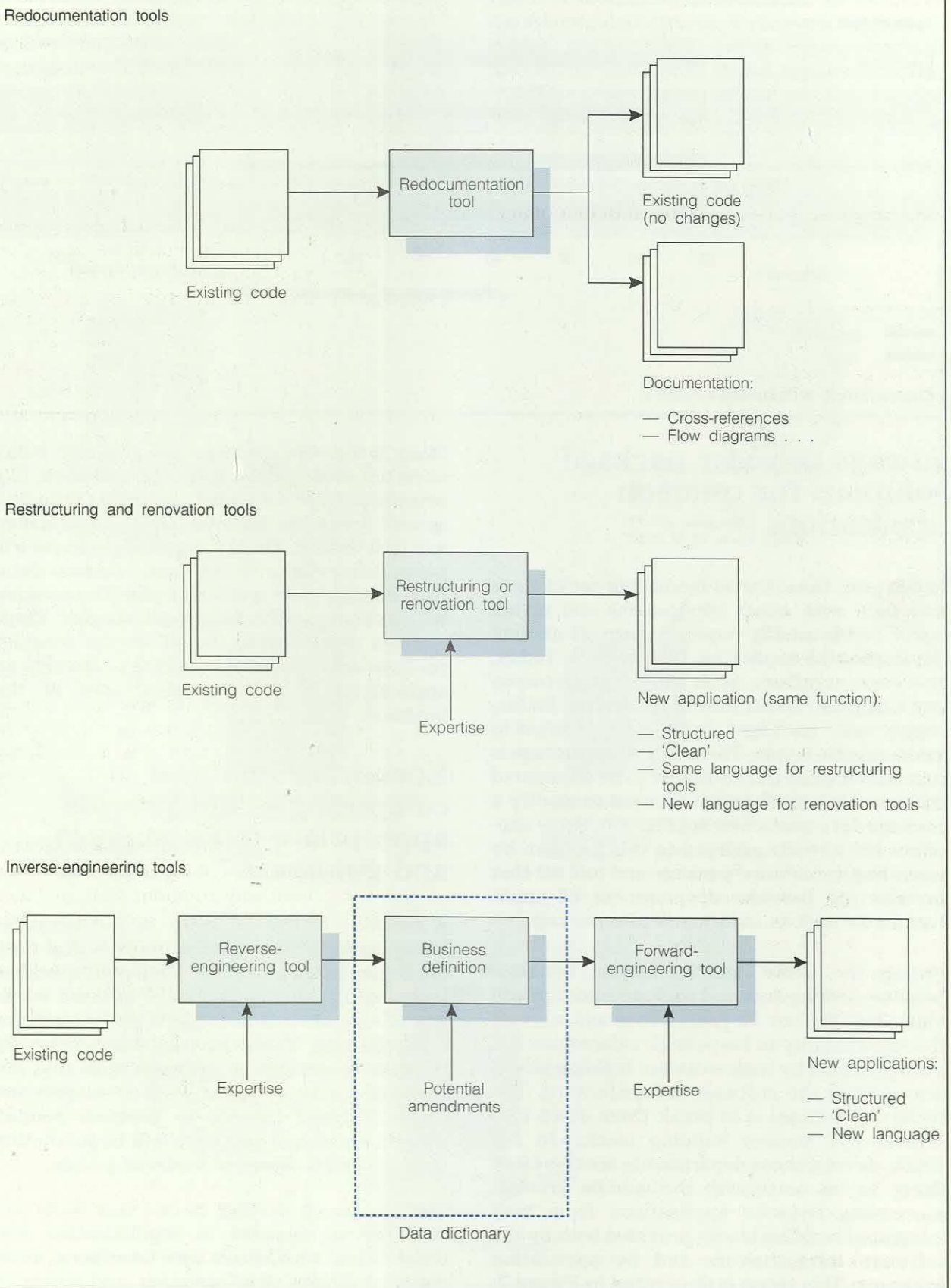
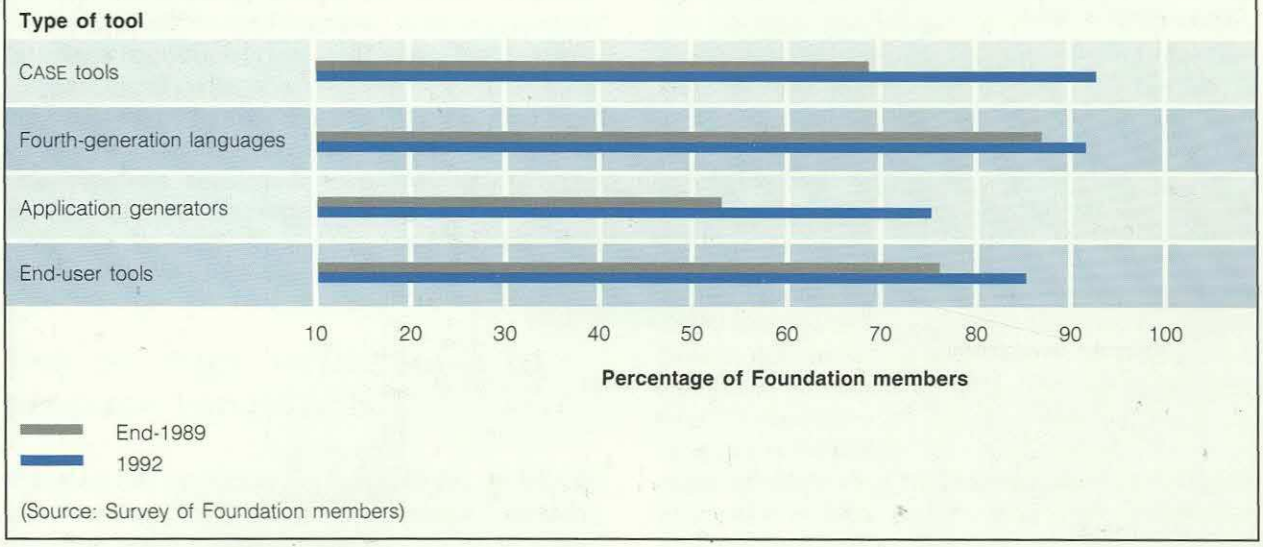


**Figure 6** Over 90 per cent of Foundation members expect to be using CASE and fourth-generation-language tools by 1992




(Source: Survey of Foundation members)

## Always consider package solutions for common applications

In the past, the effort of modifying packages to interface with other applications and match users' needs usually required more effort than developing the application from scratch. Today, however, suppliers, such as SAP in Germany and Computer Associates in the United States, supply 'soft' packages that can be modified to meet specific needs. Their only disadvantage is that they require the developer to be acquainted with the tool set that will be used to modify a package for a particular application. Some suppliers are already addressing this problem by providing a combined package and tool set that permits the bespoke development of applications as well as tailoring of the package.

During the 1990s, the distinction between bespoke development and package solutions will blur. Over the last 20 years, more and more of the functionality in bespoke developments has been provided by basic software building blocks drawn from the software infrastructure. The trend in packages is to break them down into smaller and smaller building blocks. In the 1990s, development departments are therefore likely to be occupying the middle ground, assembling bespoke applications from well integrated building blocks provided both by the software infrastructure and by application packages. This trend is illustrated in Figure 7.

Many integrated packages are available today covering areas common to all businesses, like accounting and personnel, as well as industry-specific areas, like manufacturing, construction, and distribution. These integrated packages will provide interfaces to the more common database management systems, making integration with existing applications much simpler. Thus, systems departments should always consider package solutions when seeking to develop an application in an established area of the business.

## Encourage end-user computing — but provide appropriate tools, support, and guidance

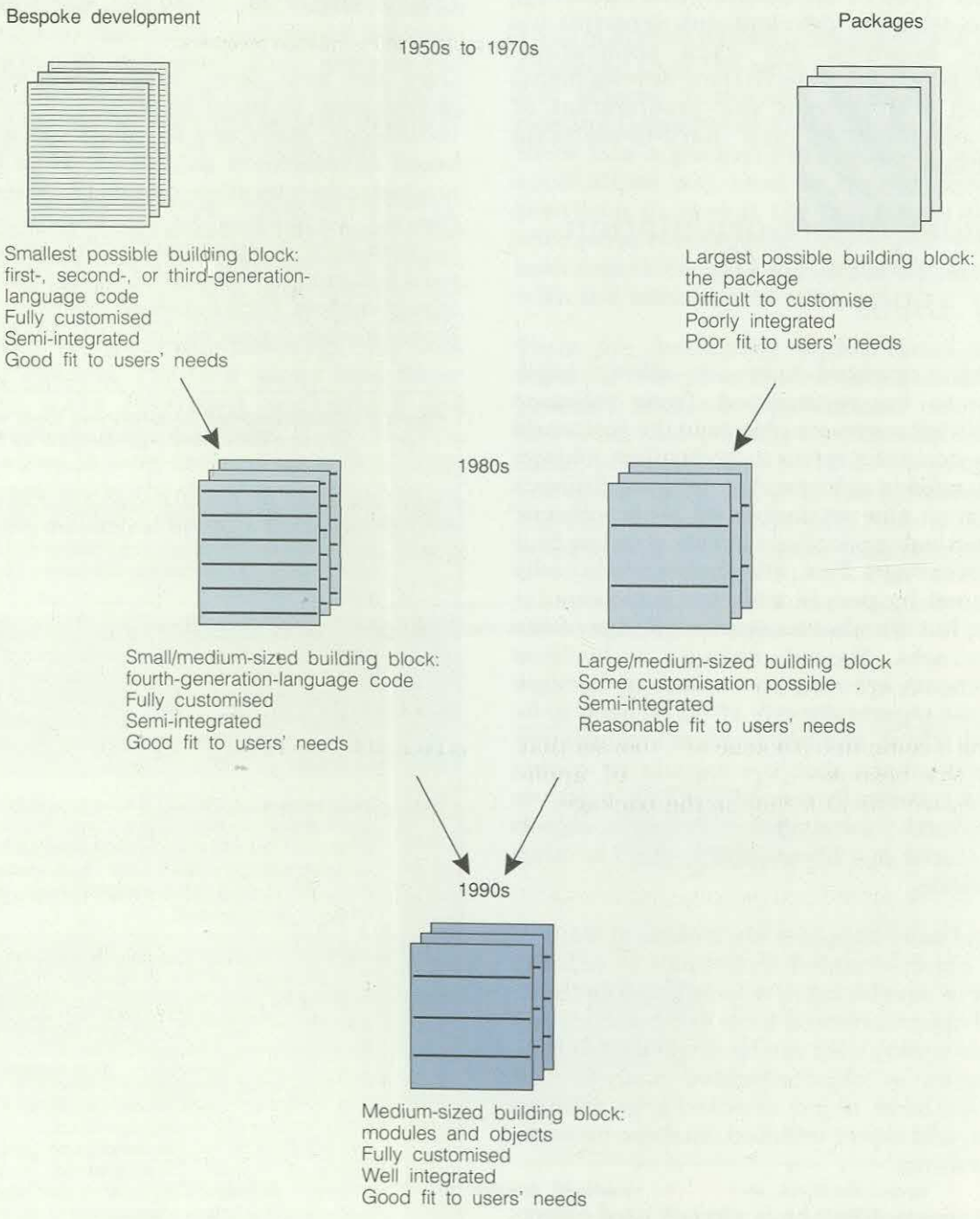
A common feature of many of the advances being made in development tools is that they are becoming easier to use. Moreover, specialist technical knowledge about the internal workings of computers is no longer a prerequisite for using the tools. This means that it is increasingly difficult to distinguish between tools that are designed for use by professional developers and those designed for use by business people. Indeed, some tool suppliers will be marketing their products direct to business people.

The increasing number of end-user tools will continue to improve in sophistication and capabilities, with better user interfaces, automatic validation of information, and powerful

commands making them easier to understand and use. The trend is likely to accelerate as advances continue to be made in both fourth-generation languages and window-based interfaces. As these developments occur, business managers will be able to use the tools to gain direct access to most of the information stored in computers, and to analyse and manipulate it.

As the demand for applications continues to outstrip the development department's ability to produce and maintain them, it is crucial that end-user computing complements the efforts of the systems department. If correctly supported, guided, and managed, end-user computing will not only be a source of useful applications, but it will also relieve some of the pressures on the development department. The challenge is to

**Figure 7 Packages and bespoke developments for common applications will merge in the 1990s**



maintain some control without discouraging the enthusiasm of users.

The systems department does not have unlimited resources, and they must be allocated carefully to ensure that end-user computing is used to the greatest possible effect. We recommend that systems departments categorise the different types of user so as to provide each category with both the tools and the level of support, guidance, and education that will enable it to get the most out of its involvement in end-user computing. The systems department should also set guidelines for developing different types of application, and encourage users to seek the development department's 'seal of approval' for each application. Encouraging users to have their developments approved will prevent the proliferation of poorly documented and hard-to-maintain applications.

## Consider object-orientation as it matures in the early 1990s

The object-oriented approach allows applications to be constructed from software 'objects' that represent objects in the real world and the way they relate to each other. Objects are arranged in a hierarchy, with each object building on the attributes of other objects. Constructing applications in this way has four main advantages. First, the application is easily understood by people who are not computer experts, but who have some knowledge of the business area. Second, changes in business requirements are easy to implement because only those objects directly affected need to be changed. Third, the hierarchical arrangement of objects means that the amount of coding required is radically reduced, which improves both productivity and reliability. Fourth, objects can be stored in a library and re-used in other applications.

More and more suppliers are looking at ways of adding object-oriented techniques to existing tools, or of developing new tools based on them. Several object-oriented tools and methods are available today; they can be divided into three main types — object-oriented analysis and design methods, object-oriented programming systems, and object-oriented database management systems.

Several organisations have already used object-oriented tools to develop a range of applications and have found them to be very productive. Many of the tools available today, however, lack the basic features, such as a high-level language and data recovery and back-up, required for commercial computing. The current shortcomings will undoubtedly be overcome, and the take-up of object-oriented development techniques is likely to develop quite dramatically in the next five years, as described in Figure 8. There are even proposals to enhance the Cobol standard to include object-oriented features.

## Adopt rule-based tools as they mature in the 1990s

Over the next five years, there will be a phenomenal increase in the use of rule-based technology. Many tool suppliers will add rule-based development facilities to their existing products, thereby allowing rule-based (or expert system) techniques to be integrated with conventional development techniques. These tools will enable commercial applications to use expert-system techniques to resolve complex problems that cannot easily be resolved by more

**Figure 8 Object-oriented concepts and tools will be in use within most organisations by 1994**

| | |
|---|---|
| 1990 | Several organisations use object-oriented programming systems and object-oriented database management systems to develop complex applications. Organisations start introducing window- and icon-based interfaces to in-house applications. |
| 1991 | Tool suppliers release more comprehensive object-oriented tool sets and database management systems for a wider range of hardware and software environments. Users place pressure on the development department for more object-oriented interfaces similar to the windowing facilities available in high-street products. |
| 1992 | Most in-house applications have a standard object-oriented-type interface. Object-oriented facilities appear in several tools and in application areas such as office automation and process control. Organisations assess the viability of using object-oriented concepts for a wider range of developments. |
| 1993 | Early users of object-oriented development tools start to reap the benefits from the library of objects that will have been built up. Organisations extend their use of object-oriented tools for a wider range of complex applications. |
| 1994 | I-CASE suppliers start supplying object-oriented tools and database management systems. Packages containing commonly used objects become available, and these will be incorporated into an organisation's library of objects. Object-oriented development becomes established as one of the approaches available to the systems department to aid in the task of developing applications. |

traditional techniques. In particular, rule-based tools will make it possible to develop rapidly changing or complex applications that would previously have been too expensive, or that would have resulted in an impossible maintenance task.

Rule-based techniques will also be combined with object-oriented and more conventional techniques to form fifth-generation toolkits, which will enable very sophisticated and complex applications to be developed and easily maintained. Application-specific tools for building expert systems will enable developers with no experience of knowledge engineering to develop knowledge-based applications. Sophisticated tools with embedded expert systems will also appear, and will be a very powerful aid to developers because they will provide details of how the tools should be correctly used.

These advances will help organisations overcome many of the barriers that have limited the take-up of rule-based or expert systems to date, such as the need for specialist expertise, the lack of success to date with expert systems, and the inability to integrate rule-based systems with existing systems. Figure 9 shows how these advances make rule-based applications and expert systems more accessible to developers, with the loss, in some cases, of some flexibility. By the mid-1990s, the development and use of rule-based applications will be commonplace. Systems departments should therefore assess whether rule-based tools could be beneficial to their organisation and adopt them for specific purposes as appropriate tools become available.

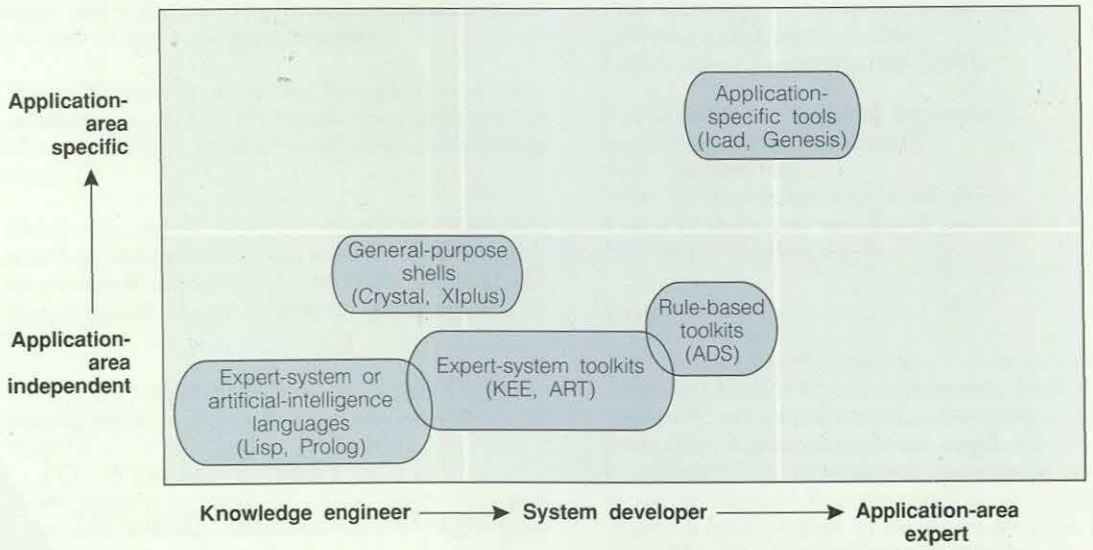
## Monitor developments in other tools emerging in the 1990s

Concurrently with the advances discussed above, there will be changes in the capabilities of existing tools and additions to them that will enable systems departments to develop different types of applications.

*Tools to exploit parallel computers* will become available. Over the next few years, such tools will become reasonably efficient at taking existing (sequential) applications and moving them into a parallel environment. Since few applications will need to be redesigned and rewritten to exploit the full power available with parallel computers, these tools will allow most organisations to exploit parallel computers with the minimum of rework.

*Tools for developing sophisticated human/machine interfaces* are becoming available. At present, in-house applications rarely have powerful, user-friendly human/machine interfaces, primarily because of development time and cost constraints. Over the next few years, the introduction of powerful tools will make it possible to develop window-based interfaces for

**Figure 9 There is a trend towards less flexible but more usable rule-based development tools**



many in-house applications. This type of tool will become much more common once the current disputes over the various 'look and feel' copyrights are settled.

*Multimedia tools* will enable new types of applications to be developed. We expect that, by about 1993, tools will be available to add a multimedia capability to windows-based interfaces. In the mid-1990s, products like NextStep from NeXT, and NewWave from Hewlett-Packard will mature to form a powerful multimedia interface to applications. The major limiting factor on the development of multimedia applications, however, is the limited availability and immaturity of multimedia database management systems. Several suppliers are currently offering such systems — Informix Software with the Informix-Online database, and Graphael Soretas with G-BASE, for example — but it will take time for them to be adopted.

*Tools for building distributed applications* will be available from major suppliers such as Information Builders, and Must Software International, both of whom have adopted phased approaches to ensure that, by the early 1990s, their respective fourth-generation languages (FOCUS and Nomad) can be used to develop distributed applications. This trend will be followed by many others, so that, by 1993, most development departments will have access to tools that can be used to develop distributed applications.

*Tools for integrating office automation and data processing applications* are now becoming available. Oracle, for instance, aims to provide a database management system that supports all information, regardless of its form or function. Thus, the Oracle system will support word processing information, facsimile information, and data processing information, regardless of its origin or business area.

Most of the tools discussed in the report have the potential to make the development department's task much easier in the longer term. However, they are still at the very early stages of development, and the claims made by their suppliers should be treated with a great deal of caution. The full report contains advice about how to exploit the tools as they emerge.

BUTLER COX FOUNDATION

## The Butler Cox Foundation

The Butler Cox Foundation is a service for senior managers responsible for information management in major enterprises. It provides insight and guidance to help them to manage information systems and technology more effectively for the benefit of their organisations.

The Foundation carries out a programme of syndicated research that focuses on the business implications of information systems, and on the management of the information systems function, rather than on the technology itself. It distributes a range of publications to its members that includes Research Reports, Management Summaries, Directors' Briefings, and Position Papers. It also arranges events at which members can meet and exchange views, such as conferences, management briefings, research reviews, study tours, and specialist forums.

***Membership of the Foundation***

The Foundation is the world's leading programme of its type. The majority of subscribers are large organisations seeking to exploit to the full the most recent developments in information technology. The membership is international, with more than 400 organisations from over 20 countries, drawn from all sectors of commerce, industry, and government. This gives the Foundation a unique capability to identify and communicate 'best practice' between industry sectors, between countries, and between IT suppliers and users.

***Benefits of membership***

The list of members establishes the Foundation as the largest and most prestigious 'club' for systems managers anywhere in the world. Members have commented on the following benefits:

- The publications are terse, thought-provoking, informative, and easy to read. They deliver a lot of message in a minimum of precious reading time.
- The events combine access to the world's leading thinkers and practitioners with the opportunity to meet and exchange views with professional counterparts from different industries and countries.
- The Foundation represents a network of systems practitioners, with the power to connect individuals with common concerns.

Combined with the manager's own creativity and business knowledge, Foundation membership contributes to managerial success.

***Recent Research Reports***

56 The Impact of Information Technology on Corporate Organisation Structure
57 Using System Development Methods
58 Senior Management IT Education
59 Electronic Data Interchange
60 Expert Systems in Business
61 Competitive-Edge Applications: Myths and Reality
62 Communications Infrastructure for Buildings
63 The Future of the Personal Workstation
64 Managing the Evolution of Corporate Databases
65 Network Management
66 Marketing the Systems Department
67 Computer-Aided Software Engineering (CASE)
68 Mobile Communications
69 Software Strategy
70 Electronic Document Management
71 Staffing the Systems Function
72 Managing Multivendor Environments
73 Emerging Technologies: Annual Review for Managers
74 The Future of System Development Tools

***Recent Position Papers and Directors' Briefings***

Information Technology and Realpolitik
The Changing Information Industry: An Investment Banker's View
A Progress Report on New Technologies
Hypertext
1992: An Avoidable Crisis
Managing Information Systems in a Decentralised Business
Pan-European Communications: Threats and Opportunities
Information Centres in the 1990s

***Forthcoming Research Reports***

Assessing the Value from IT
Systems Security
New Telecommunications Services
Using IT to Transform the Organisation
Electronic Marketplaces

## Butler Cox

The Butler Cox Foundation is one of the services provided by the Butler Cox group. Butler Cox is an independent management consultancy and research company. It specialises in the application of information technology in industry, commerce, and government throughout Europe and the rest of the world. It offers a wide range of services to both users and suppliers of information technology.

Butler Cox plc
Butler Cox House, 12 Bloomsbury Square,
London WC1A 2LL, England
☎ (01) 831 0101, Telex 8813717 BUTCOX G
Fax (01) 831 6250

*Belgium and the Netherlands*
Butler Cox Benelux bv
Prins Hendriklaan 52,
1075 BE Amsterdam, The Netherlands
☎ (020) 75 51 11, Fax (020) 75 53 31

*France*
Butler Cox SARL
Tour Akzo, 164 Rue Ambroise Croizat,
93204 St Denis-Cédex 1, France
☎ (1) 48.20.61.64, Télécopieur (1) 48.20.72.58

*Germany (FR), Austria, and Switzerland*
Butler Cox GmbH
Richard-Wagner-Str. 13, 8000 München 2, West Germany
☎ (089) 5 23 40 01, Fax (089) 5 23 35 15

*Australia and New Zealand*
Mr J Cooper
Butler Cox Foundation
Level 10, 70 Pitt Street, Sydney, NSW 2000, Australia
☎ (02) 223 6922, Fax (02) 223 6997

*Finland*
TT-Innovation Oy
Meritullinkatu 33, SF-00170 Helsinki, Finland
☎ (90) 135 1533, Fax (90) 135 2985

*Ireland*
SD Consulting
72 Merrion Square, Dublin 2, Ireland
☎ (01) 766088/762501, Telex 31077 EI,
Fax (01) 767945

*Italy*
RSO Futura Srl
Via Leopardi 1, 20123 Milano, Italy
☎ (02) 720 00 583, Fax (02) 806 800

*Scandinavia*
Butler Cox Foundation Scandinavia AB
Jungfrudansen 21, Box 4040, 171 04 Solna, Sweden
☎ (08) 730 03 00, Fax (08) 730 15 67

*Spain and Portugal*
T Network SA
Núñez Morgado 3-6°b, 28036 Madrid, Spain
☎ (91) 733 9866, Fax (91) 733 9910