Management Summary

BUTLER COX FOUNDATION

Managing the Evolution of Corporate Databases

BUTLER COX FOUNDATION

Managing the Evolution of Corporate Databases

Management Summary
Report 64, June 1988

Butler Cox & Partners Limited

LONDON
AMSTERDAM MUNICH NEW YORK PARIS

Published by Butler Cox & Partners Limited
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.

Managing the Evolution of Corporate Databases

Management Summary

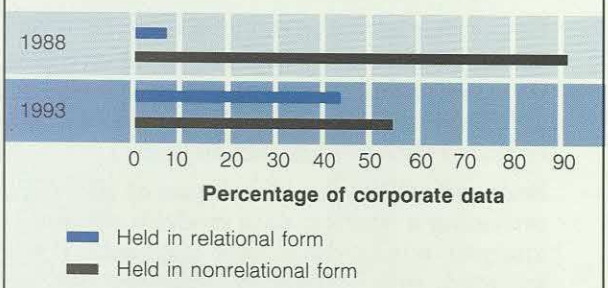
This document summarises the main management messages from Foundation Report 64, published in June 1988. The full report is available to members of the Butler Cox Foundation.

A new approach to corporate databases is needed

Most organisations now have some form of database systems based on 'first-generation' software products such as IBM's IMS and Cullinet's IDMS. They have also appointed a data administrator responsible for managing the data resource stored in corporate databases. The difficulty is that separate databases and applications have been developed independently of each other. As business requirements have changed, it has become increasingly important to integrate the separate databases. Moreover, some new business requirements need to use old and new databases and to have data organised in new ways. At the same time, there is growing demand from business users to access databases directly, so they can extract the data they require and manipulate it on their own desktop computers. First-generation database techniques and tools cannot support the required changes and are not flexible enough to cope with a wide range of ad hoc requests for data.

As a consequence, many organisations are now planning to introduce 'second-generation' database systems (in particular, relational databases and advanced data dictionaries) in order to alleviate these difficulties. Figure 1 shows the extent to which Foundation members intend to move to relational systems over the next five years. The promise of such systems is that they will make it easier to change databases and applications to keep them in line with changed business requirements, and will also make it easier for business users to access databases directly. However, it is not straightforward to make such a move because it is not possible to convert all existing databases and applications in one go. There will be an interim period where both old and new databases will need to be maintained. The problem is analogous to upgrading a highway from two lanes to four, but needing to keep the traffic flowing whilst the reconstruction work takes place.

Figure 1 **Foundation members plan a major move to relational databases during the next five years**



(Source: Survey of Foundation members)

Introducing new database systems is not sufficient by itself, however. Because the new systems will make it easier to change databases and the applications that use them, it is important to ensure that the database structures and contents are consistent and conform to a master plan, which we call a data architecture. There are thus four essential tasks in managing the evolution of corporate databases:

- To create a corporate data architecture.
- To choose the right mix of software products for implementing the corporate databases.
- To identify the requirements for user access to databases, and the tools that will allow the requirements to be met.
- To plan a cost-effective migration path for moving from existing databases and applications.

To ensure that these tasks are managed effectively, it will also be necessary to extend the role of the data-administration function.

For many organisations, carrying out these tasks will, in effect, mean adopting a new approach to corporate databases and to data management. In particular, the need to create a corporate data architecture will make data management visible at senior-management levels.

Create a corporate data architecture

A data architecture is a plan that identifies the key data items within an organisation and sets out the logical relationships between them. It provides a logical framework for the development, integration, and evolution of database applications. The most significant benefits of a data architecture are long term. Especially important are the abilities to integrate systems across different business functions and to build more-flexible applications. However, a data architecture can provide immediate benefits in two areas:

- Easier user access to relevant data. By establishing an inventory of data, based on a data dictionary, one data administrator had been able to reduce the time taken to locate the required data from weeks to days.
- Reduced application-development effort. By providing a 'starter' data model for business analysts whenever a new application was initiated, one Foundation member had removed much of the need for requirements analysis and had thus reduced the time required to design new applications by as much as 50 per cent.

A complete data architecture has three levels of detail and abstraction, as depicted in Figure 2. The highest level is an enterprise model (or a data model) describing the small number of major business activities, functions, and resources (customers, products, orders, invoices, factories, depots, and so on) and the relationships between them. The second level is a set of subject-area data models, which are derived from the enterprise model and contain the sets of data that are associated with particular business functions. The third most detailed level comprises the logical database designs. These designs contain detailed descriptions of all the data items used by business applications. Thereafter, the architecture is realised by implementing the logical database designs in a particular hardware and software environment.

Figure 3 (on page 4) describes how one organisation has used the top-down approach to develop a data architecture. Such a data architecture can be developed only in one way — by a rigorous top-down analysis of the organisation and its business activities. Creating a data architecture in this way is expensive and time-consuming, however, which means there are several obstacles to adopting this approach:

- The main benefits will not be achieved until the architecture is largely complete. Senior user management needs to be convinced that the benefits will outweigh the investment of time and effort required to develop it.
- High levels of skill and experience are required to develop the data architecture.
- Because of the time required to develop a top-down data architecture, the project can be undermined by changes in business priorities and requirements.

Because of these obstacles, many organisations have not used the top-down approach, but have used alternative approaches that are quicker and less costly. Although the end result is less satisfactory than a complete data architecture, they do produce useful benefits. There are four main alternatives:

- To base the architecture on a rapid, high-level business analysis and move directly to define the specific data models that are critical to immediate requirements. The danger of this approach is that any logical inconsistencies in the architecture will only be found later.
- To focus the architecture development on particular business objectives, and on the applications and databases needed to support those objectives. This approach is rapid and specific, but again it contains the risk that logical inconsistencies will appear as the architecture is extended to include other business activities.
- To adopt a bottom-up approach, gradually integrating separately designed databases that have been developed to support specific applications. Conflicts and inconsistencies will have to be resolved as they occur.
- To adopt a proprietary 'packaged' data architecture or to adapt one from a similar organisation or industry. The package should be well-designed, flexible, and fully documented. This approach is rapid and less risky, and the organisation benefits from the skills and experience of the package designer. However, at present few such packages are available.

Each of these alternatives is described in more detail in the main report. Figure 4 (also on page 4) shows the circumstances in which each is appropriate.

Choose the right mix of software products

The technical basis for corporate databases comprises software products, both for creating the data architecture and for supporting data management, and for maintaining the databases themselves. The full report describes the three main types of software products — data dictionaries, relational

database management systems, and distributed databases — and provides advice about when to use them and what to look for when they are being selected. These products should not be chosen in isolation, however. They should form an integrated set of products that can work together.

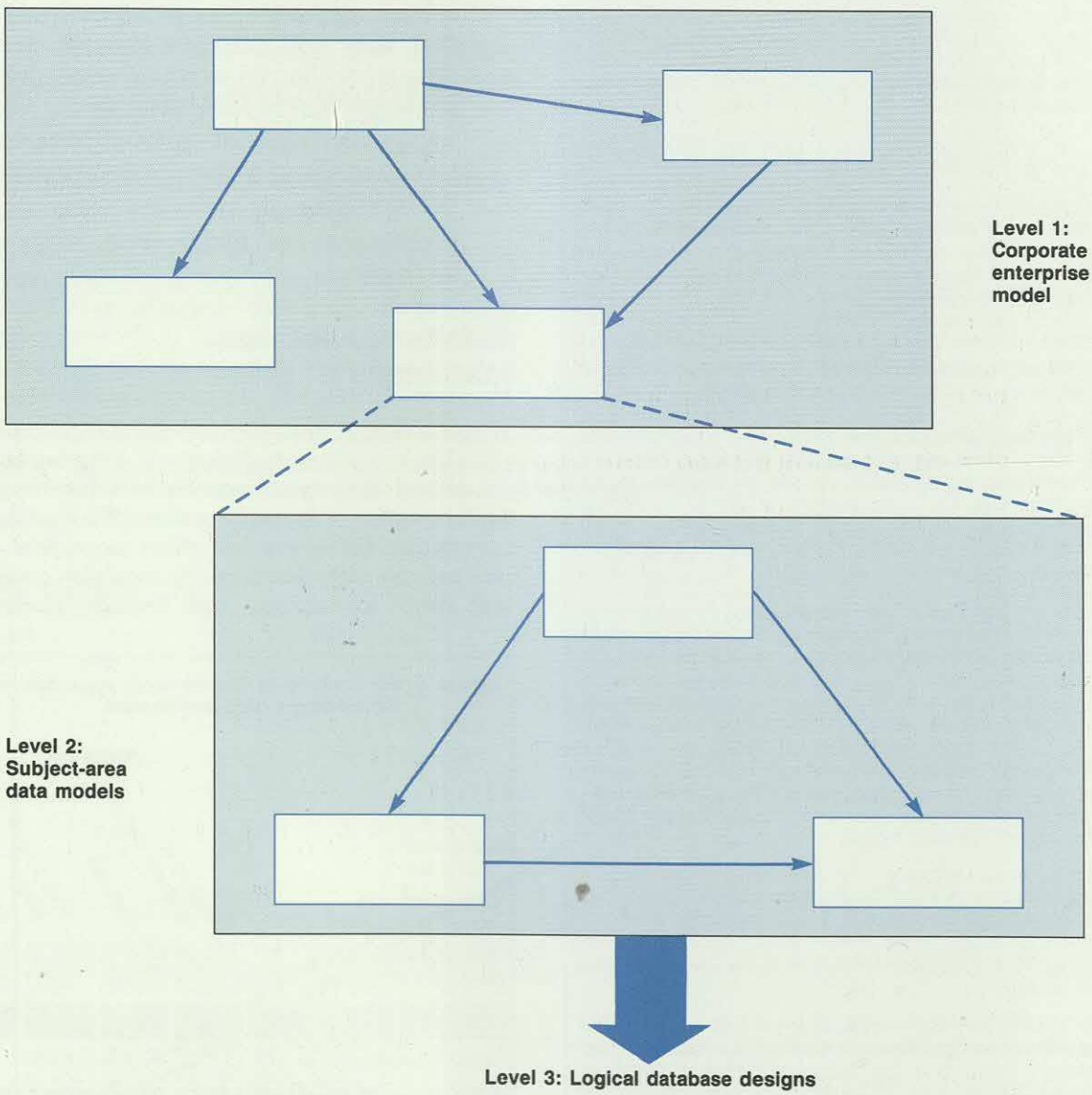
Data dictionaries

We believe that the data dictionary will become one of the most important components of data management and one of the most valuable tools to assist the evolution of corporate databases.

A data dictionary is a database containing information about an organisation's business data, its applications, its application-development projects, and its IT facilities. It is a valuable tool for various aspects of data management. For example, a data dictionary can be used to:

- Support the development and maintenance of the data architecture.
- Ensure that data items have consistent names and meanings.
- Describe the meaning of data items in terms that users will understand, and to help users to find the data they need.

Figure 2 Three-level data architecture



The boxes at level 1 represent major business entities in the corporate enterprise model, and the boxes at level 2 represent the data model of a specific subject area

Figure 3 B&Q used a top-down approach to create a corporate data architecture

B&Q is the largest do-it-yourself retail group in Europe. It has 220 stores, employs 10,000 staff and has a turnover in excess of £500 million. The group is expanding rapidly — opening a new store every two weeks.

There are 130 staff in the systems department, organised in four sections:

- The business-systems section, responsible for the information systems strategy and the applications portfolio. The business-systems account managers have full responsibility for ensuring that the systems needs of the company's business divisions are met.
- The development section, which builds the applications specified by the account managers.
- The technical-strategy section, which defines common requirements such as the network architecture, the hardware and software strategy, and the data architecture. The data administrator, who is responsible for the data architecture, is a member of this section, as is the database administrator, who is responsible for the physical implementation of the databases.
- The computer-services section, which runs the computers and communications network.

B&Q has developed a three-level corporate data architecture consisting of a corporate data model (the enterprise model), subject-area data models, and project data models. The company describes the corporate data model as the high-level model of the whole organisation's use of data. This high-level model includes about 50 of the major business entities such as products and suppliers — less important entities are excluded from the model at this level. An entity is included in the corporate model only if it is used in more than one subject area, and no attributes of the entities are included. (Thus, the corporate model does not specify that a customer record will always contain the customer's name and address, for example.) B&Q uses the corporate data model to provide an overview of the company's data for senior executives and to control the lower levels of the data architecture. The corporate data model also forms the basis of B&Q's plans for a management information database.

The subject-area data models cover specific subject areas of the company and typically map onto the organisation structure. Examples of B&Q's subject areas are personnel, property, and buying and merchandising. The starting point for developing a subject-area data model is a segment of the corporate data model. This is then enhanced and refined by further analysis of the subject area. At all times, the subject-area and the corporate data models are kept consistent. Changes resulting from refinements at the subject level are reflected in the high-level model, and changes in the high-level model lead to changes in the subject-area data models.

The third and lowest level of the data architecture is the project data models. A project data model comprises all or part of a subject-area model, and forms a working data model used by the development section. Project data models are checked by the data-administration function for quality and consistency.

B&Q believes that its multilevel corporate data architecture is essential to ensure that database applications are aligned to the business strategy and that the development of corporate databases can be controlled without the data administrator becoming a bottleneck in the development process. The company also believes that it is essential to use software tools such as a dictionary and diagramming aids. Without such tools, it is very difficult to control changes in the architecture and to communicate the results to all parties.

- Document existing applications and hence play a crucial role in providing the bridge between old and new versions of applications.

Many existing data dictionaries have serious limitations, however, implying the need for care in selecting an appropriate product. The limitations include:

- Poor user interfaces, with complex command structures and confusing screen layouts.
- Inability to hold the many types of data that may be needed by the various users of data. For example, the most basic form of dictionary would be of little use at the application-testing stage or for controlling program changes.
- Ability to interface with only a limited range of other data-management tools. For example, IBM does not currently provide a single dictionary that interfaces directly with its DL/1, VSAM, and DB2 products.
- Ability to hold data at only the database-design level, and not at the data-model or enterprise-model levels.
- Inability to exchange data with other dictionaries.

Relational database management systems

We believe that first-generation data management tools have now reached the limit of their development and that relational tools will eventually replace them. A move to relational technology is inevitable, but there are risks associated with moving too early because the available products still have limitations, and because there are

Figure 4 Alternatives to the top-down approach to developing a data architecture

Circumstances when applicable	Approach: High-level business analysis	Focused	Packaged based	Bottom up
Considerable change expected or required			✓	
Commitment level is low				✓
Need to change is low		✓		
Cost is a major factor	✓	✓		✓
Business and data modelling skills in short supply			✓	

performance and cost penalties. Undue delay will create more first-generation database applications that will eventually have to be converted and will delay the advantages and benefits.

The main business advantage of relational databases is that users can be given access just to the data relevant to their needs, in the format that they expect and understand. This is the main benefit sought by Foundation members from relational systems. However, there are also substantial benefits for the systems department:

- The physical design of the databases can be decoupled from the logical design, allowing the physical design to be changed and optimised after the database applications have been implemented without compromising the application logic.
- The processes of designing and accessing databases can be simplified, and programmers do not need a detailed knowledge of the physical layout of the database.
- Data can be changed or added without affecting other parts of the database.

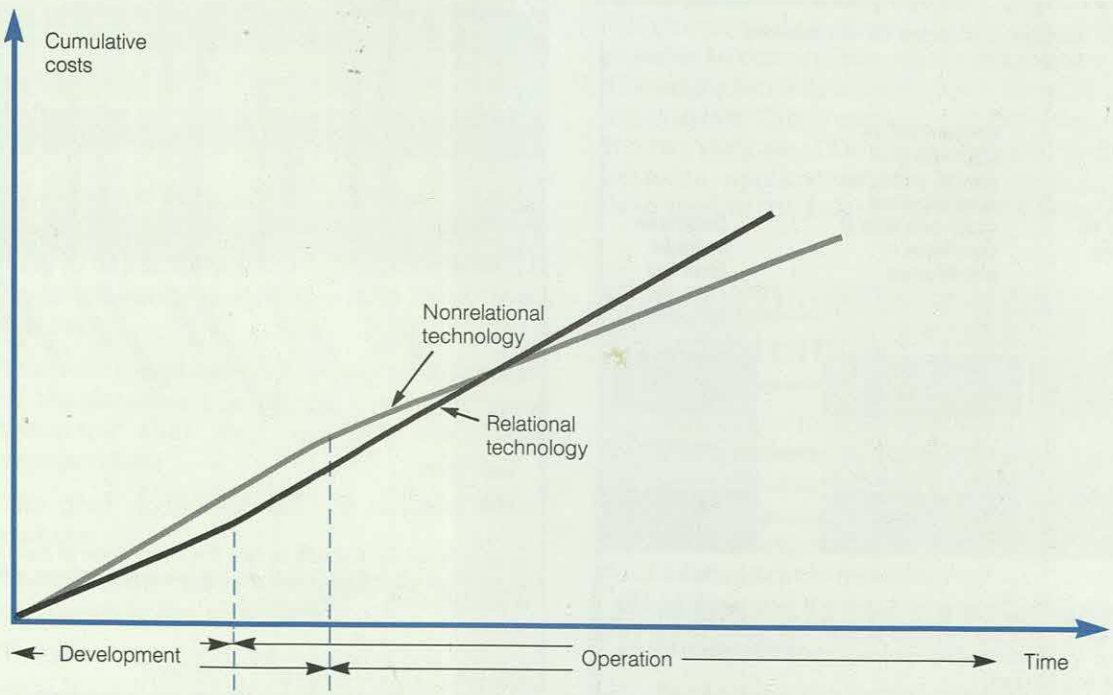
Relational products are still relatively immature, however, and not all the practical requirements have yet been satisfied. The major weaknesses are poor performance, the absence of automatic features that ensure the database structure and contents are not corrupted, and lack of a standard database-access language — although IBM's structured query language (SQL) is becoming a *de facto* standard. Relational databases still have slower retrieval times than well-designed first-generation databases. However, their performance is improving rapidly and is now adequate for most commercial applications, except for those with very high transaction rates.

The projected cost of relational technology for a typical application is depicted in Figure 5. This shows that application-development costs are lower, but that running costs are higher than for first-generation products. A relational database is therefore particularly suitable for applications with a short operational life, for those with complex database requirements, for those that change rapidly, and for those with low transaction rates. However, as the performance of relational databases continues to improve, they will become suitable for an even wider range of applications.

Distributed database systems

Distributed database systems allow the contents of a database to be stored on several different computer systems in a way that is transparent to users of the database. The database can either be partitioned, which means that different parts of the database are stored at each location, or replicated, which means that the same data is held at several

Figure 5 Relational technology allows applications to be implemented earlier and at lower cost, but the running costs are higher than nonrelational technology



locations. Figure 6 provides guidelines for deciding when to partition or replicate data, or to hold it centrally. Regardless of how the data is held, a distributed database should appear to users as if the whole database is local to them.

Although great progress has been made in distributed-database technology, current products still have limitations, particularly:

- The requirement for users to know the location of data in order to access it.
- The inability to update simultaneously several replicated parts of a distributed database.
- The inability to maintain the consistency of a distributed database after a failure.
- The inability to provide acceptable response times for queries that need to access several parts of a distributed database.

By the end of 1988, products that overcome these limitations will be available. Distributed database technology is thus reaching the stage when it can be used with confidence to support suitable applications.

Select an integrated set of products

The most powerful and effective data-management systems usually form part of an integrated set where each system can exchange information with, and support the functions of, the other systems. It is therefore important to choose products that conform to an overall architecture that will allow them to work well together and to exchange data easily. Certain suppliers are now developing their products to ensure that they present a common interface to both users and application programs. These benefits stem from the fact that all the data-management systems have access to the same set of data descriptions stored in the data dictionary, and that their interfaces are based on SQL.

Figure 6 Design guidelines for distributed databases

Ratio of updates to read-only access	Proportion of updates that would need to access more than one site if database is partitioned	Database storage strategy
High	High	→ Centralised
High	Low	→ Partitioned
Low	High	→ Replicated
Low	Low	→ Partitioned

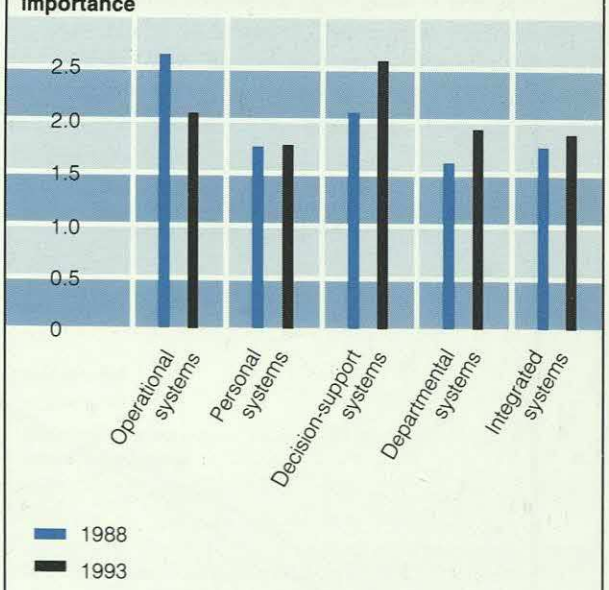
The most important criteria for deciding how to distribute data are the ratio of database updates to read-only accesses and the number of updates needing to access more than one site if the database is partioned.

Identify the requirements for user access to databases

In our survey of Foundation members for this report, we found that satisfying the requirements for improved decision support was expected to replace operational support as the highest priority for systems departments (see Figure 7). Many of these requirements will be met by providing business users with direct access to corporate databases. It is therefore necessary to identify the requirements for user access to databases and to provide appropriate products.

Three conditions must be satisfied in order to provide users with effective access to databases —

Figure 7 The focus of attention of the systems department is changing from operational to decision-support systems



Respondents were asked to rate the importance of each type of system both today (1988) and in five years time on a scale 0 (unimportant) to 4 (important).

(Source: Survey of Foundation members)

they must understand the meaning of the data required, the data must be easy to access, and the data provided must be relevant to their needs. These conditions may be met by providing users with access to a suitable data dictionary and by setting up separate databases that are specially designed to meet users' needs. We have already indicated that data dictionaries have an important role to play in data management. They can also be used to maintain an inventory of all the information used regularly by managers, to list the available data, to describe the meaning of each type of stored data, and to indicate where the data may be accessed. Some organisations are beginning to introduce artificial-intelligence techniques to assist users to find the data they need. Such techniques are used to help look for the right data, and to allow access requests to be expressed in natural language.

Separating the databases that provide management information from those that support operational applications prevents the two different sets of requirements from interfering. For example:

- It avoids performance degradation of operational applications.
- It allows external and historical data to be stored alongside current operational data.
- It enables the structures of the separate databases to be optimised to their use.
- It allows occasional users to update the data without the risk of contaminating operational databases.
- It allows management-information databases to be set up and subsequently removed according to managers' current requirements.

Plan a cost-effective migration path

The problems of migrating from a conventional database environment to a relational environment are likely to be much greater than those of converting file-based batch applications. The most usual problems are:

- The need to manage both old and new versions of the database throughout the conversion, ensuring that any updates are fully synchronised.
- The poor documentation of existing applications.
- The effort needed to translate and convert old databases to the new format.
- The short times available in which to change over to new online operational applications.

These problems mean that the migration will require substantial resources and may take a considerable time, so it is important to select an approach that minimises the cost and time and the consequent disruption. There are two factors that determine the migration path: the quality of the existing applications and the availability of suitable conversion tools. There are four possible migration paths:

- If the existing applications are of very poor quality or are badly structured, then often the only option is to redesign and rewrite them for the new environment.
- If the existing applications are of good quality, meeting users' requirements, with a robust and coherent applications architecture, then the databases and the applications logic can be converted using specially designed conversion tools. The technique of 'systems re-engineering' is especially useful in this situation.
- If the existing applications are well structured and the database contents are well defined, it may be possible to convert just the database to the new format, but then to use special software that allows the old applications to work with the new database.
- If the applications have been developed using fourth-generation languages or powerful Case (computer-aided systems engineering) tools, then recent software developments may make it possible to regenerate the applications from the original specifications but for the new environment.

Figure 8 overleaf summarises the situations in which each migration path should be used. Whichever path is chosen, it will generally not be possible to convert the whole portfolio at one go. The migration will have to proceed application by application. This creates the problem of managing the two versions of the database and of sharing data between applications. Various techniques are described in the full report for achieving this.

Extend the role of data administration

A powerful and independent data-administration function is required to manage the development of a corporate data architecture and to ensure that the ensuing data models and database designs conform with the architecture. The responsibilities of the data administrator include:

- Developing an accurate and consistent data architecture.
- Controlling the logical databases.

- Resolving data-ownership problems.
- Educating users throughout the organisation in the value of the data resource, and in the means to store and retrieve data.

In the future, we believe that as different types of information are increasingly stored in, and manipulated and disseminated by, computer systems, the data-administration function will evolve to encompass information administration. This implies that the role of the data administrator will be extended to include:

- Managing new forms of data, such as text and image.
- Controlling the organisation's use of external data and the transmission of data to the world at large (via EDI systems, for example).
- Taking on the custodianship of the organisation's knowledge bases.

Given this wide, expanding, and crucial set of responsibilities, we believe that large organisations should establish the data-administration function at a sufficiently senior level to ensure that it can operate on a corporate-wide basis. Without a strong data-administration function charged with the responsibility of creating the corporate data architecture and ensuring that the databases and systems conform with the architecture, the potential benefits of the new approach to data management will be put at risk.

Figure 8 Selecting a database migration path

Condition	Convert	Database(1) transparency	Re-create(2)	Redesign(3)
Sound data architecture	✓✓	✓✓	✓✓	
Applications meet user requirements	✓✓	✓✓	✓✓	
Good technical quality of applications		✓	✓✓	
Good quantity data descriptions		✓	✓✓	

Migration path

✓✓: Fully meets the condition

✓: Partially meets the condition

(1) Database transparency requires suitable software products to enable an existing application to access the new databases.

(2) Re-creation requires an applications generator that can re-create an existing application so it can be used with the new databases.

(3) If none of the conditions apply, the only choice is to redesign and rewrite the applications.

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 one of the services provided by Butler Cox. It provides the executives responsible for information systems in large organisations with a continuous analysis of major developments in the technology and its application.

The Foundation publishes six Research Reports each year together with a series of special Position Papers. The programme of activities includes a wide range of meetings that provide Foundation members with a regular opportunity to exchange experiences and views with their counterparts in other large organisations.

Butler Cox & Partners Limited
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 791c,
1064 EB Amsterdam
☎ (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
☎ (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
3rd Floor, 275 George Street, Sydney 2000, Australia
☎ (02) 236 6161, Fax (02) 236 6199

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

Italy
SISDO
20123 Milano, Via Caradosso 7, Italy
☎ (02) 498 4651, Telex 350309, Fax (02) 481 8842

The Nordic Region
Statskonsult AB
Stora Varvsgatan 1, 21120 Malmo, Sweden
☎ (040) 1030 40, Telex 12754 SINTABS

Spain
Associated Management Consultants Spain SA
Rosalía de Castro, 84-2°D, 28035 Madrid, Spain
☎ (91) 723 0995