1

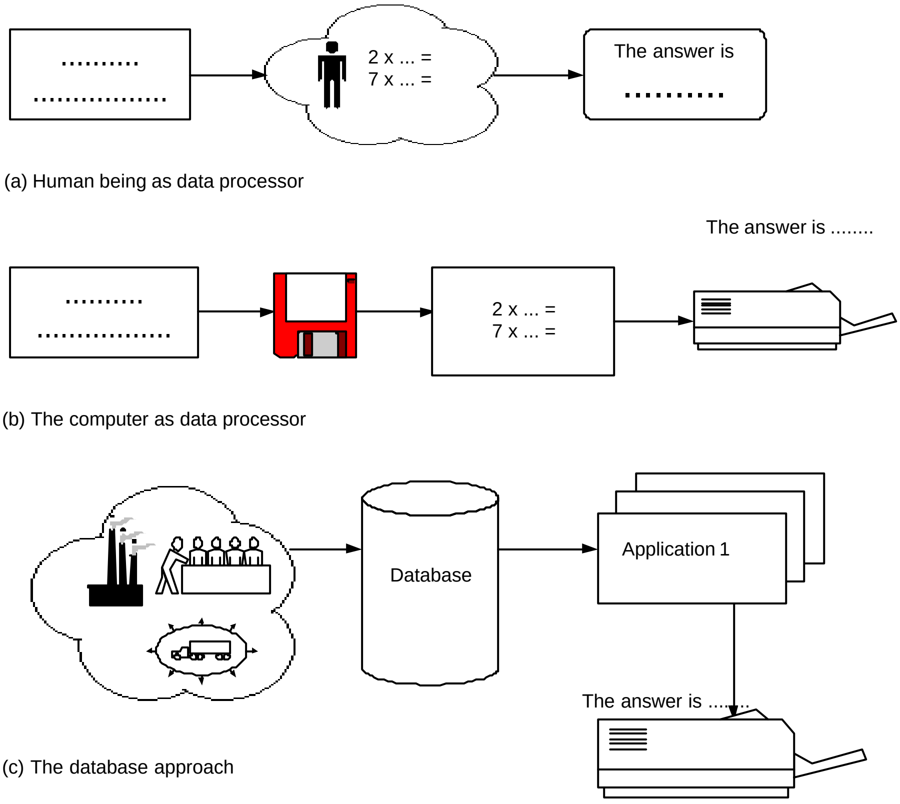
Database Approach

1.1 Introduction

The main objective for this chapter is to set the scene for the database approach. It shows the differences between the conventional file-based systems and database applications, discussing the advantages of the database approach. We also look at the difficulties of the database approach, and suggest that we need a strategy and methodology to develop database applications. Such an approach is outlined in this chapter and described in the rest of the book

With database systems, it is possible to hold the facts relating to parts of the organisation on an integrated set of computer files: a database. The 'organisation' in this context could be the whole business or, more likely, a part of it, such as a division or department. The various computer applications can use this as the data source. If the functions change, the data on the database will probably still be appropriate. If the facts change, then the database can be amended without redesigning the application systems. There is thus an element of **data independence** between the database and the applications that use it. The hardware and software can also be changed in order to reflect developments in the technology without requiring substantial changes in the application systems or the structure of the database.

Figure 1.1 illustrates three stages tracing the historical development of data processing. In Figure 1.1(a) the data is held on a document of some kind, the process carried out by the human being, and the result written down on paper. In Figure 1.1(b), the input source is the same but has been transferred to a medium that is suitable for computers to read. The process is performed by the computer, which puts the result on a laser printer, and printed out. Although a computer application, the system reflects the previous clerical system. There may well be more stages, but data processing is likely to be quicker overall, due to the speed of the computer. Figure 1.1(c) represents something much more radical. The facts of interest to the business are modelled on the computer database, and the various applications use the database as required. If the process needs to be changed in some way to reflect changes in the environment, the database will still be appropriate. The third option is likely to be the most rewarding. This should provide real added value when compared to either a manual system or a conventional computer system not using a database.



(a) Human being as data processor

(b) The computer as data processor

(c) The database approach

Fig. 1.1 Three ways of data processing

However, constructing the database is a complex task. The real world is so complex that to model the organisation, or even part of it such as a department, is no easy task. This text proposes a methodology to help construct the database. In the next two chapters, which complete Part I of the book, we show how we can best understand the organisation – the environment of our task – and the roles of the various personnel who will be involved in developing the database applications.

With background knowledge of the organisation established, it is possible to develop a formal model of the organisation. We discuss three modelling approaches - entity-relationship modelling, relational modelling and normalisation, and object modelling - in Part II of the book. All three modelling approaches are being used successfully; indeed, they are sometimes combined in the more flexible object-relational database management systems.

In Part III, we show how the model can be implemented on computer systems using database management systems (DBMS). We discuss implementation both on personal computers (PCs) and large computers. The well-used database language, Structured Query Language (SQL), is introduced, and we look in particular at two DBMS, Access and Oracle. Various issues are discussed, such as security and tool support in particular. We consider simple applications, implemented on PCs, as well as large corporate-wide distributed database applications.

In Part IV, we look at major database applications, including data warehousing, data mining, customer relationship management and Internet applications. Finally, Part V considers the issues relating to developing information systems that use a database as a source of information.

In the next section we look at the key factors for a database approach to business computer applications.

1.2 Motivations for the database approach

The database provides a major data resource for the organisation. It is an important asset of most organisations – just imagine if a company lost its customer or accounts data. The motivations for companies choosing a database approach for their data handling are many and include:

- Increase data shareability
- Increase data integrity
- Increase the speed in implementing applications
- Ease data access by programmers and users
- Increase data independence
- Reduce program maintenance
- Provide a management view of the organisation
- Improve the standards of the systems developers.

Increase data shareability: Large organisations, such as insurance companies, banks, local councils and manufacturing companies, have for some time been putting large amounts of data onto their computer systems. Frequently, the same data was being collected, validated, stored and accessed separately for a number of purposes. For example, there could be a file of customer details for sales order processing and another for sales ledger. This 'data redundancy' is costly and can be avoided by following a database approach.

In fact some data duplication is reasonable in a database environment, but it should be known, controlled and be there for a purpose, such as efficient response to some database queries. However, the underlying data should be collected only once, and verified only once, so that there is little chance of inconsistency. With conventional files, the data is often collected at different times and validated by different validation routines, and therefore the output produced by different systems could well be inconsistent. In such situations the data resource is not easily managed and this leads to a number of problems. With reduced redundancy, data can be managed and shared, but it is essential that good integrity and security features operate in such systems. In other words, there needs to be control of the data resource. Furthermore, each application should run 'unaware' of the existence of others using the database. Good shareability implies a ready availability of the data to all users. The computer system must therefore be powerful enough so that performance is good even when there are a large number of users concurrently accessing the database.

Increase data integrity: In a shared environment, it is crucial for the success of the database system to control the creation, deletion and update of data and to ensure its correctness and its 'up-to-dateness' - in general, ensure the quality of the data. Furthermore, with so many users accessing the database, there must be some control to prevent failed transactions leaving the database in an inconsistent state. However, this should be easier to effect in a database environment, because of the possibilities of central management of the data resource, than in an environment where each application sets up its own files. Standards need only be agreed and implemented once for all users.

Increase speed of implementing applications: Applications ought to be implemented in less time, since systems development staff can largely concentrate on the processes involved in the application itself rather than on the collection, validation, sorting and storage of data. Much of the data required for a new application may already be held on the database, put there for another purpose. Accessing the data will also be easier because the data manipulation features of the database management system will handle this.

Ease data access by programmers and users: Early database management systems used well-known programming languages such as Cobol to access the database. Cobol, for example, was extended to include new instructions, which were used when it was necessary to access data on the database. However we now have specific database query languages, such as SQL, and other software tools that ease the process of applications development in a database environment. Once the database had been set up, applications development time should be greatly reduced.

Increase data independence: There are many aspects to data independence. It is the ability to change the format of the data, the medium on which the data is held or the data structures, without having to change the programs that use the data. Conversely, it also means that it is possible to change the logic of the programs without having to change the file definitions, so that programmer productivity is increased. It also means that there can be different user views of the data even though it is stored only once. This separation of the issues concerning processes from the issues concerning data is a key reason for opting for the database solution. It provides far greater flexibility when compared to conventional file-based applications.

Reduce program maintenance: Stored data will need to be changed frequently as the real world, that it represents, changes. New data types need to be added, formats changed or new access methods introduced. The data independence of a database environment, discussed above, circumvents the necessity of changing each program with each change to the data structure or type.

Improve standards: Applications tend to be implemented by different project teams of systems analysts and programmers and it has been difficult to apply standards and conventions for all applications. Computer people are reputed to dislike adopting the norms of the firm, and it is difficult to impose standards where applications are developed piecemeal. With a central database, it is possible to impose standards for file creation, access and update, and to impose good controls, enabling unauthorised access to be restricted and providing adequate back-up and security features.

Provide a management view: This is a very important aspect in the context of this book. Managers have frequently complained that they do not get the benefits from the expensive computing resource that they have sanctioned. However, managers have become aware of the need for a corporate view of their organisation. Such a view requires data from a number of sections, departments, divisions and sometimes companies in a larger organisation. This corporate view cannot be gained if files are established on an application-by-application basis and not integrated as in a database. With decision-support systems using the database, it becomes possible for problems previously considered solvable only by intuition and judgement to be solved with an added ingredient, that of information, which is timely, accurate and presented at the required level of detail. Some of this information could be provided on a regular basis whilst some will be of a ‘one-off’ nature. Database systems should respond to both types of request.

1.3 Potential difficulties with the database approach

There are a number of problems with the database approach, but many have been largely overcome. The technology has improved out of all recognition, and this applies to both hardware and software. Costs have continued to go down, whilst both capacities and speed of access have improved. The technologist has learnt how to use the computer database environment to the good advantage of the organisation. User-friendly languages have been developed so that those users who are not computer experts may profitably access the database themselves. Managers are more involved with computing, partly as a result of their use of PCs, but also through their use of decision support systems, executive information systems and the like. A number of tools have been developed to support the user and developer and we are more aware of the importance of personnel in this success. All these aspects have played an important part in improving the likelihood of success for database projects.

However, a database project is large, expensive and long-term: there is a risk of failure ... and it will be a most expensive failure. We need a methodology that supports the development process. Such a methodology need not be too prescriptive. It needs to be more like a framework to guide us in the process of modelling the organization onto the database and developing our applications. This text provides such a framework, and one that will harness some of the most sophisticated aspects of the new technology. It looks at the techniques, the role of the innovator, user and manager, for, without considering all aspects of the process and the people involved, the database project will fail.

In the next section we outline our framework.

1.4 Database methodology

A major part of the process of developing our database is to derive a model of the area of discourse. This model aims to accurately represent the real world it is supposed to reflect. This book provides a modelling approach that guides this process. A **model** is a representation of the ‘real world’, often simplified, so that it represents only those aspects in which we are interested. Choosing and constructing appropriate models is difficult and the methodology described in this text uses a series of models and modelling techniques to derive the database.

This series of models begins with a very abstract model of the situation that is developed in the organisational analysis phase discussed in Chapter 2. It needs to be enhanced by understanding the roles of the personnel involved in the project. This includes users, managers and developers. These roles are discussed in Chapter 3. We can call this type of model ‘coarse-grained’, but it is a necessary first step to creating our more finely grained models later.

So the first phase of our approach is to understand that part of the organization that we are to model on the database. However, we cannot claim too much. It is impractical to represent the whole organisation on a database. The analyst aims to hold only that relatively small part which will be a good basis for most applications. More fundamentally, however, is that it can only be A and not THE model of the organisation. No model can reflect reality completely and accurately for all purposes. It may present a distorted picture for some purposes and even if the modelling process has 'gone according to plan', the resultant database cannot be perfect. We can never fully know reality; our perceptive process distorts our view of the real world.

In a database project, we are aiming to produce the most appropriate model of the organisation that we can, and transform this on to a computer database that can be used for the information systems applications. This is difficult. The coarse-grained modelling developed in the next two chapters is vital in this process because it is about choosing the most appropriate model for the organisation and people involved. It will be the basis for the finer-grained modelling approaches that follow.

Once we have created our coarse-grained model through the process of organisational analysis, we need to map this model into a form more suitable for database development. There are a number of potential models, but the most well used are entity-relationship, relational and object models. These are all described in Part II of the book. By mapping, we mean that the new model, though in a different form, accurately reflects the old model. In other words, everything on the old model that we want to represent in our database is contained in the new model, albeit in a different way.

Entity-relationship and relational models are the result of applying data analysis to our coarse-grained model. Object modelling is an alternative approach which models data and processes using one representation form. Each modelling approach has its supporters, but whichever is followed there are a series of well-defined steps to achieving the model. Indeed many aspects of this process are mechanistic and rule-based. This cannot be said of the organisational modelling which preceded this stage.

The model derived at this stage is not arranged for particular users or applications. It is a general model. But it is still not yet our database model. There is a further mapping process, described in Part III of the book, to our database management system. This is the software that manages the database. This mapping process can be even more mechanistic. Indeed, there are software tools that automate some of the task (described in Chapter 12). Many database management systems are based on the relational or object model (or indeed both).

Once the database has been modelled on to a database management system, and we look in particular at Access and Oracle in Part III, we can then develop applications that use the database. These applications are described in Part IV and the process of developing such applications described in Part V. But first we consider in the next two chapters how we form our coarse-grained model, which starts the process.

Summary

- Through the database approach, we have data independence, that is, a separation of the database and the applications that use it.
- The database approach represents a much more fundamental change than 'computerising' manual systems.
- The database approach provides many advantages, such as, increased data shareability, data integrity, speed of developing applications, easing the access of data, reducing program maintenance, improving standards as well as supporting management in its decision-making role.
- Developing a database, which accurately reflects the organization it is meant to represent, is complex and it is necessary to follow an appropriate methodological framework.
- The modelling process, in particular, is complex, long term, costly and risky. One approach is to implement the database through a series of modelling steps, from coarse-grained to fine-grained.

Exercises

1. Distinguish between data independence and data redundancy. How are they both keys to the database approach? When might some data redundancy be appropriate?
2. What is the difference between conventional file and database applications?
3. In what ways might a database be a model of an organization?

4. What difficulties do you envisage in developing a database for a university or other organisation of your choice?

Further reading

Beynon-Davies, P. (2000) *Database systems,* Macmillan, Basingstoke.

Connolly, T. and Begg, C. (2002) *Database systems: A practical approach to design, implementation and management,* Addison-Wesley, Harlow.

Date, C. J. (2000) *An introduction to database systems,* 6th edition, Addison-Wesley, Reading, Ma.

Elmasri, R. and Navathe, S. (2000) *Fundamentals of database systems,* Addison-Wesley, New York.

McFadden, F. R., Hoffer, J. A. and Prescott, M. B. (1999) *Modern database management,* 5th edition, Addison-Wesley, Reading, Ma.

These books provide alternatives to the present book. There are a number available and they tend to emphasise the technical aspects of database systems. We stress the management, organization and human aspects.