5 USE OF INFORMATION TECHNOLOGY

5.1 Overview

If applied in an appropriate and sustainable manner, information technology can lead to considerable cost savings and efficiencies in an organisation. Alternatively, if technology is allowed to dictate strategy, costs are likely to rise and existing work patterns may be disrupted. Such situations demand a fundamental re-appraisal of the role of information technology. In essence, information technology should **support, not drive** data management objectives.

Although data can be managed without modern information technology, the latter has some important advantages over manual techniques. For instance, computers can be used to store large volumes of data and perform very rapid and complex analyses. They can also be used to validate data as they are entered and be used to produce multiple and varied reports from the same data. These advantages widen the range of purposes to which the data may be applied.

Information technology also brings certain disadvantages, particularly in the form of additional **complexity and costs**. Almost every item of new technology brings with it a maintenance, support and training overhead. Box 6 highlights several situations which, on balance, would benefit from the appropriate use of information technology.

5.2 Selecting Technology

In any given situation, the best type of information technology is that which is most appropriate to the tasks at hand — both now and in the future. In particular, the issues of scalability, connectivity, compatibility and sustainability need to be closely examined (see Box 7). Following this analysis, the advantages and disadvantages of different technological solutions should be tested under realistic local conditions before procurement takes place. Although useful, manufacturers descriptions, magazine reviews and specialist information services (e.g. Internet newsgroups and bulletin boards) should not be relied upon for strategic procurement decisions.

It should be noted that some characteristics of information technology are subjective, such as the ease of use of a software package or the quality of a scanned

Box 6 Situations where information technology could be beneficial

- Data contain relationships which are too complex, or are too great in volume for the capabilities of manual filing systems or word processors.
- It is necessary to integrate data from several sources into a combined output.
- There is a need for the data to be shared amongst more than one user in a single organisation, or with other organisations.
- Data require extensive searching, sorting or updating.
- Frequent and varied reporting of the data is required.

image. Thus, selecting technology purely from a list of features is unlikely to be satisfactory. Like before, a real-life test is the best way of determining whether technology will be suitable under the expected working conditions.

A wide range of options exist for managing data. These include single (stand-alone) computers running local copies of data-management software; locally-networked computers with shared software running on a file server (i.e. a Local Area Network or LAN); client-server architectures[4] which integrate the best characteristics of personal computers (friendly software and quick response) with the best traits of file servers (high storage capacity, fast data processing, good security); and fully-distributed databases consisting of a series of remote computers linked via permanent or dial-up communication lines (i.e. a Wide Area Network or WAN). The decision as to which option to select should be taken after a thorough examination of the factors summarised in Box 7. Clearly, the nature and extent of the data to be stored will influence this decision greatly, as will the degree to which the data need to be accessed electronically by internal and external users.

4 This option is becoming increasingly popular for medium- to large-sized organisations relying heavily on data management for their core business.

Box 7 Considerations when selecting information technology

- **Scalability**

  As the number of users, records or attributes grow, an application that once performed well on a low-cost computing architecture can deteriorate in performance quickly. Typically, stand-alone or small network computer architectures are most likely to suffer from this problem, which explains the rise of more sophisticated architectures, such as client-server.

- **Connectivity**

  To enable rapid exchange of data between individuals and organisations, electronic connectivity is desirable. This could take the form of a group of locally-networked computers sharing a common storage area, or more sophisticated dial-up communication lines to external services, such as the Internet and private networks. The capacity to connect computers together into more powerful resources is increasingly recognised as the key to rapid access and use of data.

- **Compatibility**

  The issue of compatibility is diminishing as manufacturers evolve a range of standard specifications for their IT products. However, the specifications — which are often proprietary in nature — are still too varied and numerous to discount the problem entirely. As far as computing platforms are concerned (i.e. computer hardware plus operating system), major decisions include whether to adopt IBM-PC compatible computers which running derivatives of the Microsoft Windows operating system, or larger workstations running the UNIX. Since the technologies are changing so rapidly, there is really no 'best' solution other than to adopt a platform which has proved to be reliable and useful in circumstances similar to those anticipated, working on the principle that, in such cases, compatibility issues are unlikely to cause serious disruption.

continued overleaf

Box 7 Considerations when selecting information technology (cont.)

- **Sustainability**

  For information technology to deliver long-term improvements in effectiveness, sufficient funds and expertise must be available for users to exploit its potential fully and not be disadvantaged by its costs in terms of training, technical support and maintenance. Technology which has proven effective under the prevailing conditions is usually the wisest choice.

One of the most common forms of software used to manage environmental data is the relational database management system (RDBMS). These offer flexibility and performance at modest cost, although they are not designed to manage large-scale textual sources (these are more effectively managed in a word-processing package). Other key software include geographic information systems (GIS), which store, integrate and analyse spatially-referenced data, and tools such as spreadsheets, statistical packages and special-purpose environmental modelling software.

5.3 Database Development

Database development involves designing and building the systems necessary to manage one or more related datasets. Generic methods have been proposed to develop databases, and the ideas presented in following paragraphs attempt to simplify and summarise these. For clarity, database development is partitioned into two phases: database design and applications development.

- **Database design**

  This involves identifying the structure and functionality of the database. The required sources of data are made clear, and the integration and processing techniques needed to achieve the desired outputs are identified. The design process gives rise to a **functional specification**, which is independent of both hardware and software, and does not assume any particular method of physical

data organisation (in practice, the technology available — which may be constrained by budgetary limitations — may affect the design of the database).

An important part of the design process is **data modelling**. This is the analysis of data objects and the identification of the relationships among these data objects. A common approach is to use **entity-relationship (E-R) diagrams**, as developed by Chen (1976). Quite simply, an E-R diagram depicts the contents of a database: an **entity** (shown as a rectangle) is an object (or 'thing') about which data are collected; and a **relationship** (shown as a line) shows the connections between the entities. The nature of the relationships between the entities indicates the number of occurrences of one entity that may be associated with a single occurrence of the other.

Three types of relationship are possible: one-to-one, one-to-many and many-to-many. For example, the entity 'Protected Area' may contain data such as the protected area names, legal status, size and so on. There may be a relationship to a 'Country' entity, indicating that each protected area is located within one or more countries (i.e. a one-to-many relationship). Although protected areas normally fall within the borders of a single country, being aware that it is possible for them to straddle more than one country has important implications for design. Indeed, failing to establish the correct relationship at an early design phase could restrict the development of the application at a later date. As discussed in Section 2 of Volume 2, it is vital to identify problems in the design phase, before large investments have been made in implementation.

In summary, the design process provides:

- ✓ a stable base from which to coordinate the development of the database, including the selection of appropriate equipment for implementation;
- ✓ a conceptual model which is free of implementation considerations, and which can be used as a point of reference when adding to or modifying the functionality of the database; and
- ✓ a baseline from which an optimum physical data organisation can be produced.

- **Applications development**

  This involves creating a fully-functioning database using the data management software selected for implementation (see Section 5.2). Entities in the database design become **tables** in the software, and attributes become table **fields**. The way in which relationships between the entities are dealt with depends on which software is used; if it does not support some types of data relationship, then this has to be resolved by altering the database design.

  Each field in the database is documented in terms of its purpose, data type, size and order in its corresponding table. When pooled across all the tables of the database, these definitions are known as the **data dictionary** of the database, and provide a description of its content, format and structure.

  After the database tables have been created, they are **populated** with data. If the data are already computerised, this may be achieved by directly importing them into the database, plus associated re-structuring and formatting. If the data are only available in hard copy form, they will need to be entered manually into the database via the keyboard or, in the case of maps, images and structured text, via other input devices, such as scanners, digitising tablets and related software.

  Most data management software packages enable developers to customise data entry procedures, for example by enforcing certain formats and validating or correcting data items as they are entered. This concept can be extended to other procedures, such as the querying and reporting of data, and saving data to removable media (e.g. a floppy disk) for back-up or delivery to users. The combination of data entry, querying and reporting features, security features and, of course, the underlying data tables, is known as a database **application**.

  Database applications need not be created perfectly at the first attempt. Indeed, there is an advantage in developing **prototype** applications over a short time frame, and at low cost, in order to provide users with a means of refining their needs from the database. Prototyping was discussed in Volume 3 in the context of information product development, but it is equally useful and, perhaps, more essential during the development of databases. The aim is to allow problems to be identified and corrected early on in the database development process, circumventing costly modifications at a later stage.