Managing Multivendor Environments

BUTLER COX FOUNDATION

Research Report 72, November 1989

BUTLER COX FOUNDATION

Managing Multivendor Environments

Research Report 72, November 1989

Butler Cox plc

LONDON
AMSTERDAM MUNICH NEW YORK 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.

BUTLER COX FOUNDATION

Managing Multivendor Environments

Research Report 72, November 1989

Contents

A Management Summary of this report has been published separately and distributed to all Foundation members. Additional copies of the Management Summary are available from Butler Cox.

Report synopsis

Systems directors responsible for multivendor environments have to strike a balance between procuring the most suitable computing products for particular applications, and restricting the number of different vendors in order to control the cost of managing and running a complex computer environment. To get this balance right, they need to appreciate what the cost implications are, and to concentrate their efforts on those areas where the impact on costs is greatest. Three areas warrant special attention — the technical architecture, which needs to be designed to reduce complexity; staff, who need to be managed to encourage flexibility; operations management, which needs to be organised to ensure that systems can operate smoothly in support of the business. Considerable management effort will be required, but it is essential if the benefits of a multivendor environment are to be realised without a heavy cost penalty.

Chapter 1

Balancing the conflicting pressures of a multivendor environment

This report is concerned with the management issues that arise when an organisation's computing facilities are supplied by several vendors, resulting in a 'multivendor environment'. The word 'environment', as used here, encompasses all of the computing facilities used by an organisation. This overall computing environment is usually made up of various levels of computer systems — mainframes, minicomputers, personal computers, workstations, and so on. Often, the computer systems at one level will be based on a different vendor's products from those at another level.

A particular computer system will consist of hardware that supports one or more *software environments*, each defined by a software architecture. Sometimes, an organisation may have several systems at one level, with each system conforming to a different vendor's architecture — IBM and ICL mainframe systems, for example. Some software environments are available for systems at more than one level. Thus, Unix can run on systems from PCs to supercomputers, and applications conforming with SAA will run on a wide range of IBM computers.

If the computer systems supplied by different vendors operate independently of each other, the technical problems of managing a multivendor environment are not much more difficult than if all the systems were provided by a single vendor. The problems escalate when the products from one vendor need to interwork (or be integrated) with the products of another vendor. In this context, multivendor environments can be of two kinds:

— Where equipment and software from different vendors are used to construct a computer system that conforms to a particular software environment. This situation is commonly found with IBM plug-compatible hardware that conforms to IBM's proprietary architecture. We refer to this as a single-architecture, multivendor environment.

— Where applications running in different software environments (usually defined by two different manufacturers' proprietary architectures) need to be integrated. We refer to this as a multi-architecture, multivendor environment. Many Foundation members have this type of multivendor environment.

Figure 1.1 (overleaf) shows the four possible combinations of single/multi-architecture and single/multivendor. It is important to understand the distinction between a multivendor and a multi-architecture environment, because it has a significant bearing on how the systems function must be managed to serve the business to best effect. An organisation that falls in the upper right-hand quadrant has the most complex situation to manage — a multivendor environment that is also multi-architecture. It might, for example, have ICL and IBM software environments, or IBM and Digital software environments.

It is equally possible to have a multivendor environment that is not multi-architecture — an organisation in the lower right-hand quadrant might, for example, have IBM and plug-compatible mainframes such as Comparex or Amdahl. An organisation in the upper left-hand quadrant has just one vendor, but several architectures — for example, MVS and DOS from IBM, or GCOS6, GCOS7, and GCOS8, all from Bull. The situation in the lower left-hand quadrant, where an organisation has a single vendor and a single architecture, is the simplest and the easiest to manage.

Figure 1.1 also shows that there are advantages and disadvantages associated with each

Figure 1.1 It is important to distinguish between multivendor and multi-architecture environments

	Single vendor	Multivendor
Multi-architecture	*Examples*: IBM: MVS and DOS; Bull: GCOS6, GCOS7, and GCOS8 *Advantage*: Simpler relationships with vendors *Disadvantages*: Limited ability to choose most appropriate functionality for applications; Reliance on a single vendor	*Examples*: ICL and IBM; IBM and Digital *Advantages*: Wider range from which to choose best product for the job; Reduced dependence on a single vendor; Rapid access to new technology *Disadvantages*: Limited multisourcing; Increased complexity
Single architecture	*Example*: Siemens BS2000 *Advantage*: Simplicity *Disadvantages*: Reliance on a single vendor; Reduced ability to choose most appropriate environment for a particular application	*Examples*: IBM and Amdahl; MS-DOS *Advantages*: Negotiating power; Reduced dependence on a single vendor; Opportunities for multisourcing *Disadvantage*: Reduced ability to choose most appropriate environment for a particular application

quadrant. Some organisations will deliberately aim to position themselves in a particular quadrant, or indeed, choose to move from one quadrant to another.

If the aim is to achieve technical and managerial simplicity, the goal should be a single-vendor, single-architecture environment at each level. If specialised functionality is required — for example, for fault-tolerant operations, or for CAD — a multi-architecture environment will usually be necessary. A multi-architecture environment will also give an organisation greater flexibility to select the best package for specific applications. A single-architecture, multivendor environment will usually increase the negotiating power of an organisation because it will have a choice of suppliers. Thus, users of IBM and plug-compatible machines have found that they can make large savings, and users of Unix systems find that the hardware is priced very competitively.

This report, however, is concerned primarily with managing a multivendor, multi-architecture environment — the upper right-hand quadrant of Figure 1.1 — and with the problems of integrating applications running in the different architectures. From now on in this report, the term 'multivendor environment' is used in this context, unless otherwise stated. Nevertheless, many of the issues addressed apply equally to single-vendor, multi-architecture environments. It can be nearly as difficult to integrate applications running in two different IBM software environments, for example, as it is to integrate applications running in IBM and Digital software environments.

The main advantage of a multivendor environment is that there is a wider range from which the best combination of hardware and software can be chosen for a particular application. Indeed, of the 80 per cent of Foundation members who have computer equipment from several suppliers, nearly half quoted this as the greatest advantage. However, a multivendor environment does result in increased complexity and increased costs in some areas, and does not necessarily give an organisation the greater negotiating power that might be expected, because there is only limited scope for competition between the vendors. Ken Taylor, the executive and office systems manager for the UK Post Office, running NCR and Unisys systems, is reported as saying: "We had a policy of playing one supplier off against the other, but that did not work. Both companies knew that we couldn't simply switch all our applications from one to the other."

Systems directors are thus faced by two conflicting pressures: on the one hand, there are strong business (and other) pressures to increase the number of suppliers; on the other, the management problems and the high costs are strong incentives to standardise on a single architecture or a single vendor. Managing successfully in a multivendor environment will require systems directors to strike a balance between the business pressures to increase the number of architectures and vendors, and the extra management time and costs that will undoubtedly be incurred by doing so. Whatever the balance that is struck, the implications will be significant — on costs, on efforts to integrate applications running in the different software environments, on staffing, and on day-to-day operations.

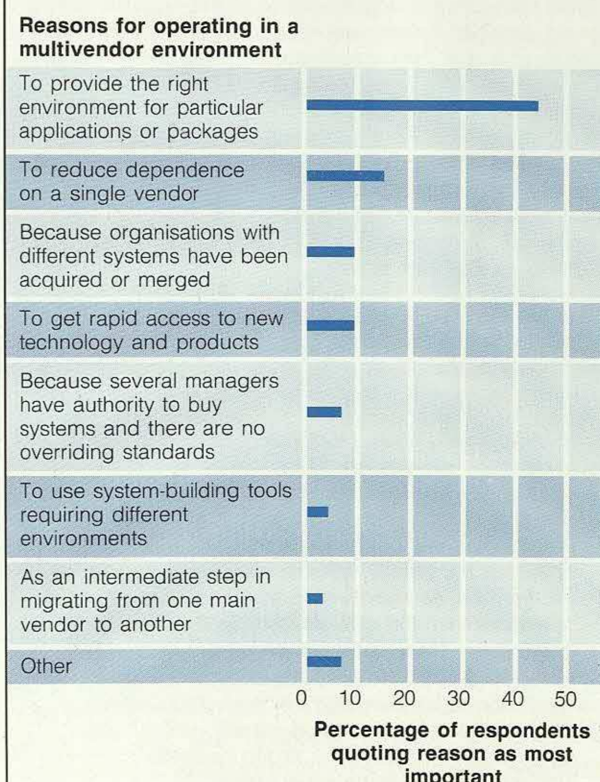
There are many pressures to increase the number of vendors

There are many pressures on organisations to increase the number of hardware and system software suppliers that they use. Most of these pressures are business-related:

To provide the most appropriate computer systems for particular applications or packages: Nearly half of the Foundation members who responded to our questionnaire had a multivendor environment because it provided the best hardware and software for their particular mix of applications or packages (see Figure 1.2). They may, for example, have chosen different suppliers for batch and online workloads; they may have chosen an applications package that was particularly well suited to their needs, but that required different hardware; they may have had a main supplier who could not cover the whole range of computing requirements.

To reduce dependence on a single supplier: Some organisations prefer to have a choice of IT supplier, partly to encourage competition (and hence, be in a position to negotiate better deals), and partly to secure multiple sources of supply for this vital corporate resource. This

Figure 1.2 Providing the right environment for particular applications or packages is the reason most often quoted for operating in a multivendor environment



(Source: Survey of Foundation members)

approach works well in a single-architecture environment, but there is only limited scope for playing one vendor off against another in a multi-architecture environment.

To provide better service to decentralised units: Decentralised business areas are likely to have different computing requirements that cannot all be met by a single supplier.

To buy the most cost-effective hardware: Organisations may create a multivendor environment because they aim to buy the most cost-effective hardware. One organisation reported a saving of $1.6 million in its hardware expenditure as a result of its multivendor policy.

To provide rapid access to technology: Organisations may wish, or need, to use a particular technology that their main computer supplier does not (yet) support.

Other organisations may find themselves operating in a multivendor environment not as a result of a conscious choice, but by force of circumstance. As the result of an acquisition or a merger, an organisation may have acquired systems departments with equipment and system software from different suppliers. This is likely to happen more frequently as international mergers and acquisitions become more common, but even if a merger is a reason for moving into a multivendor situation, it is not a reason for staying there. Figure 1.3 describes the experience of an organisation that found itself in a multi-architecture environment as a result of acquisitions, but that has a strategy to revert to a single architecture.

Multinational organisations may have no choice other than to create a multivendor environment if there are different major suppliers in each of the countries in which they operate. A good

Figure 1.3 A German organisation is pursuing a single-architecture strategy despite numerous acquisitions

A financial services group

The independent systems function of this financial services group provides systems services to other members of the group, and sells them to companies outside the group. On the mainframe level, the organisation runs a single-architecture system, based on IBM standards, and acquires products from IBM and from compatible suppliers in order to increase its choices in procurement. For non-mainframe products, too, it has standardised on IBM architecture, but will consider vendors of compatible equipment. There are no applications which would benefit from a non-IBM architecture.

Over the past few years, the group has made a number of acquisitions. One of these, a bank, had based its own mainframe operation on a Bull machine. The group was therefore faced with a multi-architecture environment and had to decide what direction it should take. It decided to convert the Bull workload to IBM, for two main reasons:

- Part of the business rationale for the acquisition was to be able to cross-sell products to the different customer bases. This will inevitably lead to increased integration between the bank's systems and those of other group companies, as individual financial advisors sell all the products from the different companies to their customers. There is thus a strong business reason for integrating, and this is easier to do in a single-architecture environment. Since all the business functions can be supported equally well by either architecture, there are no strong business pressures for a multi-architecture environment.
- The group believes that the complexity of the interfaces between applications running in two different software environments would reduce its flexibility.

The group's systems function was supported in its decision by the bank's plan to move towards IBM over the next 10 years. The aim is now to convert in four years. The group believes that three features are of particular importance in the transition period:

- The software strategy and the applications architecture need to be considered carefully. A well defined strategy provides guidelines on what to do with the systems of newly acquired companies and how to integrate them with the existing applications. The group believes that it is important to involve user functions in the formulation of the software strategy, since they often have a better view of how new technologies can be used in the systems that support the business.
- Staff questions merit special attention. Many of the group's established staff were reluctant to convert, so a retraining programme was introduced. Careful personnel planning, taking account of natural staffing fluctuations, was particularly important in managing the transition.
- Strong management commitment to the single-architecture strategy is the most important factor in ensuring its success.

Further acquisitions in the future may mean that other machines with different architectures are brought into the group. At that time, the group will consider whether there are business reasons for maintaining different architectures. If there are not, it expects to continue with its single-architecture, multivendor policy.

example is Solvay, a European chemicals manufacturer with its head office in Belgium. Only 10 per cent of its 45,000 staff are based in Belgium, and there are large manufacturing and sales operations in many other countries. The central systems function has found it very difficult to standardise on a limited number of vendors. It would have been quite impossible, and inappropriate, in view of the distinct nature of the business conducted in each country and the dominance of different vendors, to standardise on a single vendor throughout the group.

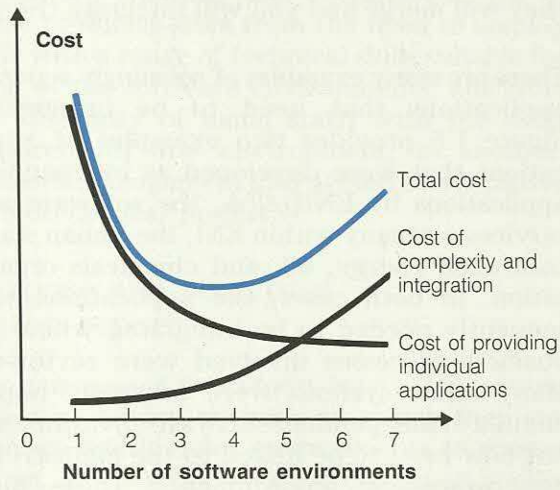
Multivendor environments are costly and complex to operate

Even if it is technically possible for an organisation to meet all of its computing needs with a single level of computer systems, it may not be economical to do so. Thus, although it may be possible to provide personal computing facilities on an existing mainframe, it is likely to be cheaper to do so on a personal workstation from a different supplier. The other extreme of choosing the optimum hardware and software for each application would also be very expensive because of the resulting complexity, particularly if applications running in different software environments need to be integrated.

Figure 1.4 shows how the total cost of the systems function is likely to vary as the number of different types of computer systems increases. There are two cost factors at work here. At first, the cost of providing individual applications will fall sharply as the number of types of computer systems is increased, reflecting the fact that it is not economical to meet all computing requirements with a single level of computer systems. Thus, most Foundation members now have mainframe, minicomputer, and PC systems. As the number of levels increases, the cost of the resulting complexity increases, albeit slowly at first. However, if *additional* software environments are added at any of the levels to meet specific application requirements, the cost savings, in terms of hardware and software, will be less dramatic, while the costs associated with the resulting complexity and integration needs will rise steeply.

The overall cost of the systems function is, of course, the 'U'-shaped curve in Figure 1.4 that is the sum of the other two curves — with the bottom of the 'U' indicating the optimum number of software environments. The optimum number will depend on particular circumstances, but is likely to be between three and five. In some cases, this will mean increasing the number of vendors. More often, however, it will mean reducing a large number that has built up in an *ad hoc* manner over several years. Many of the organisations we spoke to during the research for this report are doing just this — one UK-based company is aiming to reduce the number of software environments it uses from 19 to four (IBM and ICL mainframes, Tandem, and MS-DOS).

Figure 1.4 The total cost of a systems function varies according to the number of different software environments



Even with the number of vendors reduced to the optimum, the demands on management will be significant. Multivendor environments are very complex to manage. The main areas of concern are the difficulties of integrating applications, the problems of maintaining an acceptable level of service, and internal staffing problems.

Integrating applications

Difficulty in integrating applications that operate in different software environments was quoted as the biggest single problem in managing a multivendor environment, yet there are increasing business pressures for this to happen. More and more large organisations are finding that applications originally designed to be standalone need to be integrated so that they can share data. As the systems director of one major international bank put it: "It is an illusion to think that applications will stay separate: they will merge and you will integrate them."

There are many examples of seemingly separate applications that need to be integrated; Figure 1.5 provides two examples of applications that were developed as independent applications by ENIDATA, the software and services company within ENI, the Italian state-controlled energy, oil, and chemicals organisation. In both cases, the applications subsequently needed to be integrated when the business processes involved were reviewed. Many office systems were originally implemented in independent software environments but now need to be linked to the mainstream data processing environments. These links support electronic-mail applications, or data sharing, or simply provide office systems users with access to applications running on the mainframe.

There are different degrees of integration, ranging from no integration at all to the need for close, realtime interchange between the different software environments. The pressures for integration are also varied, so that integration can be classified as being data-driven, user-driven, or process-driven. These aspects of integration are discussed further in Chapter 3.

The problems of integrating applications can be particularly acute for organisations operating across national boundaries. Where national computer manufacturers have an established position in different countries — like ICL in the United Kingdom and some ex-Commonwealth countries, Siemens and Nixdorf in Germany, Bull in France, and Olivetti in Italy — the methods of integrating systems may differ in matters of detail.

The need for integration between different software environments also varies from country to country because multivendor environments are more common in some countries than in others. In Italy, for example, where IBM is extremely strong in the mainframe market, and where there is no dominant national mainframe computer manufacturer, there are

Figure 1.5 Applications that are developed as independent applications frequently need to be integrated at a later date

ENIDATA

ENIDATA is the systems services provider for ENI, the Italian state-owned company with subsidiaries active mainly in the oil, energy, and chemicals industries. It provides services both to other companies in the ENI group and to independent organisations. In ENIDATA's experience, multivendor environments are becoming more frequent, as a result of the increasing feasibility of distributed information processing, and the wide diversity of hardware. ENIDATA is also aware of the growing importance of integration. It cites two cases in which independent applications were developed in different software environments, and which subsequently needed to be integrated as business needs changed.

In the first case, a central IBM system was used to capture timesheet information, which was used for both accounting and payroll purposes. A separate copy of the timesheet was rekeyed into a decentralised Digital Vax that supported an application to plan and monitor staff allocation. The two computer systems were completely separate. In order to speed up the staff-allocation application on the Vax, staff were allowed to use portable PCs to enter the timesheet data, rather than fill out paper forms. This meant that there was no paper copy for keying into the central IBM. A simple file-transfer procedure was therefore set up to move the timesheet data from the Vax to the IBM system, thereby increasing the level of integration between the two environments.

The second example is technically a little more complex. Again, it involves the integration of applications running on the central IBM administrative system, and on a decentralised Vax system. The Vax was part of a chemical-plant control system, and was connected to a PC that was used to monitor deliveries of raw materials to the plant. When the complete business process was considered, it became clear that the delivery of raw materials needed to be integrated with the administrative system on the IBM that controlled supplier payments. At first, this was done by transferring files to the IBM system, but the transfer was required in both directions, and will eventually use the SNA LU6.2 procedures for peer-to-peer communication. This higher level of integration has been more difficult to implement owing to the scarcity of appropriate technical skills.

In both examples, there are good reasons for using different environments for different applications; the need for closer integration arises directly from the applications. This situation is typical of many systems with which ENIDATA is involved, and of many multivendor environments we encountered during the research for this report.

comparatively few multivendor mainframe environments. By contrast, in Germany, relatively well-integrated multivendor environments are commonplace, promoted both by strong local suppliers (Siemens and Nixdorf), and by the strength of the manufacturing industry, which has been quick to exploit 'integrated' applications.

In France, there is a strong emphasis on multivendor systems integration as a result of the strength of local manufacturers and systems houses, and of Transpac, the French public packet-switched data network. The extensive use now being made of Transpac to interlink different software environments means that there is a solid base of the technical skills required to build and operate multivendor systems. In the United States, however, our research indicates that multivendor environments are less common than in many European countries, and that applications in different software environments tend to be less closely integrated than they are in Europe.

Maintaining service levels

A variety of problems jeopardise the service level offered to users in a multivendor environment. Some organisations have difficulty providing adequate levels of back-up and contingency. One organisation that had acquired a large number of different minicomputers found that it also had to acquire several back-up machines. Others have problems with upgrades. ICI (the UK-based chemicals group), for example, has IBM and Digital computer systems supporting office systems functions, which worked well with a 'bridge' between the two software environments. At one point, however, IBM upgraded DISOSS, but Digital did not upgrade its part of the bridge, with the result that it was no longer possible to transfer documents between the two environments.

Multivendor environments can also cause service-related problems for users. They may need to use several terminals (one for each relevant software environment), and to learn different log-in sequences for different applications operating in different environments. Moreover, the systems function will probably find it more difficult to provide an acceptable level of user support in a multivendor environment.

Dealing with staffing problems

The most frequently mentioned staffing problems in a multivendor environment were allocation, training, and recruitment. All of these problems stem from the need to employ staff with a range of technical skills suitable for each of the software environments, and from the difficulty of using staff, with the skills required for one environment, in another. Systems management also needs to be sensitive to motivational problems.

Purpose and structure of the report

The purpose of this report is to offer systems directors advice on ways in which they can manage multivendor environments to greater effect. Much of the advice in this report applies equally to simpler computing environments. The added complexity of a multivendor environment simply means that the penalties for not managing it well will be much greater.

Our research was not concerned with the question of whether or not an organisation should be operating in a multivendor environment, although we have provided some indications of the circumstances in which it might be wise to move towards a less complex situation: we have concentrated, instead, on the issues that are of paramount importance to an organisation that is in such an environment, either by choice or by force of circumstance. (The scope of the research carried out for this report, and the research team are described in Figure 1.6, overleaf.)

In Chapter 2, we examine the effect on costs of operating in a multivendor environment, particularly at the mainframe level. Although large savings are possible when acquiring products, considerable extra costs can arise over time, which will usually outweigh the initial savings. We describe a model that will help Foundation members to identify those areas that have a significant impact on costs and that therefore need to be managed most carefully.

In Chapter 3, we identify the basic elements of an organisation's overall technical architecture that need to be defined in order to control the costs associated with multivendor environments — the hardware and applications architectures

Figure 1.6 Scope of the research and research team

The conclusions and recommendations of this report are based on an international research programme carried out in the latter half of 1989. The research consisted of a review of the published literature, and 47 interviews with vendors of IT products and services, and users of multivendor systems. We also gathered the views of 137 Foundation members through the questionnaire sent out at the beginning of the research, and a focus group in the Netherlands. We spoke to organisations based in Australia, Belgium, France, Germany, Italy, the Netherlands, the United Kingdom, and the United States.

During the research for the report, the 1989 Benelux/UK Foundation Conference on 'Problem Management in a Multivendor Environment' was held. The conclusions of that conference are reflected in this report.

The report was researched and written by Michael Lloyd, a senior consultant based in Butler Cox's London office. He was assisted by Tony Manley, a principal consultant, also in the London office, Onno Schroder and Frans Molhoek (Amsterdam), Lothar Schmidt (Munich), Xavier Dalloz (Paris), John Cooper (Sydney), and Antonio Morawetz (Milan).

and the software infrastructure. They need to be designed to reduce complexity and to facilitate integration. Standards are helpful, but are certainly not a panacea for limiting the risks and costs of integrating applications in different software environments. Defining and maintaining a technical architecture does, however, require substantial investments of time and effort.

The two main staff management problems in a multivendor environment are staff allocation and training. In Chapter 4, we describe how recruitment and training policies can be used to build a team of systems staff that can be used effectively in different software environments. Backed up by a small group of technical experts, such a team can alleviate many of the staff problems typical of a multivendor environment. Again, considerable management effort is required to overcome the problems, but the resulting greater flexibility in the ways in which staff can be allocated will more than repay the effort.

Chapter 5 addresses the problem of maintaining a high level of operational service to users in a multivendor environment. Good procedures are particularly important, and special attention must be paid to making the multivendor environment as transparent as possible to users. In order to manage day-to-day operations problems, the relationships with existing suppliers need to be controlled, and alternative sources of supply for operational services should be considered.

Chapter 2

Understanding the cost implications

A major concern expressed by Foundation members in their responses to our questionnaire was whether a multivendor environment increases or decreases the overall cost of the systems function. In our research, we found that some members claim to be achieving substantial reductions, while others believe that a multivendor environment increases costs. This divergence of views is due largely to a failure to recognise that there are two kinds of multivendor environments.

In general, a single-architecture, multivendor environment will cost less than the equivalent single-architecture, single-vendor environment. In a multi-architecture, multivendor environment, it is usually possible to reduce the costs of acquiring hardware and software products, but the continuing costs, particularly staff costs, will usually outweigh the initial savings. In this latter case, the organisation must judge whether the business benefits deriving from the multivendor environment can justify the additional costs.

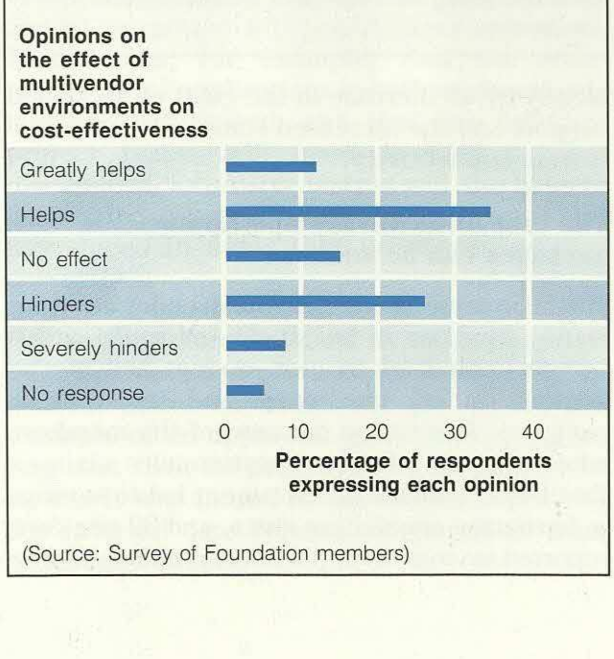
In this chapter, we summarise the varying experiences of Foundation members, and of other organisations that have tried to assess the cost implications of their multivendor environments. We identify the difficulties that prevent many systems functions from accurately calculating the cost implications of their multivendor environments, and describe a model that can be used to help them to prepare such analyses in the future. As we shall see in the final section, the most significant long-term cost item in a multivendor environment is the continuing cost of staff, not the cost of hardware or software.

Costs can be reduced in some areas but will increase in others

We asked Foundation members whether they believe that a multivendor environment helps or hinders the provision of a cost-effective systems service. Overall, more respondents believe that a multivendor environment helps, rather than hinders, the systems function in providing a cost-effective service. Nevertheless, more than a third of the respondents believe that a multivendor environment hinders, or severely hinders, their efforts. Only 15 per cent believe that it has little or no effect (see Figure 2.1).

Figures 2.2 and 2.3, on pages 10 and 11, describe the experiences of two UK financial services organisations with very different views of the cost implications of a multivendor environment. The first has been able to save money by standardising on Unix (in other words, on a single architecture) and choosing the most cost-effective hardware available. The second found that its multivendor (and multi-architecture) environment resulted in increased costs, caused

Figure 2.1 Opinions are divided about the influence of multivendor environments on the cost-effectiveness of the systems function



(Source: Survey of Foundation members)

Figure 2.2 A multivendor environment can reduce hardware and software costs

A major service-oriented organisation

This UK organisation provides a range of services to the public. Sixty-five per cent of its annual $75 million IT budget is accounted for by payments to vendors for hardware, software, maintenance, and telecommunications, and by hardware depreciation. Thus, significant savings can be achieved by relatively small-percentage cost reductions in these areas. The systems function therefore has a long-term policy to reduce these costs, while continuing to provide its users with the required functionality. Multivendor solutions are seen as a critical element in achieving this aim.

In one of its businesses (the provision of insurance services), the systems function has successfully mixed Unix-based open systems with proprietary systems on its network, and this has allowed it to choose the most cost-effective equipment for each application. The systems function claims that this approach has reduced some costs by around 40 per cent, compared with the easier option of standardising on its mainframe supplier's system. The true benefit can be measured in terms of improved service availability without the cost overheads associated with non-stop computing. The main savings are on hardware and software. In arriving at the estimated cost savings, the following factors were taken into account:

Cost allocation: If all the ancillary additional costs associated with a multivendor environment are charged to the first application, these may well be more than the hardware and software cost savings. If the move to a multivendor environment is part of a longer-term strategy, the additional costs, such as those incurred in 'reskilling' the systems function, should be recognised as an investment.

User-training costs: Users need training in the use of any new system, so user training in a multivendor environment does not give rise to additional costs. In the case of this organisation, there have been no significant user-interface problems arising from the use of different vendors' equipment.

Systems staff training costs: Training systems staff to work with a different vendor's product may cost a little more than training them to use a new product from the same vendor. In this organisation's experience, the additional cost is insignificant compared with the cost savings and commercial advantages that can be achieved. Where considerable retraining would be required, it has found it easier to recruit young, inexperienced people and train them, because it has found that it costs less to do this, and takes less time, than it would to retrain some experienced systems staff. In the words of this organisation, "The learning curve is shorter than the relearning curve".

For one of its projects, which required file servers and terminals to be placed in each of 200 branches, the systems function compared the proprietary solution available from its mainframe supplier with an open systems solution, commissioned from a software house, based on a different vendor's Unix machine. For the proprietary solution, the capital cost of the equipment for each branch would have been $14,000, compared with only $7,000 for the Unix-based solution. The equipment is written off over five years, which means that the annual equipment cost for each branch is $1,400 less with the Unix-based solution. Annual maintenance and other costs are also $1,000 less for each branch, resulting in a total annual saving per branch of $2,400 — a reduction of more than 50 per cent compared with the proprietary solution. The initial once-off implementation costs for the Unix-based solution are higher ($105,000) because of the development costs of providing an interface to the proprietary mainframe application.

Thus, for an initial additional expenditure of $105,000, this organisation is able to obtain a saving of $480,000 per annum (initially 200 branches, but forecast to grow to 400). It is also in a better position to implement additional 'open' facilities, such as those based on the X.400 protocols.

largely by an increase in the costs of technical support and the increased complexity of inter-system connectivity.

The cost of hardware and application packages can be reduced

The main areas in which a multivendor environment (whether it be single- or multi-architecture) can lead to cost savings are in the acquisition of hardware and application packages. Forty-nine per cent of the members who responded to the questionnaire claimed that a multivendor environment led to savings in hardware-acquisition costs, and 32 per cent reported savings on application packages. Some of the savings were quite large: 10 of the respondents said that they had been able to reduce the costs of procuring application packages by more than 50 per cent. Because the choice of suitable options is wider in a multivendor environment, organisations have been able to find cost-effective packages to meet a business need that would have required a bespoke application if they had been restricted to a single software environment.

Additional costs may be incurred in all other areas

Even in the hardware and application-package procurement areas, however, a significant

Figure 2.3 A multivendor environment can increase total costs

An international investment bank

A leading financial services company in the City of London has experienced substantially increased IT costs as a result of its multivendor environment. The multivendor environment arose from a combination of two circumstances — the merging of three companies, and the need to have new systems in place in time for the deregulation of UK financial services (the 'big bang'). The three companies all had different computing environments — one had ICL computers, one had Digital and Hewlett-Packard equipment and a Britton Lee database engine, and the third used IBM and Wang equipment.

In addition, the company uses fault-tolerant Stratus equipment for its front-office systems. Since there was no time to create bespoke systems to meet the new requirements, application packages were used as the basis for the new systems. As a result, the required IT systems were fully operational on the first day of deregulation. However, the lack of time to rationalise or rewrite the systems left the company with seven separate suppliers — a truly multivendor situation.

The firm subsequently analysed the costs it had incurred in providing the essential business services by the required date. There had been some savings, but in most areas of expenditure, there had been additional costs. The package solutions had been cheaper than creating bespoke applications, but the resulting increased complexity of inter-system connectivity meant that considerable effort was required by technical-support staff to overcome the problems of transferring data between the systems. For back-up and contingency purposes, the company had to install more duplicate equipment in expensive space than it would have needed with a single-vendor environment. More staff were also needed to handle the different environments, particularly in the operations area and for systems maintenance and technical support.

Additional costs will also be incurred to rationalise the unwieldy situation that has resulted from the need to meet the 'big bang' deadline. Now that the systems are operational and providing support to the business, a programme of rationalisation has been prepared. The intention is to reduce the number of vendors. The main objectives are to reduce data duplication between systems, speed up operations by eliminating tape transfers between systems, and achieve savings in hardware and operational costs. Before this can be achieved, however, considerable systems development costs will have to be incurred, much of which may be seen by the users as 'non-productive', in the sense that it does not add functionality to the application systems.

The lesson from this company's experience is that if a systems department is *forced* into a multivendor environment, it should plan for increased costs in most areas.

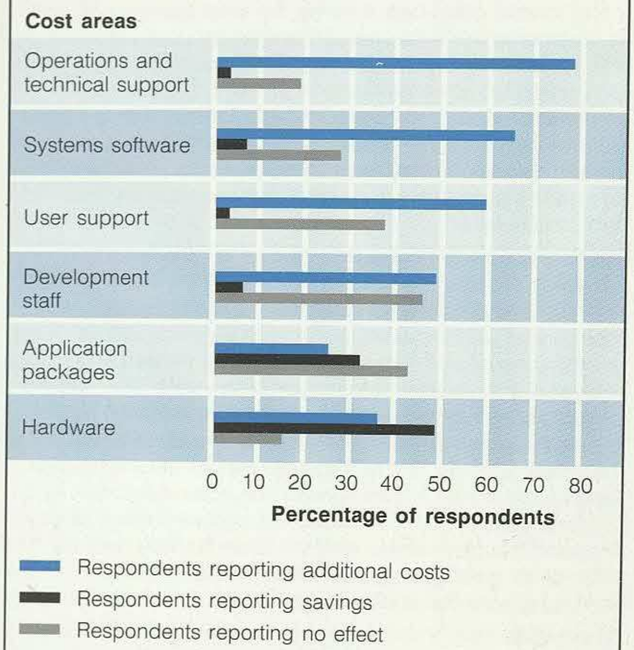
proportion of members reported that a multivendor environment gave rise to additional costs. Thirty-six per cent of members reported extra hardware costs and 26 per cent reported extra application-package costs. In all other cost areas, more organisations reported additional costs than savings, as Figure 2.4, overleaf, shows.

Staff costs tend to be higher in a multivendor environment, particularly in the technical-support area, because of the need for teams of specialists to support each of the software environments. Several organisations reported that they require between 20 per cent and 50 per cent more technical-support staff than they would for a similarly sized single-architecture environment. One interviewee estimated that his organisation spent 60 per cent more on technical-support staff because of the multivendor environment. User-support costs are also higher.

Additional development staff, to work both on specific integration projects and on 'ongoing integration', will also be needed. The costs attributable to specific integration projects are usually very obvious. The costs of ongoing integration are less obvious because they include, for example, the additional work required to make sure that new developments or enhancements in one software environment can be integrated with applications in another. This means, for example, that the documentation of applications in both environments must be scrutinised, and that the different project teams will need to hold detailed discussions. Extensive testing will also have to be carried out to ensure that no errors are introduced in areas that previously worked correctly.

One organisation we talked to pointed out that bugs are more complex and difficult to find in a multivendor environment, and quoted the example of an apparent applications-program bug that took 20 programmers three days to locate. It was caused by the incompatibility of the internal date formats of compilers from different vendors.

Figure 2.4 In most cost areas, more respondents reported increased costs rather than savings, as a result of their multivendor environment



(Source: Survey of Foundation members)

By their nature, the cost implications of a multivendor environment are very difficult to quantify accurately. In the experience of one of our interviewees, every project in a multivendor environment costs between 10 and 15 per cent more, owing to the costs associated with integrating the different software environments. Our consultancy experience confirms this figure for multivendor environments, although we would expect a much lower overhead in multivendor, single-architecture environments.

Multi-architecture environments usually cost more than equivalent single-architecture environments

Figure 2.4 combined the responses of all Foundation members with a multivendor environment, regardless of whether it was single- or multi-architecture. Figure 2.5 shows the responses separately for those members with a single-architecture, multivendor mainframe environment and for those with a multi-architecture, multivendor mainframe environment. It indicates that savings are more frequently achieved by those organisations with a single-architecture, multivendor mainframe environment, and that additional costs occur more frequently where an organisation has a multi-architecture, multivendor mainframe environment.

The main reason for this is that it is much easier to negotiate favourable equipment prices in a single-architecture environment, because the vendors know that a threat to purchase another vendor's products can be carried out. Fifty-eight per cent of the respondents to our questionnaire with a single-architecture, multivendor mainframe environment claimed to have achieved hardware savings, compared with only 45 per cent of respondents with a multi-architecture mainframe environment.

For minicomputers, too, a single-architecture is usually less expensive than a multi-architecture environment. For example, Rijkswaterstaat (RWS), the Dutch Government ministry responsible for motorways, roads, canals, and coastal works, has been able to benefit from price competition in a single-architecture, multivendor minicomputer environment. It wanted to rationalise its installed base of more than 20 minicomputers from three different vendors, and to add between 30 and 40 new minicomputers. RWS carried out detailed costings of the three main options: keeping the present mix of products, standardising on one proprietary minicomputer architecture, or moving to a multisourced Unix environment. These costings showed RWS that, over five years, moving to Unix would be the most cost-effective option. Savings in the basic hardware cost would outweigh the cost of conversion and retraining within three to four years. The most expensive option would be to move to one proprietary system.

As a result of these calculations, RWS has opted for decentralised, single-architecture (Unix-based) systems that will be purchased from several vendors, in order to remain independent of any one vendor and to ensure that it can purchase hardware at a competitive price. RWS has, however, decided to restrict itself to buying Unix hardware from a limited number of vendors, in order to minimise the problems of coordinating different vendors' versions of Unix.

Figure 2.5 identified the effects of a multivendor environment on individual cost areas in Foundation members' installations. It is important, however, to understand the overall impact on the total costs of the systems function. The cost model described in the next section provides the basis for doing this.

A model can be used to understand the cost implications

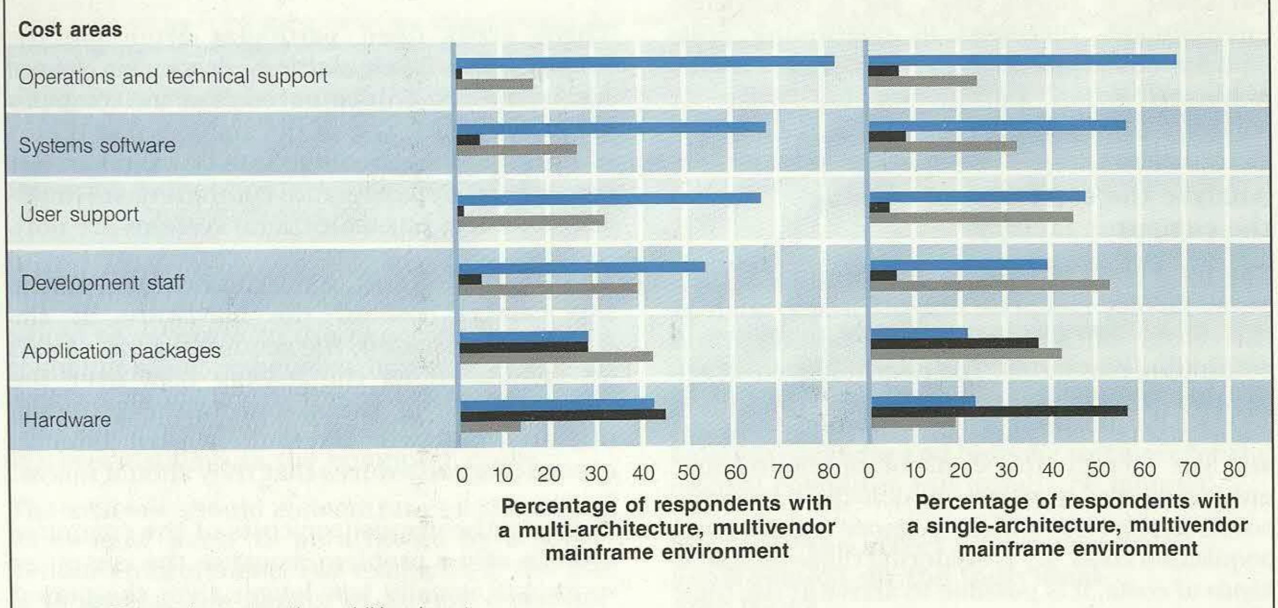
So far in this chapter, we have quoted the views of several organisations about the impact that their multivendor environment has on costs. During the research, it became clear that these views are, in the main, based on subjective judgements, rather than rigorous cost analyses. Relatively few of the organisations we talked to had made any systematic evaluation of the additional costs or savings attributable to their multivendor environment; most do not have accurate and reliable figures about the cost implications. One organisation that instinctively feels that a multivendor environment is better, expressed its view in the following terms: "It is carved in tablets of stone that multisourcing is more cost-effective."

Most systems functions do not attempt to calculate the additional costs (or savings) resulting from a multivendor environment because they believe that they have to offer the functionality provided by the different vendors' products, regardless of the costs involved. In our view, however, the cost implications of a multivendor environment should be assessed, so that business managers can decide whether the cost of providing the additional functionality can be justified in terms of the business benefits that it will provide.

Both the initial costs of purchasing the products and the continuing costs of using them have to be taken into consideration. The initial acquisition costs can be assessed accurately with little difficulty; the continuing costs, however, are much less easy to assess. In many organisations, accounting procedures do not break these costs down into sufficient detail, so special analyses will have to be carried out. There is the added difficulty that few, if any, organisations have a baseline against which to compare the costs of operating in a multivendor environment.

Furthermore, costs and savings attributable to a multivendor environment will be masked by

Figure 2.5 More respondents with multi-architecture, multivendor mainframe environments report greater additional costs than those with single-architecture, multivendor mainframe environments



(Source: Survey of Foundation members)

changes in the workload and by differences in the acquisition and support costs associated with specific suppliers' products. The cost implications can be significant. For example, the July 1989 edition of *Computer Economics Report*, a newsheet specialising in the financial aspects of data processing, quotes a study that indicates that the differences in staffing requirements for programming, operating, and managing comparable computer installations from different vendors can vary by as much as 70 per cent.

Thus, it is not easy to understand the effects that a multivendor environment has on costs. To help Foundation members understand this complex area, we have developed a model. The model first analyses the total cost of owning and running a computer facility, and breaks it down into a variety of cost items. It is then possible to evaluate the effect of a multivendor environment on each of the individual cost items, and thus on the total cost of the computer facility. Doing this will highlight the cost areas where a multivendor environment has the greatest impact. These are the areas that need to be managed particularly carefully.

As we show in the final section of the chapter, the output from the model reflects the intuitive perceptions of members about the cost implications of a multivendor environment. In particular, it shows that, for a mainframe environment, increases in continuing costs usually outweigh any savings that can be achieved when purchasing hardware or software products.

Analyse the total cost of owning the computer facility

The total costs of owning and running a computer facility comprise one-off items, such as the costs of acquiring the facility, and continuing cost items. The continuing costs are of two kinds: operations costs (including maintenance and other 'normal' running costs) and the incremental costs of upgrading and enhancing the systems, which do not occur continuously, but which are more regular than acquisition costs. By considering these different kinds of costs, it is possible to arrive at the total cost of owning the computer facility.

It is also necessary to consider separately the different categories of expenditure — equipment, software, people, communications carriers, and facilities. The different types of cost and the different categories of expenditure can be represented as a simple matrix, with the vertical columns for acquisition and continuing costs, and the horizontal rows for the different categories of expenditure. Figure 2.6 shows the matrix, with the main cost items entered into the individual cells. We have developed this matrix from a model developed originally by Dr Michael Treacy of the Sloan School of Management at MIT to analyse the cost of ownership of computer networks.

We have extended the original concept so that it can be used to analyse the total cost of owning a computer facility for a specific period of time. This is done by estimating both the initial acquisition costs and the continuing costs attributable to each of the items listed in the matrix. The continuing costs should be estimated for the period of time being considered, which will be determined by the expected life of the applications involved. In some types of organisations, this could be as little as two years; in others, it could be seven or more. It will not always be possible to calculate or allocate all the costs exactly, although most organisations should be able to get sufficiently accurate figures, based on the systems department's budget.

Three areas need particular attention: the treatment of acquisition costs for leased hardware, the acquisition costs of the computer room, and the scope of the analysis if only part of the computing facilities is to be examined (for example, if departmental computers are multi-architecture, but mainframe systems are not).

The acquisition cost of purchased equipment is the capital cost of the hardware. If the equipment is leased, the acquisition cost should be calculated as the leasing cost over the expected life of the system, although many organisations will have their own internal accounting procedures that they should follow.

Defining the acquisition costs of the computer room is also a problem, because the computer room will usually last longer than the period being considered. Again, many organisations will have their own accounting conventions to handle this situation, although the simplest method of attributing the computer-room costs

Figure 2.6 The cost implications of a multivendor environment can be determined by using a model of the total costs of owning and running a computer facility

Expenditure category	Acquisition costs	Continuing costs: Operation	Continuing costs: Incremental change
Equipment: Mainframes Minicomputers Microcomputers Communications	Equipment purchase Bridging, hardware, gateways, and so on	Maintenance	Expansion/upgrade Residual value (after five years)
Software packges	Software purchase/one-time licence Bridging and conversion software Application packages	Annual licence Software maintenance fees	Conversion/upgrade costs
Personnel (including external resources): Software development Operations Other	Bespoke software Equipment and software installation Planning, design, selection	Software support and maintenance Application enhancements User support Training for users and systems staff Personnel and head-office functions	Application enhancements Integration and testing Version upgrades
Communications carriers (including value-added services)	Initial connection	Usage charges	Additional connections Configuration changes
Facilities	Computer-room environment Supplies	Contingency provision Supplies Space expense: rental, insurance, utilities	Increased space Restructuring/rebuilding costs (for example, LAN cabling)

Most of the entries in the above matrix are self-explanatory. It is worth pointing out, though, that the 'software packages' category covers all types of packaged software, be it systems software or applications software. The cost of developing and maintaining bespoke software is included in the 'people' category, even if the software is developed as a turnkey operation by a software house. The 'communications carriers' category includes only transmission costs. The costs of communications hardware, software, and non-carrier services are included in the 'equipment', 'software packages', and 'people' categories, as appropriate. Transmission costs are not usually affected by the decision to use one or several hardware or software vendors. The exceptions are when the provision of a carrier service is being multisourced, or when the carrier provides services — such as protocol-conversion, electronic mail, or electronic document interchange — that are used to integrate different computing environments at each end of the communications link.

(Source: Adapted from a presentation made by Michael Treacy at the Networks 89 Conference, Birmingham, June 1989.)

would be *pro rata* to the expected life of the computer room. Sometimes, however, it may be appropriate to allocate the costs *pro rata* to the floor space occupied by the different computer systems installed in the computer room.

The analysis should concentrate on those cost items most likely to be affected by a multivendor environment. For example, if the aim is to analyse the impact of buying communications-carrier services from more than one vendor, it will probably not be necessary for the mainframe computing costs to be considered. Similarly, if the aim is to analyse the implication of multisourcing decentralised computers, it is probably necessary to consider only the costs of software or bespoke applications for the decentralised computers; the running costs of existing packages and bespoke software for the central computer can usually be ignored.

Evaluate the effect of a multivendor environment on the individual cost items

Once the cost items associated with acquiring and operating the computer facility have been determined, the next step is to evaluate the effect of a multivendor environment on each of the cost items. There will, in fact, be different impacts on the different items, and considerable skill and experience are required to ensure that all the implications are taken into account. The two main characteristics of a multivendor environment that determine the extent of the impacts are the level of integration required between the different software environments (both for existing and for new applications), and the nature of the architectures concerned, especially the similarity between them and the availability of cost-effective bridges and other integration aids. For each of the cost items, the savings or additional costs caused by a multivendor environment should, as far as possible, be quantified, either as an absolute value, or as a percentage of total costs.

Usually, the analysis of the total costs of owning the computing facility will show that the continuing costs of people are the single largest item, accounting for around one-third of total costs. The next largest items are continuing equipment costs and equipment-acquisition costs, but they are significantly smaller than the continuing costs of people. It follows that if a multivendor policy results in a 10 per cent reduction in hardware-acquisition costs, the resulting reduction in the total costs would be completely overshadowed by a 10 per cent increase in the continuing costs of people.

Ideally, the evaluation should be done for each of the cost items listed in the matrix. However, if that level of precision is impossible, the evaluation can be carried out for each of the five expenditure categories. It is important to keep acquisition costs and continuing costs separate, because the scale of additional costs and savings for each expenditure category may be different for acquisition and for continuing costs.

Output from the model corresponds with actual experience

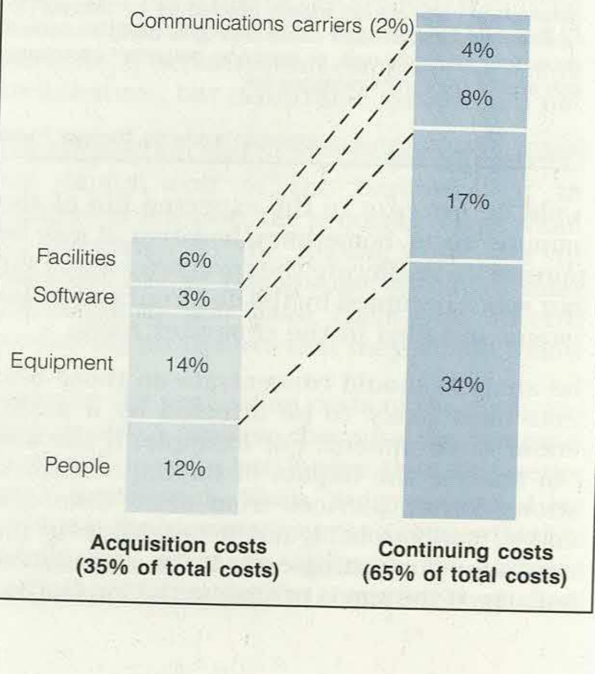
We used the model described above to analyse the impacts on the total costs of owning and running a typical multivendor computer facility. In carrying out the analysis, we focused on the cost implications of single-architecture and multi-architecture mainframe environments, compared with an equivalent single-vendor, single-architecture environment. The results show that the model accurately predicts the varying experiences of Foundation members set out earlier in this chapter.

Based on the analysis we carried out, Figure 2.7 gives a breakdown of the total costs of owning and running a typical computer facility over a five-year period. The first point to note is that about two-thirds of the total costs are attributable to continuing costs. One of the organisations we talked to had done a similar analysis, and found that about 80 per cent of its total costs were attributable to continuing costs. The second point of significance is that the continuing costs of people are by far the largest cost item, followed by continuing equipment costs and once-off equipment-acquisition costs.

The detailed breakdown of costs will differ for individual organisations, sometimes quite considerably. However, our discussions with organisations that had collected some costing information, and our consultancy experience, lead us to believe that the overall pattern of costs shown in Figure 2.7 will be similar in most organisations.

Figure 2.7 The total cost of owning and running a computer facility comprises acquisition costs and continuing costs

Over a five-year period, continuing costs account for two-thirds of the total costs of owning and running a typical computer facility; continuing personnel costs are the single largest item of expenditure.



We then compared the costs of operating in a multivendor environment with the likely costs of operating in an equivalent single-vendor, single-architecture environment. We found that the cost implications of a multivendor environment varied significantly for the different expenditure categories. In a single-architecture, multivendor environment, the main area in which savings can be achieved is hardware, especially hardware-acquisition, because it is possible to negotiate the best deal from competing vendors. Typical savings of between 10 per cent and 40 per cent can be expected. These savings will, however, be partly offset by higher software licence fees, and slightly increased staff costs.

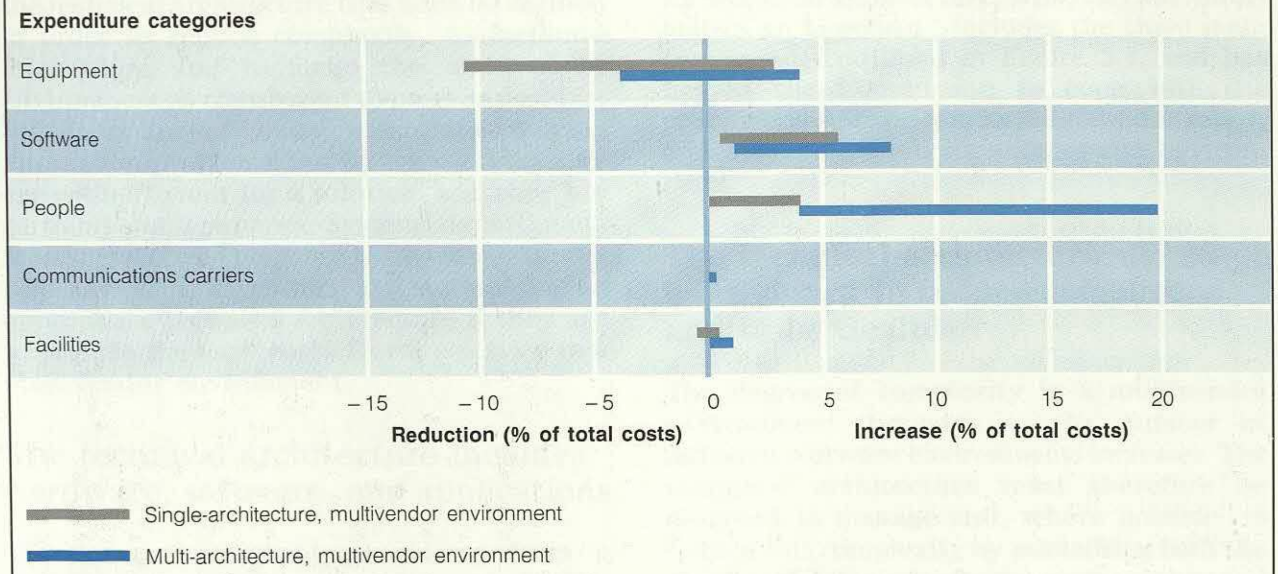
In a multi-architecture, multivendor environment, the areas in which cost savings can be achieved are the acquisition of hardware and applications software. However, these savings are likely to be more than outweighed by the substantial additional costs, especially staff costs, required to integrate applications in the different software environments.

If the percentage savings and additional costs attributable to a multivendor environment are combined with the corresponding portions of the cost-of-ownership profile shown in Figure 2.7, it is possible to obtain an overall prediction of the additional costs and savings likely to result from a multivendor environment. Figure 2.8 summarises the predictions and shows clearly that in a single-architecture, multivendor environment, equipment and software are the categories most likely to be affected. Reductions in expenditure on equipment could reduce total costs by more than 10 per cent, whereas increases in expenditure on software could increase total costs by more than 5 per cent. In a single-architecture, multivendor environment, management attention should therefore be concentrated on these expenditure categories.

In a multi-architecture, multivendor environment, continuing expenditure on people is the area that will be most affected. In contrast with the situation for a single-architecture, multivendor environment, reductions in hardware expenditure will usually be insufficient to compensate for the extra expenditure on people. Management attention should therefore be focused on controlling and containing the continuing costs of people.

Figure 2.8 Increases or decreases in cost will vary in the different expenditure categories according to whether an organisation is operating in a single-architecture or multi-architecture, multivendor environment

The figure shows the cost implications of a multivendor mainframe environment compared with a single-vendor, single-architecture environment. For each type of multivendor environment, it shows the range of percentage savings or increases in the total systems costs that can be attributed to each of the five main expenditure categories. In a single-architecture, multivendor environment, the biggest effects will be savings on hardware expenditure and increases on software expenditure. In a multi-architecture, multivendor environment, the biggest effect is likely to be increases in the costs of people.



The cost implications of a multivendor environment, summarised in Figure 2.8, relate specifically to mainframe systems. Nevertheless, we believe that the same general pattern of cost savings and increases will be found for multivendor minicomputer environments.

Each systems function should do these calculations for its own individual circumstances. The results will show the probable overall effect of operating in a multivendor environment, and will also identify the areas where such an environment will have the greatest impact on the total costs of the systems function.

The extent to which costs will be affected by a multivendor environment depends on the organisation's overall technical architecture — the hardware architectures involved, the software infrastructure of operating systems, data management, and so on, and the applications architecture. The next chapter shows how the technical architecture can be managed to reduce the complexity, and the cost, of a multivendor environment.

Chapter 3

Managing the technical architecture

The problems arising from the need to integrate applications in different software environments affect the overall costs of a multivendor environment, the timescales involved in developing new applications, and thus the quality of the service that the systems function provides to its users. The image and standing of the systems function within the organisation can suffer as a result. Any systems manager who has had to explain to a business manager why it is not easy to transfer data from one computer system to another will know just how serious a difficulty this can be.

The difficulties of sharing data between applications in different software environments, updating and enhancing such applications, and facilitating interworking between them, are all determined by the organisation's overall technical architecture. Managing this architecture is critical to the successful management of a multivendor environment.

In this chapter, we describe the components of the technical architecture that must be defined in order to reduce complexity, to facilitate integration, and to make the multivendor environment as transparent to users as possible. Standards are, of course, essential to solving integration problems, but by themselves, they are not sufficient for a solution, and they are certainly not a panacea. Systems departments must therefore be prepared to invest substantial time and effort in defining and maintaining an appropriate technical architecture if they are to provide the most cost-effective service in a multivendor environment.

The technical architecture includes hardware, software, and applications

The integration problems arising from a multivendor environment can be minimised by defining the three main components of a technical architecture — the hardware architecture (including the network structure), the software infrastructure, and the applications architecture. The elements of these components are summarised in Figure 3.1, overleaf. Several elements have been the subjects of previous Foundation reports.

Most Foundation members recognise the importance of such a technical architecture, and have defined and implemented one, sometimes with the help of outside consultants. Figure 3.2, on page 21, describes the rationale of some of the many systems functions that have recognised the need to define an appropriate technical architecture. The Appendix describes the technical architecture designed by one of our Foundation members, the Commission of the European Communities. The Commission has a well defined multivendor policy, with four major mainframe suppliers in its central computer centre, and several more vendors for departmental and personal computing facilities. Its technical architecture, which it calls 'informatics architecture', includes the three main components outlined in Figure 3.1, and has helped the Commission to cope with the problems resulting from its multivendor policy, and even to exploit its multivendor environment.

The technical architecture should be designed to reduce complexity and to be flexible

The degree of complexity in a multivendor environment increases as the number of different software environments increases. The technical architecture must therefore be designed to manage and, where possible, to reduce this complexity by minimising both the number of different software environments and

Figure 3.1 A technical architecture defines three main components

The technical architecture defines the hardware architecture, the software infrastructure, and the applications architecture. The level of detail depends on the extent to which systems activities are devolved to business units.

The *hardware architecture* defines principles and policies in the following areas:

— How many hardware environments will be used, and which suppliers should provide the different hardware elements.

— Which are the strategic vendors. They are likely to be those vendors whose products support long-term applications that have a significant impact on the business, because they would cost too much to replace, or because they produce large benefits, or because some business areas would not be able to function without them.

— The basic functions and uses of the different hardware environments. For example, one might be used for office automation, while another might be used to provide all other computing facilities.

— Where the various hardware elements are to be located, and who will be responsible for their day-to-day and long-term management.

— The network layout, both for wide-area communication between computer centres and user departments, and for local communications within each site.

— How different hardware environments are to be connected and which gateways or other interfaces will be used to link them.

The *software infrastructure* defines the operational software environment which runs on the hardware environment and in which applications will be supported. It is the central component of an organisation's software strategy, and is described in Report 69, *Software Strategy*. The main components of the software infrastructure are:

— The operating system and development environment.

— Data-management software.

— Communications software.

— User-interface standards and software.

— Core applications, which Report 69 defines as applications that usually maintain data used by more than one department.

The *applications architecture* defines which functions need to be provided, how the applications providing these functions relate to each other, and on which data structures they rely. The applications architecture therefore sets out:

— The core and non-core applications, and the functions they support.

— The mapping of applications onto the hardware architecture and the software infrastructure.

— The applications that need to exchange data.

— The main data structures relevant to the applications.

— The main users of different applications.

the differences between them, so that as much as possible of the applications coding can be independent of the environment. It must, at the same time, be flexible enough to cope with changing requirements.

The number of different software environments required can be minimised by choosing environments that can support the widest possible range of functions, over a wide range of machine sizes. Digital's VMS operating system, for example, spans a wide range of machine sizes, and Unix is now a cost-effective environment for personal workstations as well as for departmental computers. One of the aims of IBM's SAA is to broaden the range of machine sizes that can be covered with one software environment.

It may be possible to overcome the differences between software environments by using software products and development tools designed to be used in more than one environment. Examples include the products and tools available from Oracle, and the Focus family of products available from Information Builders. By writing applications to conform to the 'standard' defined by such products and tools, the differences in the underlying operating systems can, to some extent, be hidden from the applications.

Even if such products are used, however, it is not usually possible to construct applications that are completely independent of the software environment. The position is worse if traditional programming languages are used. If an application is to be used in two different environments (whether for historic reasons, or in order to multisource a particular product type), two different versions of the same application may have to be produced and maintained. In these circumstances, applications should be structured to make any environment-dependent code clearly identifiable and, if possible, to locate such code within one module.

The Commerzbank, a major German retail bank, has managed to reduce the complexity of its multivendor environment by constructing applications that can be used in either of its two main computer systems. It uses both Nixdorf and Olivetti systems to support its front-office systems, running essentially the same applications in each environment. It adopted a modular approach to designing its applications, with the result that 85 per cent of the code can be moved without change between the Olivetti and Nixdorf banking systems. The remaining 15 per cent is in vendor-specific modules that are easily identified, which means that the cost of rewriting the applications for the other system and maintaining two sets of software is minimised. At some future date, Commerzbank's applications will probably need to be converted to a common software environment, possibly Unix, and the modular structure of the programs will facilitate this task. The design philosophy is supported by in-house standards, which are currently being extended to allow applications to be run in the different mainframe environments (the bank uses Siemens and IBM mainframes). These extensions to the in-house standards are expected to take between two and four man-years.

Figure 3.2 Many systems functions have recognised the need for an appropriate technical architecture in dealing with the issues of a multivendor environment

The Australian Stock Exchange is seeking to create a "common operating environment", which it defines as "a coherent set of interworking system software components including operating system, database management system, communications systems, and applications system; the common operating environment may reside on one or more physical computers."

ENIDATA, the systems function of the Italian energy, oil, and chemicals organisation, ENI, often needs to define the architecture of its various client organisations within the ENI group. It is now considering creating a 'macro-architecture' in which it will define the major functions that will be needed for the ENI group (for example, document exchange) and specify the standards with which detailed solutions will need to comply (for example, X.400).

The Information Management Advisory Service of the Department of Finance of the Republic of Ireland also appreciates the importance of a "considered multivendor policy where the use of IT is carefully planned." In such a policy, the emphasis must be on achieving consistency by procuring products conforming to standard architectures.

Commerzbank's conversion plans demonstrate that the technical architecture must be flexible enough to cope with continually changing business requirements and technological changes. On the one hand, users are becoming more demanding, and require new applications and the integration of previously separate applications. On the other hand, technological advances mean that some applications, which in the past were not feasible, can now be implemented. Another complication is that some existing applications, based on technology that is now obsolete, cannot be replaced quickly, and systems functions will therefore have to operate and maintain systems that predate the technical architecture and that are difficult to integrate with it.

The definition of the technical architecture should therefore specify the main stages in its evolution. A technique for illustrating the evolution of the technical architecture is the 'route map', showing how a specific area is planned to develop over several years. Figure 3.3, overleaf, is an example of a route map for the evolution of an organisation's communications systems from a variety of distinct, mainly proprietary solutions, to a more unified solution based on open standards. Depicting the evolution of the technical architecture in this way highlights probable future integration needs, as well as opportunities for using products (such as a gateway between Digital's ALL-IN-1 and IBM's PROFS) as a means of achieving the required integration.

The technical architecture should facilitate integration

Many organisations are finding that there is an increasing need to integrate applications used by separate business functions. A large German manufacturing company, for example, is currently seeking to integrate applications that were decentralised in the 1970s; one difficulty it faces is to standardise the product descriptions used in applications throughout its business functions, from applications supporting scientific research and development, through CAD/CAM and production control, to its commercial applications. There are many other examples of businesses in various industries that have a growing need to integrate applications that currently run in different software environments.

Attempts to integrate such applications often run into problems. Over half of the respondents to our questionnaire considered that managing the technical problems of integrating applications in different software environments caused the most severe problems in a multivendor environment, as Figure 3.4 illustrates. The effect can be serious: nearly three-quarters of respondents said that these problems absorbed a significant amount of resources, or that they could even damage the business.

In a multivendor environment, the number of interfaces between software environments is heavily influenced by the applications architecture. In general, the applications architecture should aim to reduce the number of interfaces. It should also ensure that the interfaces are clear: the data passing across each interface must be clearly defined, as must the functions on either side of each interface. The applications architecture will also influence the level and the type of integration required.

There are two main *levels* of integration, apart from the trivial situation where there is no integration at all between software environments:

Batch interchange: Batch interchanges between applications in different software environments

Figure 3.3 Route maps are a useful technique to illustrate the evolution of the technical architecture

The route map illustrated here shows the planned emphasis on open standards.

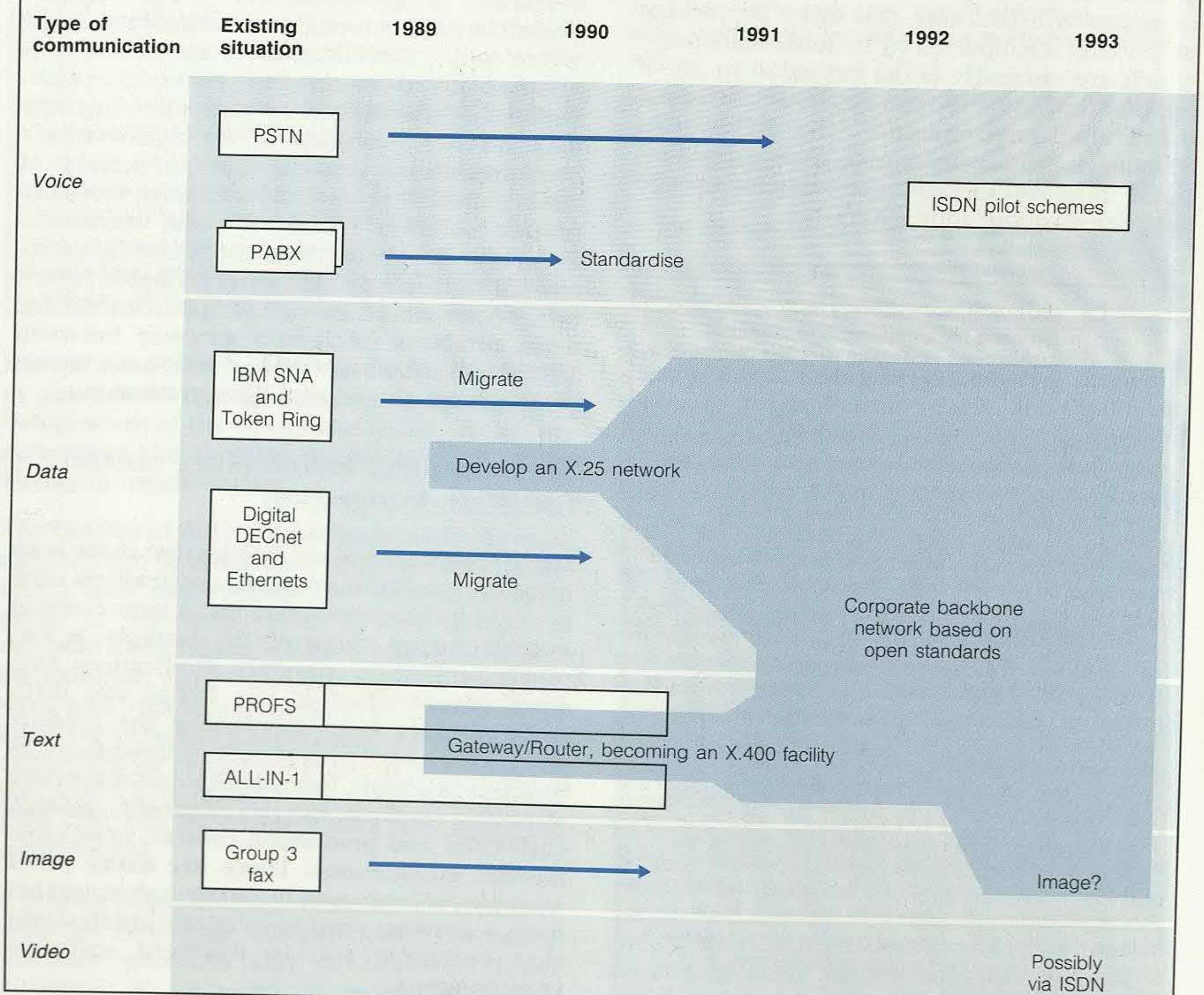
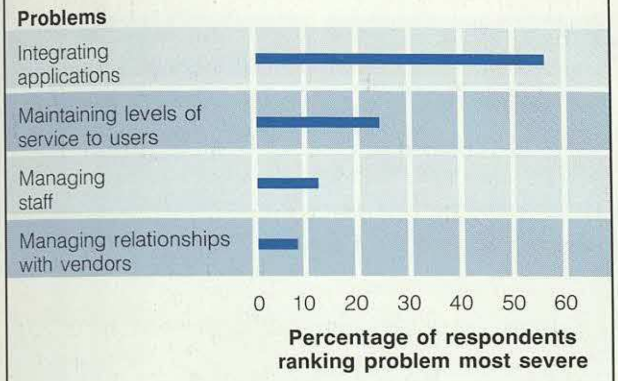


Figure 3.4 Integrating applications in different software environments is the most severe problem in a multivendor environment



(Source: Survey of Foundation members)

are quite frequent in decentralised businesses. There may be operational problems — for example, if there are very many interchanges, or if the interchanges span many time zones — but there are unlikely to be many technical problems. All that is required is a transfer medium (magnetic tape or batch file transfer via communications networks), and perhaps a means of translating the data being transferred.

Realtime interchange: This closer and more complex level of integration is where data needs to be exchanged between applications in different software environments online and in realtime. In these circumstances, not only are a communications network and a translation mechanism required, but a mechanism to coordinate the control of the processes in the different environments is also needed. This level of integration may be required because:

- The information exchanged could not be achieved in the required time by batch interchanges.
- The information needs to be available at many widely dispersed locations, so that batch interchanges would be difficult to handle.
- The information needed is a small, but not predictable, set of data items in a very large data file, and it would not be economical to transfer the whole data file.
- A file or database needs to be accessed and updated with data from applications in different software environments.

There are three different *types* of integration — data-driven, user-driven, and process-driven. They are illustrated in Figure 3.5, overleaf. While the applications architecture will largely influence the number of interfaces between applications and the level and type of integration at each interface, other elements of the technical architecture are critical for achieving the different types of integration. Figure 3.6, on page 25, summarises the critical elements for each type and also shows the means by which each combination of level and type can be implemented.

Data-driven integration

If applications in two different software environments need to share data, and timeliness and update requirements mean that a batch interchange is sufficient, the easiest means of achieving the required integration is by duplicating the data in each environment. The data can be transferred from one environment to the other either by transferring a magnetic tape or by using a file-transfer protocol over a communications link. Neither technique presents much of a problem. Of course, the data structures defined in the technical architecture would have to be made compatible in the two software environments. In the particular case of electronic mail, many large organisations have installed the Softswitch system from Softswitch Inc to interlink mail systems in separate environments.

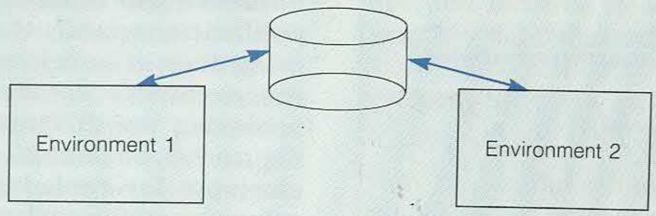
A variety of methods is available if realtime integration is required and it will usually be best to choose the simplest for the particular circumstance. The possible methods include:

- Providing access to the database maintained by one application, preferably using software provided by the vendor of the database management system.
- Placing the data to be shared in a separate file server or, better still, in a database machine.
- Implementing a distributed database management system.

The SQL standard will, despite some limitations and inconsistencies between products, generally provide a good basis for access to either a host database management system or a database machine. However, the standards for distributed database systems are still immature, and this approach should be adopted with care.

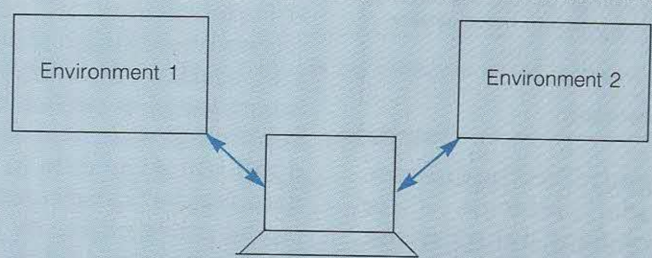
Figure 3.5 There are three types of integration

Data-driven integration: Applications in different software environments need to share the same data



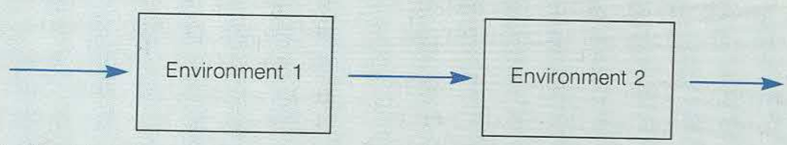
Example: A customer database is accessed from the accounting and invoicing application in Environment 1 and by a marketing application in Environment 2.

User-driven integration: Users need to access applications running in different software environments



Example: A user needs to access a departmental software environment for office systems applications, and the central software environment for accounting applications.

Process-driven integration: A business process is composed of functions supported by different software environments



Example: A withdrawal made through an automatic teller machine (ATM) needs some local processing in the ATM, and some processing in the remote mainframe.

User-driven integration

It is not usually acceptable to have several terminals on a user's desk. The ideal of one terminal can be achieved by providing a facility in one system that allows access to applications in another system. This type of facility is generally known as 'terminal passthrough' or 'remote session access'. Such facilities are increasingly available and are a convenient approach for both users and the systems department. However, this is an expensive approach, because most host computer systems are inefficient when used as switches, and it is appropriate only if relatively few users will require this facility at any time. As the need for user-driven integration increases, it will therefore usually be better to perform the switching and conversion facilities in the network.

Providing switching facilities in the network will be easier if all applications use the same protocols to communicate with workstations. Various proprietary protocols are available, but serious consideration should be given to using the X-Windows protocol, developed at MIT under the sponsorship of vendors including Digital and IBM. Products implementing X-Windows are available and they provide a standard way of giving applications access to windows on a high-quality graphical display. Management of the windows is under the control of the user's workstation. The DECwindows product, for instance, runs in VMS, Unix, and MS-DOS environments and supports the integration on one screen of data from multiple X-Windows applications. In the future, we expect that object-oriented systems will provide an even closer integration of applications in different software environments.

Figure 3.6 The different elements of the technical architecture are critical for the different types of integration

Type of integration	Critical elements of the techical architecture	Means of implementation: Batch interchange	Means of implementation: Realtime
Data-driven	Data management and data structures Communications software	Duplicate data	File-server architecture Database machine Distributed database
User-driven	Network architecture Communications software User-interface standards and software	Not appropriate	Remote-session access Protocol conversion Common user interface
Process-driven	Applications architecture Communications software	File-transfer	Terminal emulation Client-server architecture Remote procedure call Cooperative processing

Process-driven integration

With process-driven integration, the simplest means of achieving batch interchange of data is by online or magnetic-tape file transfer. A satisfactory level of realtime integration can often be achieved by emulating a standard terminal in order to gain access to an existing application. To obtain closer integration or higher performance, it may be necessary to adopt a client-server architecture in which the roles of user workstation and server are formally separated and the machines running the processes communicate using a more flexible protocol, such as SNA LU6.2 or TCP/IP.

Sometimes, process integration can be achieved by using a remote-procedure-call protocol that allows programs to call modules on other systems as if they were loaded on the same system. In many cases, however, the changes that have to be made to applications will be more difficult and time-consuming than creating the inter-process communications link.

The technical architecture should make the environment transparent to users

From a user's perspective, a basic requirement in a multivendor environment is for one terminal that can be used to access all the applications and data, regardless of the software environment in which a particular application or set of data is physically located. It is also important that the user interface (including log-on and log-off procedures, use of function keys, meaning of error messages and the actions expected from the user, and procedures for getting help) is the same for applications running in different software environments.

The requirements for a consistent user interface were discussed in Report 63, *The Future of the Personal Workstation*. The aim is for all the applications to which a user has access to have the same 'look and feel'. The experience of Apple Macintosh users shows the benefits of providing a consistent user interface, and has caused most other vendors (most notably IBM with its SAA Common User Access) to follow suit.

Consistency of user interfaces is, however, difficult to achieve in a multivendor environment. The best way of providing it is for all systems developers to offer similar interfaces, and many members will already have their own in-house standards in this area. The Commission of the European Communities is one of several Foundation members who have adopted this approach. (Its User Agent Service is described in the Appendix.) In most cases, however, in-house standards are not based on the graphical interfaces offered by Microsoft Windows, the Macintosh, and powerful workstations, and will need to be extended to cover them. We expect that industry-wide user-interface standards will be developed over the coming years, based on interfaces such as X-Windows. In-house standards, as well as bought-in software, should be consistent with these standards.

As yet, there are few products available to help in the construction of consistent user interfaces. However, it may be worth considering whether special software in the workstation can meet the need. The most sophisticated product of this type is probably the Proteo product from Proteo Software Inc, which provides a broadly consistent user interface to more than 120 applications available for IBM (and compatible) PCs. Proteo treats documents and data structures maintained by applications as objects. It provides a standard set of operations (such as 'examine' and 'mail') for all objects, and these are implemented by a mixture of Proteo modules and calls for the applications.

Standards are useful, but they are not a panacea

It is tempting to believe that internationally agreed standards will make it easier to integrate applications running in different software environments. Respondents to our questionnaire certainly expressed a high level of interest in standards, although there was no overall agreement about their usefulness, as can be seen from Figure 3.7.

Some areas already benefit from standards

At their present stage of development and application, standards are most useful for interlinking computer systems so that data can be transferred between them. X.25, Ethernet, SNA, and other standards are all useful in this respect, and hardware and software products to implement them are available for many software environments. The nature of the link required and the hardware involved will usually determine the most suitable choice of standards.

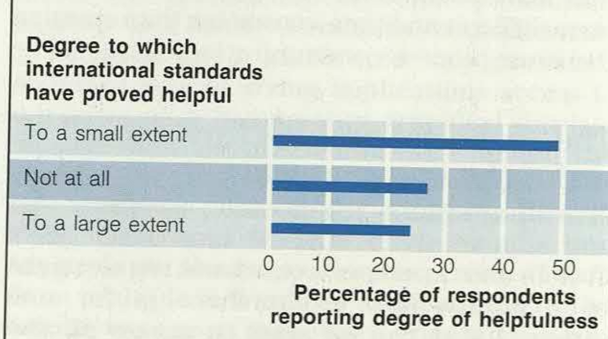
Other areas also benefit from standards. Database access in a multivendor environment can be made more consistent by using SQL, for example, and in Unix environments, the Common or Open Applications Environments proposed by X/Open and OSF will help reduce the complexity arising from using Unix versions from different vendors.

There is a multiplicity of established and emerging standards

Standards, however, are not a complete answer to integration problems in a multivendor environment. One of the main difficulties stems from the fact that there can be several versions of the same standard. Thus, the X.25 specification permits different implementations, so that the technical architecture may need to specify in more detail which version of X.25 is to be used for communications links. Before products conforming to X.25 are procured, they should therefore be tested to confirm that they can actually interwork. Such interworking tests are carried out systematically for the products of several suppliers by EurOSInet, an organisation sponsored by IT suppliers and other interested organisations. Similar problems occur when a basic standard (SQL, for example) is extended and enhanced in different ways by different vendors, so that full compatibility across different software environments is unlikely to be achieved.

In several areas, standards are not sufficiently widely adopted to facilitate easy integration. Network management is one particular area in which integrated standards are still being developed. The lack of open network-management standards is a major concern for many Foundation members, especially those with a multivendor environment. Report 65, *Network Management*, addresses this issue in detail. User-interface standards are also still being defined, as we discussed above. Nor are data-dictionary standards sufficiently developed to be able to link different data-dictionary systems or to exchange data between them.

Figure 3.7 Respondents have a mixed view of the degree of help offered by international standards



(Source: Survey of Foundation members)

While standards in these areas are likely to evolve, other areas where compatibility between interworking applications is needed will probably never be covered by standards. One such area is concerned with the exact definition and the ownership of data items in different software environments. Differences in the update cycles of data items can also cause problems. These problems must be resolved in the applications architecture. In some instances, standards may help, but ultimately, it is an organisation's applications architecture that must ensure compatibility of data structures and functions between applications in different software environments.

Systems functions should take positive action with respect to standards

The multiplicity of established and emerging standards means that each systems function needs to define its own standards policy, setting out the standards (including any variants) on which it is basing its technical architecture. This is particularly important in multivendor environments because the option of standardising on one supplier's proprietary architecture will no longer be possible. In order to define an appropriate standards policy, systems functions need to understand how standards are developing. Several organisations therefore have 'standards watchers', whose role is to track the development of standards. This activity is becoming more important, because standards are evolving rapidly. The situation is complicated further by the large number of standards-making organisations now active in this field.

Many systems functions could take a more active role in influencing the evolution and formulation of standards. They should identify the areas in which standards are important to them, but which are not yet standardised. They should discuss their standards policy with vendors and include standards issues in the evaluation criteria for tenders.

As well as deciding which external standards to adopt, systems functions also need to specify in-house standards as part of their software strategy. The in-house standards policy will refine the choice of external standards, by deciding which variants will, or will not, be used, and by setting in-house standards where there are no suitable external standards. The in-house standards must, of course, be documented, distributed, observed, and reviewed periodically.

In the next few years, there will be a continuing development of standards. Nevertheless, we do not expect the role of standards in the management of a multivendor environment to change significantly. Network-management standards, user-interface standards, and data-dictionary standards can be expected to be agreed, and products supporting them will appear, facilitating integration. However, many standards will continue to permit variants, and many integration problems will continue to arise in areas that are not covered by standards, but that must be tackled by each organisation's technical architecture.

Considerable effort is required to define and maintain the technical architecture

Defining and implementing a technical architecture that will reduce the complexity, and hence the cost, of the multivendor environment will take a lot of time and effort. A poorly designed technical architecture may greatly increase the problems of integrating different software environments. Business pressures for the rapid development of new systems may encourage *ad hoc* solutions. They are rarely satisfactory in the longer term. Once the immediate business pressures have been relieved, systems functions must find the time to develop a sound technical architecture.

Defining the technical architecture will, of course, involve staff with a good understanding of the technical issues. Users should also be involved, because they may well have a better grasp of how the technical architecture can support their business needs, as should vendors, whose product plans will probably be crucial to the implementation of the technical architecture.

A wide range of business and technical skills is required

In our consultancy work, we have found that defining and implementing a technical architecture can take several man-years, even though the consultants are experienced in this kind of work and have a detailed understanding of developments in standards and technology. This experience is shared by the organisations we talked to during the research for this report. In general, a systems director should expect the development of the technical architecture to require at least two to three man-years of effort, spread over a year's elapsed time.

The skills required to define and maintain a technical architecture differ from normal systems development skills. Business skills and technical skills are both needed. Report 49, *Developing and Implementing a Systems Strategy,* gives guidance on the mix of business and technical skills required. It is important to have a clear view of how the business is likely to develop, and to understand how different business functions could and should be supported by information systems. On a technical level, the development of a technical architecture requires a broad understanding of the development of technology markets, of the impact of standards, and of different products and the roles that they can play. In a multivendor environment, this understanding needs to be particularly broad, as several architectures and vendors' products will need to be integrated.

Users and vendors should also be involved

The technical architecture should not be defined by systems staff alone. Users can often see the business implications of new technologies more clearly than systems staff, and should be included in discussions about proposed technological developments on an 'equal partnership' basis. The most common way of involving users is through joint systems-strategy groups. Some organisations also involve user departments in formulating invitations to tender and evaluating the tenders. Aachener und Münchener Informatik Services GmbH, the independent systems function of a major German financial services company, makes a point of involving user departments in the reviews of its IT scenarios, which it carries out every two years. In total, 10 to 15 people are involved, including systems and user staff, and the exercise requires two to three man-years of effort.

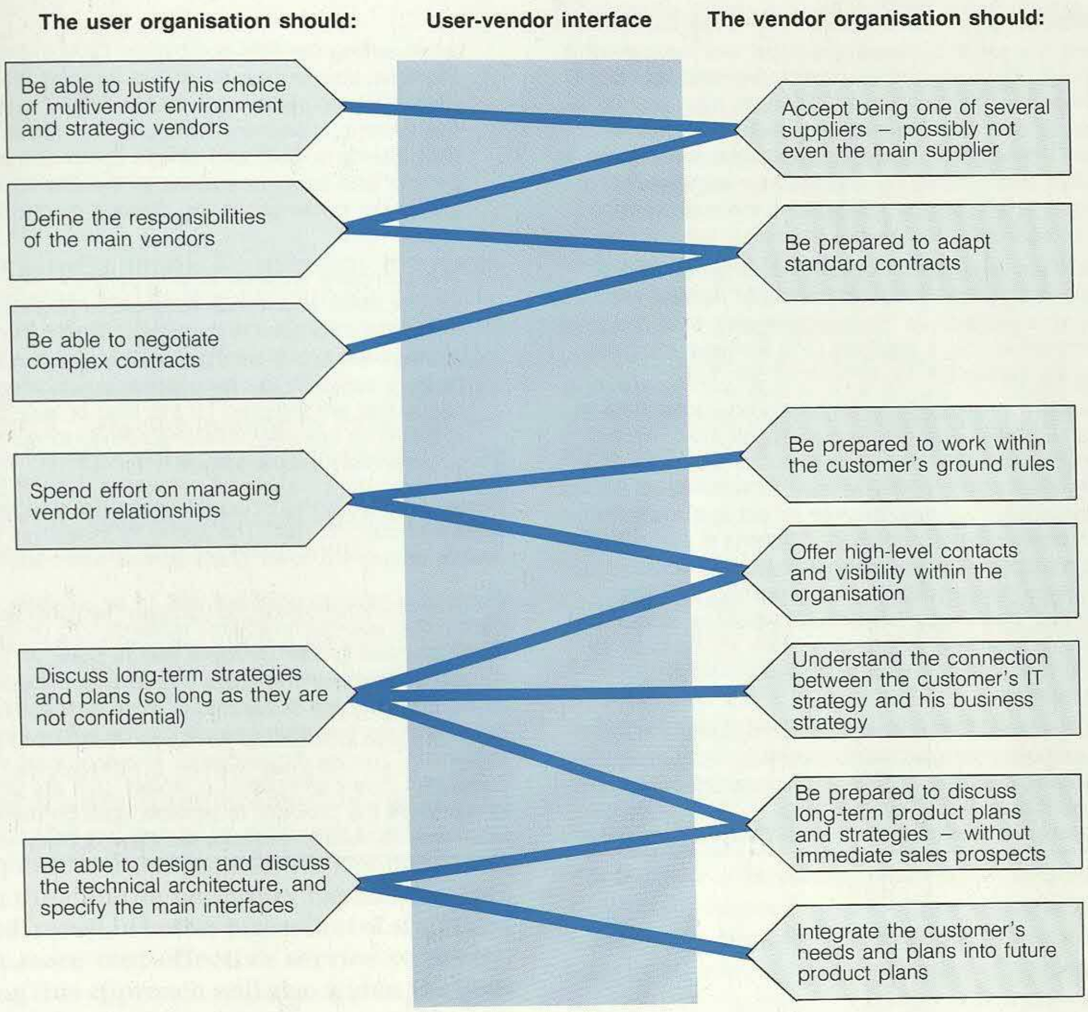
The technical architecture will usually imply restrictions on the use of certain suppliers and types of products. Business units with their own IT facilities and skills may well wish to procure equipment that does not conform with the technical architecture, and may find such restrictions difficult to accept. Involving such business units in the definition of the technical architecture has two main advantages. First, it ensures that the architecture can evolve to accommodate new business requirements, and second, it helps the business units to accept that they may need to restrict their choices in order to be able to integrate their applications with those of other business units.

The main vendors (and potential vendors) should also be involved in the definition of the technical architecture. A dialogue should be established so that the systems function can learn about current and planned products, and the vendors can understand the organisation's requirements. Vendors will be able to suggest the most appropriate products from their range, and may even propose developments in their range to meet an organisation's requirements. Figure 3.8 illustrates the nature of the dialogue that should take place between the systems function and vendors. A series of regular and *ad hoc* meetings and briefings should be held with the vendors, ranging from operational meetings to resolve day-to-day issues, to strategy meetings involving senior managers or even board members in the two organisations.

This partnership approach will mean a change in attitude for many vendors, who would traditionally have set themselves the objective of becoming the single vendor and of controlling the account. As multivendor environments become more common amongst their major customers, many suppliers, especially mainframe suppliers, will have to adapt their expectations and accept that another vendor may continue to be an organisation's main supplier for the foreseeable future.

This change of attitude is not easy, as the experience of Siemens, the major German electrical, electronics, communications, and computer manufacturer and vendor, shows. Siemens is currently reorganising its data and information systems business. It is making a concerted effort to offer better support to its customers who operate in multivendor

Figure 3.8 A working partnership in a multivendor environment requires a dialogue between user and vendor organisations



environments where Siemens may not be the main supplier. Its experience is described in Figure 3.9, overleaf.

Although it is important to involve vendors in defining the technical architecture, it is usually unrealistic to expect them to take on the responsibility for this activity. (The exceptions are where one vendor is to be responsible for systems integration, or where a single-architecture environment, based on one vendor's proprietary architecture, is to be implemented.) Vendors will usually be willing to help resolve any difficulties, but ultimately, the systems function is responsible for making sure that the products from the different vendors can work together.

At several points in this chapter, we have stressed the need for the systems function to be prepared to take on more responsibility in a multivendor environment than in a single-vendor environment. In the next chapter, we address this and other staffing issues, and show that many of them can be tackled by creating a team of systems staff with a range of skills that makes it feasible for them to work in more than one software environment.

Figure 3.9 Vendors, as well as users, have to adapt to working in a multivendor environment

Siemens AG

Siemens is a major European producer and supplier of IT equipment. Since most of its major customers also buy IT equipment from other suppliers, Siemens has recently re-oriented its data and information systems business to provide IT solutions in specialist application areas (such as distributed data processing and software engineering) to customers where another company is the main vendor. This new approach has several significant implications:

- The sales organisation needs to change its outlook from an aggressive stance (aiming to replace non-Siemens products at the customer site), to a cooperative stance (seeking to fit Siemens products into the framework set by another vendor).
- The product policy must be based on connectivity to other systems, and the product range must contain smaller systems and workstations that can work in a networked environment, where the mainframes are not provided by Siemens. In order to achieve maximum connectivity for its customers, Siemens is committed to the principles of open systems (including OSI). However, it will also provide connectivity to other vendors' proprietary network architectures, specifically SNA, either by providing compatible products or by using open systems standards, where this is appropriate.
- The company is preparing to offer support and maintenance for a complete installation, including systems from other suppliers. Siemens believes that it is well placed to do this, because it still has extensive in-house experience of the IBM architecture, acquired when selling the IBM-compatible 7800 mainframes. Siemens also has experience of working in large, multivendor projects such as the START system. In this system, a Siemens computer links travel agents to IBM, Siemens, Bull, and Unisys computers operated by tour and travel operators, so that the travel agents can make bookings on the different machines from one terminal.
- The company must be able to communicate effectively with its users about their business strategies, and the implications for their IT strategies. Working on IT strategies with customers obviously requires Siemens to have staff with the experience, qualifications, and status that are required for this kind of task. Siemens is therefore currently investing heavily to build up an appropriately qualified team.

Siemens is also considering reviewing the incentives for sales and support staff. The aim is for incentives to be flexible enough to cover goals agreed with customers.

Siemens is keen to point out that, to be effective, the partnership approach requires customers to be open about discussing their business and IT plans. In some situations, Siemens envisages shared pilot projects in which products are developed in close cooperation with customers, so that the products can be tailored to the customers' precise requirements. These projects will be successful only if competent customer staff are available to represent the projects, to provide input on the technical requirements, and to evaluate technical options.

Chapter 4

Managing staffing issues

Not surprisingly, many Foundation members found that staffing the systems function is more difficult in a multivendor environment. In particular, because staff may have specialist skills that are relevant to only one software environment, systems managers have limited flexibility in allocating staff according to the needs of the business. Ideally, systems staff should have a range of skills that enable them to work in any one of the environments.

It is not, of course, easy to build a team of systems staff that can be used in such a flexible way. In Report 71, *Staffing the Systems Function*, we pointed out that it will become more and more difficult to recruit suitably qualified staff. It will be especially difficult to find people experienced in a range of software environments. Recruiting staff with the *potential* to perform the broader roles required in a multivendor environment, and training them in the necessary capabilities and skills, will result in better utilisation of staff (and hence a more cost-effective service to users). Adopting this approach will also widen the pool of potential recruits, and improve staff competence and motivation. There will be barriers to be overcome in adopting this approach, but in view of the fact that continuing personnel costs are by far the biggest cost item in a multivendor environment (as we saw in Chapter 2), it is essential that it be implemented if costs are not to increase to an unacceptable level.

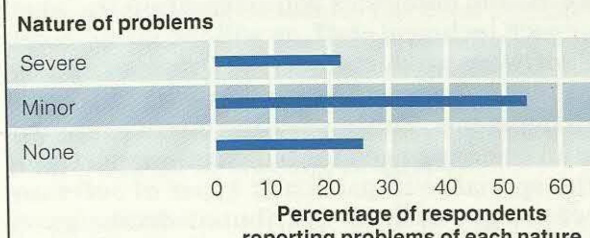
Aim for a more flexible allocation of staff

Three-quarters of the Foundation members who responded to our questionnaire had problems allocating staff in a multivendor environment (see Figure 4.1). One frequently mentioned problem was that of a project being delayed because staff with the required skills were assigned to other projects. These problems are exacerbated where the project involves some measure of integration between software environments. Staffing problems also arise where managers are reluctant to allow their staff to work outside their own particular unit. Such managers often succeed in building a team of people who are specialists in one software environment, but who cannot be used elsewhere.

Some of the organisations we talked to have gone quite a long way towards developing more flexible teams. ENIDATA, whose experience was described in Figure 1.5, claims that 75 per cent of its staff are fully competent to work in several software environments. Through training and job rotation, Commerzbank, a major German bank, aims for as much flexibility as possible in the way in which it allocates staff to projects.

The degree of flexibility that can be achieved will, however, depend on the type of systems staff concerned. Figure 4.2, overleaf, summarises the extent to which different types of systems staff can be used in more than one software environment. We recommend that, in a multivendor environment, systems directors should aim to employ as many staff as possible who can work

Figure 4.1 Three-quarters of respondents reported problems with staff allocation in a multivendor environment



(Source: Survey of Foundation members)

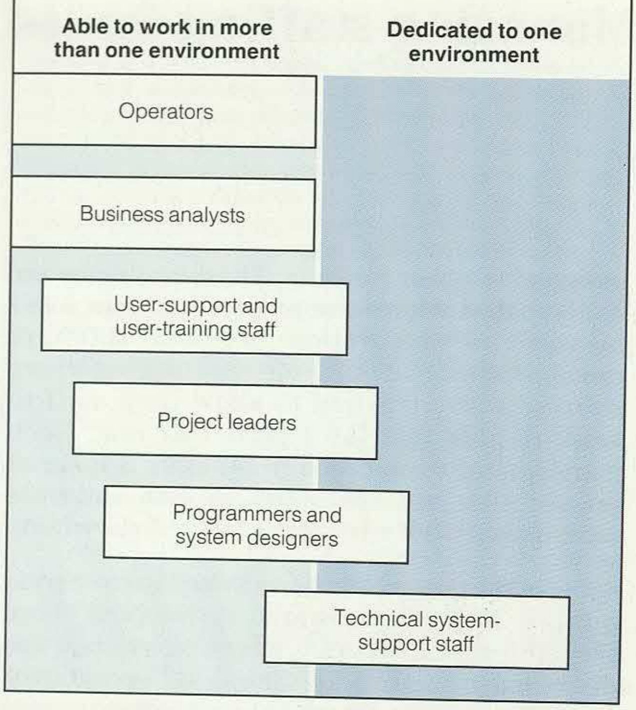


Figure 4.2 Most types of staff can be assigned to work in more than one software environment

in several environments, with only technical-support staff, and some specialist programmers and systems designers, dedicated to a single software environment. The ideal staffing profile of the systems department will therefore be a 'broad T-shape', which means that the majority of staff will have a broad skills base and will be competent to work in several software environments. They will be backed up by a smaller number of technical experts, each of whom specialises in a particular aspect of one of the software environments.

In the future, however, even technical-support staff will need to be able to work more flexibly in a range of software environments, as it becomes more common to integrate applications in different environments. The emergence of distributed databases will strengthen the need for such technical staff, as will the growing use of software products (such as Oracle or Ingres) that can be used with several different environments. The implication is that there will be an emerging need for technical-support staff who specialise in particular types of software technology, such as distributed databases or Unix. This new type of technical-support staff will complement the existing ones who support a particular vendor's hardware products.

Having staff who can operate flexibly in more than one software environment will not only make it easier for the systems director to allocate work according to the demands of the business. Our research indicates that other advantages will also accrue to the organisation:

— *Increased staff motivation:* Many organisations have found that developing systems staff so that they can work effectively in several software environments increases staff motivation. The experience of Electricité de France/Gaz de France, the French electricity and gas utility, is described in Figure 4.3.

— *Increased loyalty:* Ensuring that staff have a flexible range of skills helps to prevent too close an association with the equipment or products of a particular vendor.

— *Lower rate of staff turnover:* Several Foundation members reported that staff turnover was lower amongst staff who had been trained to work in several software environments.

— *Enhanced general level of skills:* One German organisation reported that the enhanced level of skills of its systems staff was particularly useful in negotiating with suppliers, because systems staff were more clearly aware of the strengths and weaknesses of the products being offered. Staff also found it easier to implement packages because they were already used to dealing with products from different vendors.

— *Easier recruiting:* Where there is an opportunity in the organisation to learn about new areas, potential recruits have an incentive to join.

— *Broader source of potential recruits:* While it is unlikely that applicants will be skilled in all of an organisation's software environments, there will be larger numbers of applicants with expertise in at least one of the environments.

Recruit and train staff for more flexible roles

In Report 71, we pointed out that most systems departments will find it increasingly difficult to recruit sufficient numbers of qualified staff. Systems managers will therefore need to change the emphasis of their recruitment efforts to

Figure 4.3 Job rotation in a multivendor environment can have a marked effect on staff motivation

Electricité de France/Gaz de France (EDF/GDF)

This enterprise is a public utility supplying electricity and gas services in France. It has revenues of $26,713 million, and employs 150,000 people. Each service has its own separate organisation, but there are three central divisions — distribution, administration, and personnel. There is also a central Information Systems Committee, with responsibility for strategy, policy, and coordination of EDP and telecommunications.

The enterprise runs a large number of computers, all of which may be linked via its own telecommunications network, which operates on private lines, based on France Telecom's Transpac X.25 network. EDF/GDF has:

- Over 200 mainframes (Cray, IBM, and Bull).
- 2,350 minicomputers (mainly Bull).
- Over 12,000 personal computers.
- 700 process-control computers.
- 250 engineering workstations.

To meet the needs of such a large organisation for a very diverse range of applications, EDF/GDF's general policy is to purchase from a variety of vendors. While no detailed cost comparisons are available, the Information Systems Committee is convinced that the multivendor environment has resulted in substantial savings. As a major customer of several suppliers, it is in a strong position to negotiate at a very high level in their organisations, and to adjust the existing balance of purchases as it sees fit.

The multivendor environment has had one of its most marked effects on staff motivation. The 3,500 staff in information management are allocated to specialist teams (for example, a team dedicated to Bull systems, or a team dedicated to IBM systems), in operations, support, and systems development. However, the unofficial policy of the central personnel division and of the specialised service departments is to have as many multiskilled staff as possible, and there are plans to move staff between teams. Initially, the teams dedicated to Bull and IBM systems were trained by the respective suppliers. Now, EDF/GDF runs the main training sessions for both suppliers' systems in-house.

The ability to offer staff the opportunity to work on new systems from different suppliers is considered to be a critical factor in motivating and retaining staff. Turnover of information systems staff is very low. The multivendor environment also provides more opportunities for advancement for high-level systems engineers, who, at certain levels in the career structure, need to change function in order to move on to a higher salary.

The enterprise is pleased with the multivendor environment in which it operates. It can handle the integration problems through network design and standardisation on operating systems, and it enjoys substantial cost savings and a marked improvement in staff motivation.

employ staff with the *potential* to acquire the necessary skills. On-the-job training and job rotation have as important a role to play as formal training.

Concentrate less on technical skills

Many organisations have recognised the need to recruit staff with potential. As Hoskyns, the UK software house, put it, "We need to attract the cream". It is also reflected in the attitude of the many organisations who recruit only graduates, because they see a degree as indicating that the applicant has the ability to acquire new skills.

As the shortage of new graduates becomes more severe, other types of candidates will need to be considered. In particular, organisations with multivendor environments will need to set new criteria for selecting staff with the potential to be trained. This means that they will need to spend less time assessing candidates' technical knowledge and more time evaluating whether the candidates have a good understanding of the principles behind the technical details, and a flexible attitude to their work.

Selecting candidates with potential requires appropriate recruitment procedures, such as the use of personality measurements, job and personnel specifications, and regular reviews of the success of the recruitment process by monitoring the careers of recruits. We provided detailed advice on how to improve the recruitment process in Report 71.

Develop staff to fulfil their potential

Once a promising candidate has been recruited, his or her potential should be developed to fit into the staffing plan of the systems function. This will require a policy of rotating staff between software environments, backed up by a well-designed training programme.

Staff rotation

Providing staff with the opportunity to work in several software environments can be achieved in one of two ways:

— By moving staff between organisational groups, each of which specialises in one environment. For example, there could be one development group for, say, an IBM-MVS environment, and another for a Bull GCOS8 environment.

— By creating an organisation that is not oriented to specific software environments. In this case, there would be one development group, each member of which would be expected to be able to work in all the environments.

These two approaches can be mixed. Thus, there may be a single operations team for all the environments, but several technical-support teams, one for each environment. Figure 4.4 provides guidelines on when one approach or the other would be more appropriate.

Systems departments with teams not dedicated to specific software environments have little difficulty ensuring that staff gain broad experience. All they need to do is to ensure that an individual does not come to be seen as 'the expert' in a particular environment, preventing him or her from being assigned to work elsewhere. If the systems department has teams dedicated to each software environment, staff must be helped and encouraged to move between teams, perhaps once every two years. Many of the organisations we interviewed recognise the desirability of such a policy, but far fewer achieve it in practice. One organisation that encourages this type of staff rotation is Electricité de France/Gaz de France. We described its experience in Figure 4.3.

Training

Our research for this report and our consultancy work indicate that it is important to maintain a learning culture in the systems department. This means that systems staff should expect to learn from training courses and from job rotation throughout their careers. In this way, it should be possible to ensure that systems staff are provided with the skills to enable them to work effectively in a range of software environments. Such an approach has two main implications for the organisation of training activities:

— The training budget is likely to increase, and individual staff will need to spend more of their time on training activities. Report 71 provided guidance on the amount of training appropriate for systems staff.

— New sources for training will need to be considered. In the past, many organisations have relied on the training courses provided by their main vendor. In a multivendor environment, this approach is less appropriate, because the courses will be too product-specific and are unlikely to be objective about the products.

One frequently mentioned difficulty is that older employees often appear to be less easy to train than younger employees. As a consequence, some organisations prefer to recruit younger staff, and train them, rather than retrain their older employees. Other organisations have different experiences. The UK Post Office is reported to have retrained its NCR and Burroughs mainframe programmers successfully, to work on IBM's AS/400. An organisation we talked to during the research found that there was much prejudice against older employees being retrained, and that the employees themselves had little confidence that they could learn to work in a new software environment. In practice, however, it found little difference in the performance of the older employees and that of their younger colleagues, who were also learning the new environment.

Figure 4.4 Each of the approaches to providing staff with the opportunity to work in several software environments is appropriate in particular circumstances

Separate teams for separate environments, with job rotation between teams	Single team with staff qualified to work in several environments
Technical support, or any other area requiring detailed knowledge of the environment	Operations staff and business analysts
Systems functions where there is little integration between the environments	Systems functions where the different environments are closely integrated
Systems functions that do not, as yet, have many staff qualified to work in more than one environment	Areas that already have many staff qualified to work in several environments
Very large systems functions, with several hundred staff	Small- to medium-sized systems functions

Recognise the barriers to staffing flexibility

Many of the organisations we met during the research had achieved only limited success in broadening the skills of their staff in their multivendor environments. The barriers they came up against can be grouped into three categories: organisational, motivational, and managerial.

Organisational barriers

Organisational barriers can arise both within and outside the systems function. Outside the systems function, the problem is most pronounced in organisations that have, say, a research and development function with its own computing facilities, based on a different software environment from that used in the systems function. In such organisations, we found evidence of conflict and lack of cooperation between those responsible for the two software environments. It will not be possible to move staff freely between the environments until these 'organisational politics' have been sorted out.

Within large systems functions, staff are often organised into groups that specialise in a particular software environment, which obviously makes it more difficult to move them from one environment to another. This was the situation at Barilla, one of Italy's largest manufacturers of food products. Its systems function used to be split into two distinct structures (one for its Digital environment and one for its IBM environment). A 'Division 1' and 'Division 2' mentality had started to develop; Barilla wanted to get rid of this, to create a structure where attention would be focused on the applications, and to make it possible to integrate applications running in each of the environments. It has recently re-organised its systems function to create functional groupings, which has aligned it more directly with the structure of the business units, and removed the barriers between the IBM and Digital environments.

The flexible use of systems staff can also be inhibited when staff are assigned long-term to development projects or to maintenance. We have described earlier the virtues of job rotation in a multivendor environment; the area of maintenance is no exception.

Lack of motivation

While many Foundation members have found that a multivendor environment increases staff motivation, others have systems staff who are reluctant to broaden their skills and move away from a software environment with which they are familiar. One of the biggest obstacles to such movements of systems staff is the way in which they have traditionally been promoted. In many systems functions, promotion still depends on gaining technical expertise in one particular environment. As we pointed out in Report 71, career planning based on this criterion is appropriate for technical experts. It is inappropriate for other types of systems staff, for whom promotion should be based on breadth of experience as well as detailed technical expertise.

Salary levels can also inhibit the movement of staff between software environments. For example, UK salary levels for staff with ICL experience are lower than the average; in the Netherlands, salary levels for staff with IBM experience are slightly lower than the average for staff with equivalent experience in other environments. Such salary differences will be difficult to maintain in an organisation with a policy of moving staff between environments. To attract the more highly paid staff, it may be necessary to raise the general level of systems salaries slightly higher than would be necessary to attract and retain other staff. This is unlikely to cause a problem for most organisations, because the salary differentials are quite small. It may also be worth considering increasing the salaries of staff who become qualified to work in more than one software environment.

The 'softer' personnel problems, however, are more difficult to overcome. One organisation we talked to was decentralising its systems activities and was in the process of redeveloping applications for a different vendor's software environment. This was having a very serious impact on its staff. The problem arose because systems development staff were required to work over a long period of time alongside much better paid contract staff, who were being used because of the extra workload during the conversion and redevelopment period. As a consequence, the in-house staff became completely demotivated. The younger staff could not see a future with the organisation and left, thus aggravating the resources problem. The older and longer-serving staff did not leave, but their effectiveness was severely diminished. In these situations, it is very important to devise training plans that demonstrate to staff that they have a continuing role to play, and to keep everyone fully informed about their long-term future.

Introducing a new software environment can also cause those staff working with the 'old' environment to become demotivated. This is a problem faced by Bassani Ticino, an Italian manufacturer of electrical components. Although it has found that its multivendor environment (which comprises a Unisys mainframe, Digital Vaxes, and Apple Macintoshes) has, in general, increased the motivation of its systems staff, some of the staff working in the older mainframe environment feel that they are missing out. Similar problems were reported by a Belgian manufacturing organisation with staff developing applications for older computer systems. Both of these organisations are tackling the problem by training staff wherever possible to work in more than one software environment, and encouraging staff mobility. Emphasising the business importance of the 'older' applications can also play an important role in motivating staff.

Lack of management commitment

From the preceding discussion, it is clear that systems management must be prepared to devote time and effort to career-development plans, if the barriers to staff mobility and flexibility are to be overcome. Some groups of employees, such as the older and longer-established members of staff, will need particular attention.

In practice, however, there may be conflicting priorities that make it difficult to release staff from urgent tasks so they can attend a training course or be re-assigned to gain experience in a different software environment. Project deadlines and the lack of replacement staff are powerful arguments against job rotation, even in organisations where the principle is well established. In other cases, managers will go to considerable lengths to 'protect their own empire'. Systems managers must therefore be committed to implementing the policies that will ensure that their staff acquire the range of skills and experience to allow them to be used effectively in a multivendor environment. Figure 4.5 summarises the actions that will ensure the right mix of vendor-independent and vendor-specific skills.

Ensuring that the majority of systems staff can be used effectively in several software

Figure 4.5 Systems directors must ensure that the right mix of skills is available in a multivendor environment

Make sure that staff understand the rationale for a multivendor environment and that all the different environments are important to the running of the organisation:

— Avoid one system being perceived as the 'second class' system.

— Show management commitment to all machine and software environments.

Aim to create a 'T-shaped' staffing profile for the systems function, with the majority of staff having a broad base that allows them to work effectively in several software environments, together with a small number of environment-specific technical experts:

— Establish staffing plans for each software environment.

— Identify areas where specific technical expertise is required.

— Develop and promote staff with the skills to work in more than one software environment.

— Set staff development objectives for systems managers.

— Make sure the salary structures are appropriate for a multivendor environment.

Recruit staff who have the potential to acquire broad environment-independent skills:

— Emphasise potential in the selection process.

— Look for understanding, as well as knowledge.

Set up and implement appropriate training and staff-development plans:

— Make sure that training plans and job-rotation plans are consistent.

— Make sure that training is carried out early enough.

Use job rotation to broaden people's skills:

— Organise the systems department as an environment-independent function, if possible.

— Plan long projects to allow for staff rotation.

— Plan to rotate staff into and out of the maintenance function.

— Do not allow technology changes to become barriers to staff mobility.

environments is a necessary, but not sufficient, condition for ensuring that the most cost-effective service is provided to users in a multivendor environment. The systems team must also ensure that the operations-management procedures are defined and implemented in such a way that users are unaware of the differences between the environments. In the next chapter, we discuss the most important operational procedures. Most of these are not unique to a multivendor environment — but in a multivendor environment, the penalties for not following them can be much higher.

Chapter 5

Organising operations management

The operations department is in the front line in a multivendor environment. It has to make the multiplicity of systems transparent to the users, it has to run an efficient service in a complex situation, and it has to manage the sources of supply for the IT products and services that are essential for the efficient running of the business.

Make the multiplicity of systems transparent to users

The need to make a multivendor environment as transparent as possible to the users was discussed in the context of the technical architecture in Chapter 3. Operational support can also be organised with this objective in mind by providing users with a single help desk, and standardised service-level agreements.

A single help desk is required

The principle that each user should have one help desk to telephone is well known, but in a multivendor environment, the work of the help desk is complicated by the need to provide a unified service for all software environments. This has implications for the way in which the help desk is organised:

— The help desk must be staffed with broadly skilled staff so that problems arising in any of the software environments can be handled effectively. In this way, a balance can be struck between the requirement to have a single contact point, and the aim described in Report 65, *Network Management*, of having 80 per cent of all calls to the help desk answered without the problem being referred.

— A higher level of skills and understanding is required by help desk staff, who have to be able to analyse a problem in order to know who should be contacted if they cannot resolve it themselves immediately. To achieve this, there must be an effective means of disseminating up-to-date information to help desk staff, so that they can decide who to contact for expert assistance. There must also be good procedures, to which the help desk staff must adhere.

— Help desk staff must have rapid access to an inventory that includes the name and telephone number of the user, identification of terminal device, and details of the applications used and software routines installed at the workstation. A user with a problem may not know which syster [illegible] multivendor environment he or she i[illegible], nor whether the application is running on the local PC or on the mainframe, nor even the correct name of the application being used. In this situation, help desk staff need to be able to look up the relevant details in the inventory.

Service-level agreements should be standardised

In many organisations, service-level agreements are negotiated between the systems department and the user departments, specifying, for example, scheduled up-time and service availability. In a multivendor environment, the number of different service-level agreements with the user should be kept to a minimum.

To provide the level of service stipulated in such agreements, the systems department will need to have corresponding agreements with external suppliers — for example, hardware maintenance providers. It may, therefore, be limited in the extent to which it can offer its users standard agreements by the unwillingness (or inability) of external suppliers to vary their standard conditions of support.

Ensure that good operations procedures are in place

Most systems departments have standard procedures — for example, to get a terminal connected to the computer, to report problems, or to produce accounting information. These procedures should, as far as possible, be identical for the various systems in a multivendor environment. A lack of standard procedures will be more confusing and frustrating for the user in a multivendor environment than in a simpler environment.

The basic procedures for operating in a multivendor environment are no different from those for operating in a single-vendor or single-architecture environment; they are, however, applied to a more complex set of circumstances, and the problems arising from failure to follow standard good practice are likely to be more serious. This means that the discipline in applying the procedures needs to be more stringent. Certain areas need particularly close attention — configuration management, contingency provision, and security.

Keep close control of the configuration

The technical architecture discussed in Chapter 3 covers a wide range of elements. Managing these elements and their evolution means keeping a detailed inventory of what is installed, planning upgrade paths, and managing the network linking the software environments.

Keeping an inventory

There must be a well organised and up-to-date inventory of all the equipment and software under the control of the operations department. (This inventory is sometimes referred to as a 'repository', but should not be confused with the repository product that was recently announced by IBM.) In a single-vendor environment, where all the connections between terminals, peripherals, and processors have been supplied through one vendor, relatively few details need to be held in the inventory, because the operations manager can rely on the vendor to ensure that the components fit together.

In a multivendor environment, with units from different suppliers, and interconnections perhaps supplied by another set of vendors, the inventory needs to be maintained at a greater level of detail. It must contain accurate records of installed hardware and software releases, and of the connections, particularly where equipment is decentralised or networked. Without this level of detail, it will be difficult to pinpoint the likely causes of faults and restore a full operational service, and to exert satisfactory control over the level of service provided by the vendors.

Keeping the inventory up to date is a difficult and time-consuming task, as the experience of the Commission of the European Communities demonstrates. The Commission has a wide range of decentralised equipment from many vendors. Although Unix is used as the basis for the software environment, there is a significant problem in managing the decentralised equipment. The final section of the Appendix describes how the Commission is tackling these problems.

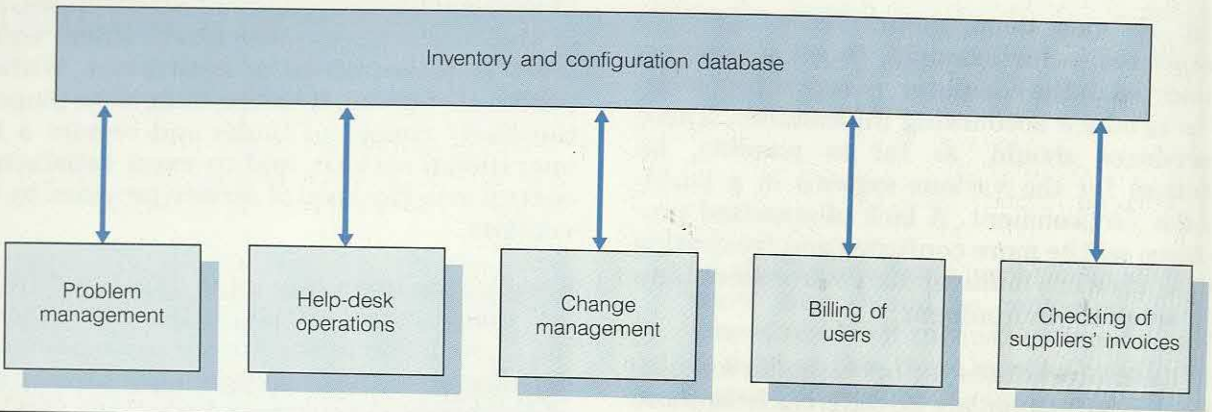
Computer-based inventories can be important aids to control, and are the basis of some network-management systems. In Report 65, *Network Management*, we describe an ideal network-management system incorporating an inventory and configuration database. A similar approach is desirable in multivendor environments, where the inventory and configuration database would form the basis for change management, billing of users, and checking of suppliers' invoices, and so on. This approach is illustrated in Figure 5.1, overleaf.

Planning upgrades

The timing of hardware upgrades is always a matter of fine judgement, requiring a balance between the availability of products and discounts, and the real requirement for additional capacity. The situation is complicated by the fact that, in a distributed multivendor environment, there is a choice to be made about whether to upgrade the central or decentralised machines. There may also be opportunities to choose between various software environments for developing a new system. In such cases, the possible upgrade path for each environment should be taken into account.

Software upgrades also require special attention in a multivendor environment. A Dutch Foundation member reported that a planned

Figure 5.1 A common inventory and configuration database provides the basis for many operations-management tasks



upgrade, to enable an important CAD application to use additional functions, could not be made because it required a new version of the software. This version could not be installed because it would have caused problems in a linked environment — the interface to another vendor's software would not have worked.

Planning software upgrades can also be complicated when the same software needs to be installed on different vendors' hardware. In such cases, software versions for the different hardware products may be released at different times and with slightly different facilities. Standard good practices for upgrading software, such as those listed in Figure 5.2, will usually be sufficient to avoid major problems.

Managing the network

As we explained in Report 65, *Network Management*, the management of the network presents special operational problems, and following good industry practice is essential to deal with them effectively. The management of multivendor networks requires the use of tools, each of which is oriented towards one particular network architecture. For most users with several network architectures, the use of several network-management systems is unavoidable.

In the absence of complete, integrated network-management toolkits, there are three possible ways of ensuring an adequate level of network management:

— Obtain all the network elements from a single supplier. While this will create a consistent and comprehensive system, it is not a feasible option for most organisations, because they already have an installed base of multivendor equipment. Furthermore, as we pointed out in Report 65, no one supplier can provide the full range of network equipment required to construct the complex type of communications network now commonly in use by Foundation members.

— Opt for a custom-built system. The facilities will be better, but the cost is considerably higher, to the point where it can be justified only by those few organisations where the network is an integral part of the business.

Figure 5.2 Procedures should be set in place for upgrading software in multivendor systems

— Before authorising a software upgrade, check that all the releases to be supplied and installed on the different equipment, in the various locations or countries, are compatible.

— Verify all claims about the compatibility of new software releases, and ensure that, in the event of problems, it is a simple matter to fall back on the previous release.

— Initiate software upgrades on a pilot system, if possible, so that difficulties can be detected before they cause disruption.

— Use standard test sets to test the links between the systems that have been affected by the upgrade before giving it production status.

— Keep a record of all the own coding introduced and of the extensions made to standard software — for example, the contents of parameter files.

— Maintain records of all the program bugs found, or implementation difficulties experienced.

— Follow these procedures for all kinds of software upgrades.

- Mix and match products from a variety of suppliers to achieve the facilities required, at an acceptable cost. In this option, a variety of incompatible proprietary tools will be used for network management, possibly enhanced by small, customised facilities. Increasingly, it will become possible to purchase tools that appear to provide integration, through filtering and correlating alarms and other messages from a number of proprietary tools to provide a simpler user interface.

Set up appropriate contingency arrangements

There is no doubt that a multivendor environment makes the provision of back-up facilities more difficult. The responses to our questionnaire indicate that there is no easy solution, since it is unlikely that a comprehensive back-up facility will be available from a disaster-recovery service supplier for a particular organisation's mix of hardware and systems software. At one extreme, a large public corporation had made no provision whatsoever for back-up facilities, and admitted that it had not even examined the options for contingency arrangements. At the other extreme, a company in the financial services sector had duplicated the computers from its different vendors, at substantial expense, because the different computers were unable to provide back-up for each other.

For most organisations, neither of these approaches is entirely satisfactory. In a multivendor environment, the requirements for back-up facilities in the event of equipment failure or unavailability need to be analysed in detail, perhaps with some specialist help. The aim is to identify which applications need to be backed up, what level of back-up is appropriate, for what period of time the loss of the application is acceptable, and what level of integration must be provided in the back-up arrangements. As always, back-up procedures will need to be documented carefully, and trial runs should be organised regularly to ensure that the contingency plans do work.

It is probable that providing emergency back-up facilities will be more expensive in a multivendor environment. Planning for the appropriate level of back-up can help to keep the cost down; one Foundation member even found the cost to be lower than it would have been in a less well planned, single-vendor environment.

Ensure that security is not jeopardised by the weakest link in the chain

Security will be more difficult to guarantee in a multivendor environment. Where users have access to more than one mainframe or minicomputer system, the security controls are more difficult to administer successfully. At Electricité de France/Gaz de France, for example, there is concern over the difficulty of defining standard procedures in a situation where many users have access to both Bull and IBM mainframes. The Bull security systems are different from the global systems available for IBM systems (such as RACF and ACF2), making it very difficult to control access to the multivendor environment with a single, standard, software interface. The company is concentrating on developing standard risk-assessment procedures as a first step, while encouraging each site to develop its own security arrangements.

In a large, highly devolved company such as Electricité de France, this may be the only practical solution, but in more centralised organisations, a central security officer or department may be required to cover all the installations in the multivendor environment. This unit will be charged with designing standard security procedures as far as is possible, and monitoring their installation and usage. Some of the actions it can take are:

- To ensure that passwords for each of the systems are changed regularly.
- To discourage the storing of passwords on intelligent workstations by providing stored log-in procedures that require the password to be input as a parameter.
- To encourage the technical department to provide front-end systems that will make it easy for users to log-in without the need for varied and complex procedures.
- To monitor failed attempts to enter the systems, and either take steps to improve the procedures for legitimate users or to follow up attempted security infringements.

The subject of systems security will be the topic of a Foundation Report to be published during 1990.

Manage the sources of supply

On occasions, vendors may disagree about each other's responsibilities or commitments in a multivendor environment. If disputes occur, the systems department must take the initiative in resolving them, but every effort should be made to prevent such problems arising in the first place. This will mean clearly defining the responsibilities of the vendors, and if necessary, considering alternative sources of supply.

Allocate responsibilities among vendors

In any systems activity, it is important to make sure that the responsibilities of the participants in a project are clearly defined — that everyone knows what he is expected to do, when he is expected to do it, and how his performance will be evaluated. In a multivendor environment, this is particularly important. The level of performance that the vendors and their products are expected to reach should be set out in formal contracts, and the various vendors should understand precisely how responsibilities are allocated between them. In this way, systems directors will be able to monitor the performance of vendors, and to provide them with regular feedback. They will then be in a position to judge which vendor is providing the best value for money, and to negotiate with those whose performance is not up to standard.

A surprisingly large number of organisations make no attempt to measure the performance of their suppliers; others go to considerable lengths. For example, the Sun Exploration and Production Co., a US oil exploration company, has developed a computerised trouble-reporting system that is used to track maintenance and support performance. Such sophisticated tracking systems are neither common, nor essential, for most organisations, but some tools and systematic procedures will need to be introduced to record performance data. The most important items to be recorded by such a performance-monitoring procedure are listed in Figure 5.3, although in particular circumstances, other items may need to be included, or given particular emphasis.

Clearly, managing relationships with vendors is not a spare-time task. It should be recognised as a separate role in the systems function, and be made the responsibility of one or more specific individuals. Usually, the systems director will take responsibility for much of this task.

Consider alternative sources of supply

Several Foundation members commented that their multivendor environment had 'inevitably' led them to consider third-party maintenance as an effective way of unifying the maintenance of equipment from different vendors, of feeling less 'in the hands' of hardware suppliers, and of introducing an element of competition, and so reducing the overall cost of maintenance.

Having a single source of maintenance services can be a very attractive option for systems directors:

- They will receive unified support, billing, fault reporting, contracting, and service-level agreements.
- They will avoid having to comply with the conditions imposed by various hardware vendors — for example, change-control or testing procedures.
- They will avoid disputes about who is to blame for particular faults in the system.

Hardware suppliers see the provision of third-party maintenance services as a growing business opportunity, and are now offering to maintain other vendors' equipment as well as their own. To date, however, few contracts of this type have been signed. Systems directors may be concerned about the motive of their main vendor if he offers to maintain other vendors' equipment, or about his capability to do the job well, or about his ability to obtain spare parts and provide satisfactory documentation.

It is more difficult to obtain a single source of maintenance for the telecommunications network, because of the multiplicity of suppliers usually involved in the configuration. Not all telecommunications vendors are willing to take on the role of lead supplier, and they may charge heavily for accepting responsibility for all the interfaces to and between other suppliers' equipment. This is a worthwhile option to consider only if there is a main

Figure 5.3 The performance of vendors and their products should be judged according to agreed measures

Product-related measures

Mean time between failures
May distinguish between different types of failure, but in such cases, the categories of failures in different environments must be comparable.

Planned and unplanned down-time
Usually expressed as a proportion of the scheduled time for which the product should be available. In order to be comparable in a multivendor environment, such measures should be based on the same basic availability schedules. If this is not possible, every effort should be made to ensure that the derived measures for different software environments are compatible.

System response times
It will usually be extremely difficult to define system response times in a way that makes it possible to compare them directly in different software environments, because of different workloads and different configurations. Nevertheless, objectives should be set and performance monitored for each environment.

Compliance with standards
Used to evaluate whether products conform to the standards set by the technical architecture. These could be open standards, or proprietary standards, or even standards defined by the user organisation itself, possibly in collaboration with suppliers. This is unlikely to be a quantitative measure; instead, it is likely to indicate whether the product conforms to various aspects of the standards underlying the technical architecture.

Vendor-related measures

Call-out time and time to fix
Basic measures of the quality of support provided by the vendor. These may be difficult to standardise, especially if different vendors are given different objectives — for example, a vendor of a computer system supporting realtime applications may have much more stringent criteria to meet than a vendor of a system that supports a batch workload only. Instead, it might be more useful to measure how often the vendor responds within the agreed time.

Performance against commitments
Monitors how the vendors honour their commitments, especially delivery dates. Rather than attempting to keep track of each and every commitment, it is usually advisable to focus on either a particular type of commitment — for example, delivery dates or implementation of standards — or to categorise aspects of the commitment and to measure only the most important. It will be very difficult to make these measures strictly comparable between different vendors, but it will show up any major differences in vendors' performance in this area.

Price/performance
In some ways, the most interesting of all measures, but probably also the most difficult to handle correctly. Many organisations use comparatively straightforward measures, such as the cost of a 'standard' configuration, or of a specific service.

Level of support provided
Measure of the basic scope of the support given, including hours of coverage, availability of online assistance, and definition and handling of major and minor failures. It also covers ancillary services, such as training and documentation.

communications supplier who already plays a major role in the success of the network. Alternative sources of supply can also be considered for services usually provided by an organisation's own staff. This will inevitably lead to considering facilities management, with all the same advantages and disadvantages that apply in a single-vendor environment. In a multivendor environment, facilities management may be an attractive option for one of the environments, where the systems function is in the process of converting from one environment to another, or where one software environment is no longer being actively developed.

Report conclusion

Because there are strong business pressures favouring multivendor environments, they will be inevitable for many organisations. They are, however, more complex, and therefore more difficult to manage than single-vendor or single-architecture environments, and they are costly to operate.

While multivendor environments can provide benefits, their effect on costs is often not clearly understood. Multi-architecture mainframe environments are nearly always more expensive than equivalent single-architecture environments. In a single-architecture, multivendor environment, the greatest savings can be made in the acquisition of hardware, because it is possible to negotiate the best deal from competing vendors. In a multi-architecture, multivendor environment, savings can be achieved in the acquisition of hardware and applications software, but they are likely to be more than outweighed by substantial additional costs in other areas, especially the continuing cost of the staff required to integrate applications in the different software environments.

The substantial continuing cost of staff can be controlled to some extent by employing as many people as possible who can work effectively in several software environments. This will give managers much greater flexibility in allocating staff to projects, and at the same time, will increase staff motivation and reduce staff turnover. Good recruitment, training, and job-rotation schemes all have a role to play in achieving the right mix of environment-independent and vendor-specific skills.

A suitable technical architecture, covering hardware, software infrastructure, and applications, will play a significant part in reducing the costs of integration, but defining and implementing it will require a lot of time and effort on the part of senior systems staff and users. The aims should be to reduce the number of software environments to the minimum consistent with business needs, and to make the multivendor environment transparent to users. Software environments, such as Unix, which allow different vendors' hardware to be included in the technical architecture, are useful in pursuit of the first of these aims. Standards, while representing only one part of the solution, can be very helpful in the second.

From an operational point of view, making the multivendor environment transparent to users has implications for the way the help desk is staffed and run, and for the operational procedures, particularly with respect to managing upgrades and enhancements to the overall configuration. To maintain a stable operational service in a multivendor environment, managers may sometimes need to call on the services of third-party maintenance companies, subcontractors, or organisations offering facilities-management services.

These are demanding and challenging tasks. Managing successfully in a multivendor environment will require systems directors to strike a balance between the business pressures to increase the number of vendors, and the management time and costs that will undoubtedly be incurred by doing so. They will need to be prepared to make this commitment of time and effort if the benefits of operating in a multivendor environment are to be realised without a heavy cost penalty.

Appendix

The Commission of the European Communities operates successfully in a multivendor environment

The Commission of the European Communities (CEC) is one of the institutions administering the European Economic Community. It employs about 12,000 civil servants, and is based in Brussels and Luxembourg. The systems function has over 500 permanent staff, most of whom are based in Luxembourg. Most users are 200 kilometres away in Brussels.

The CEC has consistently pursued a multivendor policy, which it sees as necessary to retain independence from individual vendors and to have access to the most appropriate technology for its users. The central computer facility in Luxembourg is based on mainframes from four manufacturers — Amdahl, Bull, ICL, and Siemens — all running proprietary operating systems, although a strategy for migration to open systems has been formulated for mainframes. Departmental computing in Luxembourg and Brussels is based on Unix machines (from a number of suppliers including Bull, NCR, Nixdorf, and Olivetti), and some personal computing on MS-DOS-compatible systems. There are over 40 local information systems units in Brussels and Luxembourg, each with one or more departmental Unix machines, with terminals and personal computers attached.

Tackling the problem of technical integration

The CEC distinguishes between hardware integration and software integration. The major problems of linking up different vendors' hardware are essentially solved. Integration of different software environments is more difficult, and is more of an issue for the departmental systems and personal computers than for the central mainframes, which run largely independent applications. Integration is predominantly user-driven, because users need to be able to access many different services from their workstations. The pressure originating from users is illustrated by the growth in the number of workstations — from 450 in 1980, to an expected 6,000 in 1990, when half of the CEC's staff will have a workstation on their desks.

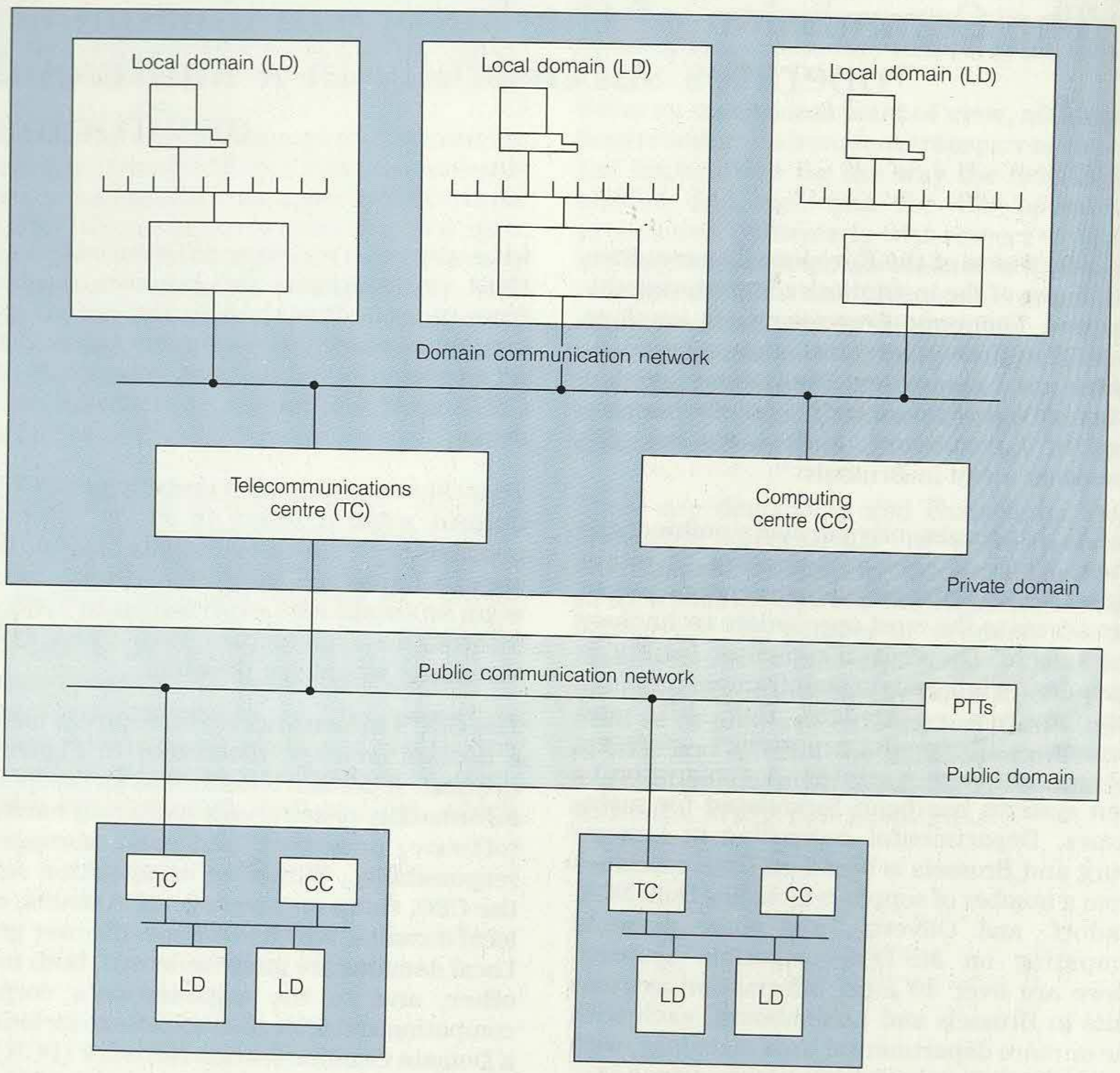
To tackle the integration problems, the CEC has defined what it refers to as an 'informatics architecture', which essentially sets out how it sees its hardware, communications, and software environments evolving up to 1995. This informatics architecture uses open systems standards whenever possible.

The CEC's informatics architecture is based on a domain concept, illustrated in Figure A.1, overleaf. A domain is defined as an independent information systems unit including hardware, software, personnel, and some management responsibility. Within an organisation such as the CEC, there may be several domains, called local domains, which serve specific user groups. Local domains are interconnected, both to each other, and to the organisation's corporate computing and telecommunications facilities, by a Domain Communication Network (DCN). The ensemble of local domains, the DCN, and corporate facilities constitute the organisation's private domain, which can be linked to public facilities (the public domain) provided by, for example, the PTT, and from there to the private domains of other organisations.

The informatics architecture specifies the hardware and software underlying this structure. Some of the central elements are listed below:

— The operating systems for the central computing facilities are VME (ICL), BS2000 (Siemens), VM/CMS and MVS (Amdahl/IBM), and GCOS8 (Bull). Unix will eventually replace the proprietary operating systems.

Figure A.1 The domain structure of the CEC's informatics architecture



(Source: Commission of the European Communities)

- Local-domain computing is based on Unix for departmental computers, and initially, on MS-DOS (later Unix) for personal computers.
- The central mainframes will be linked by a high-speed network.
- A packet switched data network provides the DCN.
- Ethernet LANs interconnect Unix machines within one local domain, and later, will connect workstations to Unix machines.
- Protocol converters, initially standalone, and later incorporated in the local-domain Unix computers, will provide access to the different mainframe systems in the central computing facility for interactive working.

The hardware supporting this environment could come from any of the selected suppliers.

With these elements, the CEC is now seeking to assemble user services with commercially available software components. It is looking particularly for software which:

- Is easily integrated with other software components in the Unix environment.
- Is transportable or available in as many different software environments as possible.
- Provides a consistent user interface.
- Is designed for distributed processing.

The main concern is for the departmental local-domain computers, where software components need to be able to share information and offer a unified user interface. So far, Q-Office and Oracle are the main software elements selected for local-domain computers, but eventually, the CEC aims to select a set of products capable of interworking, without being tied to a particular vendor. This is difficult since there are, as yet, few standards in prospect which might be of help here. The CEC expects client-server architectures to become increasingly useful.

Creating a consistent user interface

As the pressure for integration comes largely from the users needing to access applications running in different software environments, networking and protocol conversion facilities are important elements of the informatics architecture. Simply having access to different machines is not enough; a consistent user interface is required for accessing different services. In order to achieve this, the CEC is developing a User Agent Service (UAS), an application which connects the user with any service that he is authorised to access. The UAS will:

- Give guidance on identifying the appropriate service and formulating access requests.
- Support a first-level access control to requested services.
- Establish the connection to the requested service.
- Collect accounting data on the service used.

The UAS is based on a client-server concept. The relationships between its main constituent parts, and their location in the domain structure, are shown in Figure A.2, overleaf. A consistent graphical user interface will be provided when appropriate workstations are widely available.

Developing the informatics architecture

In its informatics architecture, the CEC has defined an evolutionary path for its IT systems, identifying four major stages. This makes it much easier to select products that can be integrated into the existing infrastructure without too great a cost. The evolutionary architecture also shows up areas where specialised integration products are required. For example, Steps 2 and 3 call for multiprotocol convertors, which allow users to access old systems using different proprietary terminal protocols on different mainframe computers from the same terminal.

The informatics architecture was developed in close liaison with suppliers, and many ideas were discussed with vendors to get their feedback. Periodically, the CEC organises meetings with all of its suppliers in order to present different elements of its informatics architecture to them. In particular, the CEC needs to understand suppliers' attitudes to standards, of which it must take account when updating the informatics architecture.

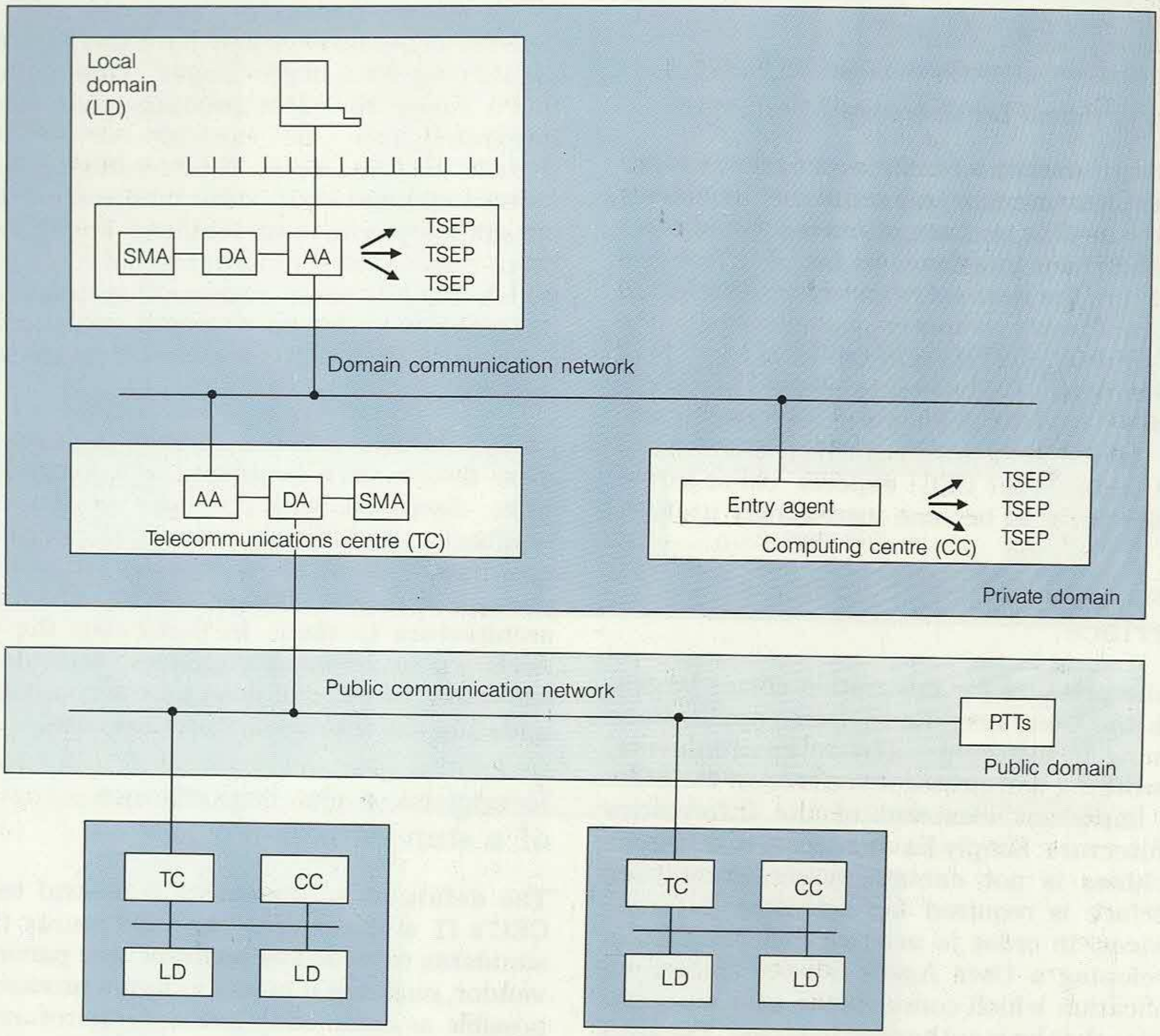
Recognising the importance of a standards policy

The definition of standards is central to the CEC's IT architecture. The CEC wants these standards to be independent of any particular vendor, enabling it to buy as many products as possible as commodity items. It therefore has a strong commitment to open systems standards.

Waiting for standards to become established before incorporating them in the informatics architecture is too slow, and would restrict users' access to new technologies. Therefore, the CEC has decided to intercept emerging standards. Thus, Unix was adopted as a central plank of the CEC's IT strategy in 1984, well before most industry experts considered it to be an industry standard. The CEC is aware that this approach increases the risks, and it attempts to limit these risks in two ways:

- By influencing the process of defining standards. The CEC participates in standards-setting bodies, discusses standards requirements with its vendors, and includes compliance with standards as a mandatory

Figure A.2 The structure of the CEC's User Agent Service



Key
AA Access agent
DA Directory agent
SMA Systems management agent
TSEP Target service entry point

(Source: Commission of the European Communities)

requirement in its procurement. The CEC believes that many user organisations could do much more in this respect and that they could exert a much greater influence over suppliers in support of standards. (The CEC also directly influences standards by issuing directives and regulations, such as Council Decision 87/95/EEC, making the compliance with open standards a mandatory criterion in public procurement wherever possible.)

— By keeping a close watch on the technology, the vendors, and the economics of different technologies, in order to be in a better position to predict which standards are likely to become widely supported. The CEC has appointed 'product managers', whose task is to follow the evolution of the product types of interest to the CEC (workstations, departmental computers, LANs, database management systems, and so on). Product managers also play a major role in procuring new products and defining the informatics architecture of the CEC. The CEC believes that for a product manager to play his part successfully, he needs to be technically competent, to understand the users' needs and to be able to command the respect of his colleagues — a rather rare mix of skills.

Following this policy obviously requires additional effort; the CEC estimates that it could amount to the equivalent of two people full-time, in addition to the broadly spread effort required to define and maintain the informatics architecture. The CEC believes, however, that these costs are outweighed by being able to bring new facilities to users earlier, and by the reduced cost of integration.

Keeping the inventory up to date

The day-to-day operational management of the CEC's decentralised multivendor environment is the responsibility of a separate unit within the systems directorate. Its mission is to supply and support the decentralised hardware and software used by the 40 or more local information systems units, which are of different sizes and at different stages of IT development.

As there are so many different versions of hardware and software installed with the users, a good and effective release-control and configuration-change-management system is far more important in the CEC's multivendor environment than it would be in a single-vendor environment. Support and maintenance can be effective only if detailed and reliable records exist of the hardware and software installed in each of the local systems units. The CEC has a policy of rotating equipment between user departments as their requirements evolve; this policy is also dependent on the existence of an up-to-date inventory.

It is not, however, easy to keep the equipment inventory up to date. Much of the equipment installed in the users' departments is easily moved by the users themselves. Furthermore, the procedures that have been designed to deal with office moves are lengthy, and when a move needs to be made fast, it may be carried out before the paperwork is completed. Site visits are therefore essential to ensure that the inventory is accurately maintained.

Although the CEC has developed a computer application to control the inventory, maintaining it still requires a considerable effort. Around 20 people keep track of the locations of all items of equipment. There is no doubt in the CEC's mind that the task is considerably complicated by the numbers of vendors involved, and by the slight differences between the products of different vendors. The CEC therefore intends to select ranges of equipment from a reduced number of suppliers, and each supplier will be required to guarantee that the different products within his range are fully compatible.

BUTLER COX FOUNDATION

Butler Cox

Butler Cox is an independent management consultancy and research organisation, specialising in the application of information technology within commerce, government, and industry. The company offers a wide range of services both to suppliers and users of this technology. The Butler Cox Foundation is a service operated by Butler Cox on behalf of subscribing members.

Objectives of the Foundation

The Butler Cox Foundation sets out to study on behalf of subscribing members the opportunities and possible threats arising from developments in the field of information systems.

The Foundation not only provides access to an extensive and coherent programme of continuous research, it also provides an opportunity for widespread exchange of experience and views between its members.

Membership of the Foundation

The majority of organisations participating in the Butler Cox Foundation are large organisations seeking to exploit to the full the most recent developments in information systems technology. An important minority of the membership is formed by suppliers of the technology. The membership is international, with participants from Australia, Belgium, France, Germany, Italy, the Netherlands, Sweden, Switzerland, the United Kingdom, and elsewhere.

The Foundation research programme

The research programme is planned jointly by Butler Cox and by the member organisations. Half of the research topics are selected by Butler Cox and half by preferences expressed by the membership. Each year a shortlist of topics is circulated for consideration by the members. Member organisations rank the topics according to their own requirements and as a result of this process, members' preferences are determined.

Before each research project starts there is a further opportunity for members to influence the direction of the research. A detailed description of the project defining its scope and the issues to be addressed is sent to all members for comment.

The report series

The Foundation publishes six reports each year. The reports are intended to be read primarily by senior and middle managers who are concerned with the planning of information systems. They are, however, written in a style that makes them suitable to be read both by line managers and functional managers. The reports concentrate on defining key management issues and on offering advice and guidance on how and when to address those issues.

Selected reports

8 Project Management
20 The Interface Between People and Equipment
25 System Development Methods
31 A Director's Guide to Information Technology
32 Data Management
34 Strategic Systems Planning
36 Cost-effective Systems Development and Maintenance
37 Expert Systems
40 Presenting Information to Managers
41 Managing the Human Aspects of Change
42 Value Added Network Services
43 Managing the Microcomputer in Business
45 Building Quality Systems
46 Network Architectures for Interconnecting Systems
47 The Effective Use of System Building Tools
48 Measuring the Performance of the Information Systems Function
49 Developing and Implementing a Systems Strategy
50 Unlocking the Corporate Data Resource
51 Threats to Computer Systems
52 Organising the Systems Department
53 Using Information Technology to Improve Decision Making
54 Integrated Networks
55 Planning the Corporate Data Centre
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

Forthcoming reports

Emerging Technologies
The Future of System Building Tools
Assessing the Value from IT
Systems Security

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.

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 BV
Burg Hogguerstraat 791,
1064 EB Amsterdam, The Netherlands
☎ (020) 139955, Fax (020) 131157

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

United States of America
Butler Cox Inc.
150 East 58th Street, New York, NY 10155, USA
☎ (212) 891 8188

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 1091

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