1

Introduction

1.1 Information and information systems

What is information? Many people have attempted to give a definition but most of them are not complete. A typical explanation is that information is processed data that has meanings to its users. But then questions arise in what meaning is. If *information* is to the study of information as object is to physics, and there are many laws by which we can study objects, then what are the laws by which we can study *information*? What is the study of information anyway?

What can be said here is that *information* is not a simple, primitive notion. Devlin (1991) compares the difficulties for a man in the Iron Age to answer the question 'What is iron?' and for a man in today's Information Age the question 'What is information?' To point to various artefacts of iron in order to answer his question would not be satisfactory; to demonstrate some properties of information as an answer to 'What is information?' is not good enough either. People can feel the possession of information, and can create and can use information. They gather it, store it, process it, transmit it, use it, sell it and buy it. It seems our lives depend on it; yet no one can tell what exactly it is.

In order to understand the nature of information, one may have to find some fundamental and primitive notions with which the question can be investigated and explained. The concept of a *sign* is such a primitive notion that serves the purpose. All information is 'carried' by signs of one kind or another. Information processing and communication in an organisation are realised by creating, passing and utilising signs. Therefore, understanding signs should contribute to our understanding of information and information systems (Stamper 1992).

The investigation of *information systems* is an active area where much attention has been paid by other research and industrial communities.

Much discussion has taken place on its pluralistic and interdisciplinary nature and its foundations. An increasing number of researchers and practitioners define information systems as social interaction systems. Social and organisational infrastructure, human activities and business processes are considered as part of information systems. Information systems in this definition can produce messages, communicate, create information, and define and alter meanings. The UK Academy of Information Systems provides the following definition of the domain of information systems (UKAIS 1996). 'The study of information systems and their development is a multi-disciplinary subject and addresses the range of strategic, managerial and operational activities involved in the gathering, processing, storing, distributing and use of information, and its associated technologies, in society and organisations.' Many authors emphasise the importance of information systems study in this postmodern society and its interdisciplinary nature. For example, Avison and Nandhakumar (1995) suggest that information systems encompass a wide range of disciplinary areas such as information theory, semiotics, organisation theory and sociology, computer science, engineering, and perhaps more.

1.2 Problems and challenges in information systems

Information processing has become a major industry which plays a significant role in a national economy (Machlup 1980). Porat (1977), based on analysis of various sources, suggests that in developed countries such as the USA and west European countries, about 50 per cent or even more of the labour force is engaged in information work. However, the following occurrences have been reported in numerous studies: large company budgets for IT (information technology) made, huge expenses on systems development projects spent, and, yet, low economic return from the high investment. Strassmann (1990) argues that there is no guarantee that large investment on information technology will lead to high business performance or an increase in revenue: 'no relationship between expenses for computers and business profitability' on any usual accounting basis across an industry. Some companies use IT well but others balance this positive effect with poor returns on IT investment. Many surveys in industry exhibit evidence that a considerable number of information systems developed at great cost fail to satisfy users or have to be modified before they become acceptable. According to an analysis done by the US General Accounting Office in the late 1980s, among a group of US federal software projects totalling $6.2 million, less than 2 per cent of software products were used as

Shameful numbers

- ☐ 31.1% of projects are cancelled before they ever get completed
- ☐ 52.7% of projects go over time and/or over budget, at an average cost 89% of their original estimates
- ☐ 16.2% of software projects are completed on time and on budget
  - ⇒ In larger companies only 9% of their projects come in on time and on budget with approximately 42% of the originally proposed features and functions
  - ⇒ In small companies 78.4% of their software projects get deployed, with at least 74.2% of their original features and functions

Figure 1.1. Shameful numbers in software development crisis.

delivered, more than half were not used (including those not delivered). The situation has not been improved. In 1993, the Taurus project (Transfer and Automated Registration of Uncertified Stock) of the London Stock Exchange was aborted after the expenditure of £400 million over the preceding five years (*De Volkskrant*, March 12, 1993; *Computing*, March 15, 1993). In 1997, Advanced Technical Strategy Inc. published their study 'The software development crisis' on the Internet, as quoted in Figure 1.1 (Boustred 1997). In this figure, the 'shameful numbers' show a majority of projects cost nearly twice their original budgets. The customer acceptance figures display that more than two-thirds of the software products were never used or never completed. One of the main reasons for these failures is inappropriate users' requirements studies which lead to incorrect systems analysis and design.

The social, cultural and organisational aspects play more decisive roles than technology itself. 'A computer is worth only what it can fetch in an auction', Franke (1987) writes. 'It has value only if surrounded by appropriate policy, strategy, methods for monitoring results, project control, talented and committed people, sound relationships and well designed information systems.' These information systems are composed of organisational infrastructure, business processes and technological systems. An organisation with an improperly designed bureaucracy and unprepared culture will not guarantee that the possession of more information will improve the quality of management; on the contrary, the quality of management will suffer from being overloaded with irrelevant information (see Ackoff (1967)). As Lee (1988) points out, '. . . certainly, this [information] technology has done much to improve bureaucratic efficiency. On the other hand, we claim, it has not helped at all in the management of bureaucratic complexity. Indeed, it has aggravated the problem by obscuring bureaucratic rules and procedures in the form of computer codes. (For instance, many have experienced the frustrations of trying to rectify a computer-generated billing error)'.

There are two other problems observed in systems developments. One is the long lead-time between the commencement of the projects and the systems becoming available. The commencement stage may be to study business strategy of a given organisation, to conduct information systems planning, followed by business analysis, system design, construction design, etc. (see the stages considered in the information systems methodologies framework discussed in Olle *et al*. 1991). Many systems development projects have taken too long in reaching fruition. Hence the immediate risk arises that, during the development processes, the users may wish to change the requirements they specified when the projects began. If the development methodologies do not allow the incorporation of the modified requirements into the developments, then the products after the long lead-time may hardly satisfy the users' present needs.

Another problem observed in practice is related to systems analysis, particularly to analysis documentation. Applications of most development methodologies produce huge volumes of documentation. These documents are supposed to address the technical people involved in the development projects. Because the primary purpose of the analysis documents is usually to function as a formal basis for the next stage, i.e. systems design, the deliverables from analyses are often directed particularly to systems designers in the development teams. The business users, who may actually be the owners of the systems, are not considered as recipients of the information analysis products. As a result, the analysis products are presented in some artificial languages that are specially invented by and for the systems development communities. Syntactic elements such as jargonised notation (of the method of analysis used) and technical disciplines (e.g. logical consistency and data normalisation) receive more attention in those methodologies than semantic aspects. The large volume of documentation may contain a great deal of design symbols and special notation. Each project has to define the meanings of the terms used in the information system in a *data dictionary*. The products of analysis may then be very 'precise' and convenient for the next step, i.e. systems design; they may even be automatically translated into a database schema by a CASE (Computer Added Software Engineering) tool. These documents are too difficult for the systems' owners to understand.

The users need to check above all the meanings incorporated into the specification in order that results of the analysis can be verified. However, documentation in most of the conventional methodologies creates barriers to, rather than facilities for, requirements analysis. Voluminous documentation builds up psychological difficulties; in addition to that, many complicated technicalities have to be learned through undergoing intensive training. Every time the users want to understand a fragment of the representation, the *data dictionaries* have to be used. Even with the help of data dictionaries, the users are still unlikely to be able to make an adequate check on the meanings of words. The risk arising from such a situation is that the users' requirements may be wrongly understood by the analysts. Wrong requirements may be sought, though represented in a correct syntax, as foundations for systems development. In such cases, it is unwise to expect satisfactory information systems to be produced based on these requirement models. As a solution, it is suggested that a method with an emphasis on semantics is needed. Possibilities of preserving and clarifying meanings in requirements analysis should be considered as more important criteria than any others. Therefore, a guarantee of producing a correct requirement analysis is possible only if the users can validate and verify an information systems model, which can be technically implemented. Ideally, this model should be a by-product of the business analysis.

1.3 Approaches and methods for information systems development

Frameworks and methodologies have always been regarded as fundamentals for information systems. Many authors offer comprehensive reviews and critique of frameworks and methodologies for information systems (see, for example, Olle *et al*. (1991), Hirschheim *et al.* (1995), Avison & Fitzgerald (1995)). A suggestion to characterise the methods is made, with three features: *data-oriented, process-oriented* and *behaviour-oriented*. An elaborate analysis and comparison of some methodologies and models for information systems can be found in Hirschheim *et al*. (1995) where methodologies and models are evaluated against some 50 criteria which are grouped into technical, usage, economic and behavioural aspects. It is suggested that there are four paradigms: functionalism, social relativism, radical structuralism, and neohumanism (Hirschheim & Klein 1989; Hirschheim *et al.* 1995). Each information systems development approach may be based on one of the paradigms. Goguen (1992) discovers two cultures in information systems development and requirement engineering: the 'dry' and the 'wet'. Bickerton and Siddiqi (1993) classify some 30 methods for systems development and requirements engineering in a framework in which two of the important considerations are whether the methods are 'hard' or 'soft' in character. More studies and comparisons of information systems development methodologies can be found in the literature.

There are confusions in concepts and theories of information systems, which actually contribute to the difficulties of development and selection of a suitable methodology. The FRISCO task group (FRamework of Information System COncepts) of IFIP WG 8.1[1] has set up as its goal the removal of the confusions in concepts and theories (Falkenberg *et al.* 1998). The following passage (quoted from Lindgreen (1990)) manifests dissatisfaction about information systems development:

> There is a growing concern within IFIP WG 8.1 about the present situation, where too many fuzzy or ill-defined concepts are used in the information systems area. Scientific as well as practice-related communication is severely distorted and hampered, due to this fuzziness and due to the frequent situation that different communication partners associate different meanings with one and the same term. There is no commonly accepted conceptual reference and terminology, to be applied for defining or explaining existing or new concepts for information systems.

A great deal of effort has been put into information systems studies. One of the purposes is that criticism and suggestions regarding the methods can be learned from, and improvements can be made. There are some problems noticed in many of the above mentioned studies. One of the problems is that the majority of the commercially available methods for information systems development are 'dry' or 'hard' in nature; social and organisational aspects tend to be ignored. However, as noted in the preface of Galliers (1987), 'Few would argue that the study of information and the need to treat the development of information systems from the perspectives of *social* as opposed to *technical* systems remains as crucial today as it was when Stamper first published his book in 1973.'[2] Motivated by these same reasons, many people have put effort into bringing about workable theories and methods for information systems development enabling people to handle both the social and technical systems (see, for example, Mumford & Weir (1979), Checkland (1981), Lyytinen & Lehtinen (1986), Stowell (1995)).

[1] WG 8.1, part of IFIP (the International Federation for Information Processing), is one of the Working Groups on Information Systems, and focuses on methodologies and issues of design and evaluation of information systems.

[2] Stamper (1973) began his book with a significant remark that the explosive growth of information technology has not been accompanied by a commensurate improvement in the understanding of information. He appeals for an understanding of both machine and human information systems.

1.4 MEASUR: a semiotic approach to information systems

MEASUR (Methods for Eliciting, Analysing and Specifying Users' Requirements) is a research programme initiated in the later 1970s by Ronald Stamper. The main objective of the programme is to investigate and deliver a set of methods that can be used by researchers and business users in their understanding, development, management and use of information systems. The research programme has evolved in the last two decades. As he later elaborated (Stamper 1993), 'MEASUR' is now a rich acronym:

Methods, Means, Models . . . for
Exploring, Eliciting, Evaluating . . .
Articulating, Analysing, Assessing . . . and
Structuring, Specifying, Stimulating . . .
Users'
Requirements

The various MEASUR methods enable one to start with a vague and unstructured problem, perhaps in a messy problem situation, and gradually 'dry' it out until it is crisp and precise enough to derive a set of technical solutions. MEASUR is focused upon solving business problems in the broadest sense. It helps in solving a wide range of problems, especially those that require organisational or social intervention to solve them. MEASUR addresses information technology problems, as well as organisational problems. In dealing with information technology problems, it can handle the 'up-stream' end of software engineering. In a conventional situation it would primarily serve the managers and other system users, helping them to identify and solve their problems, and lead them to a precise statement of information requirements. In this role it covers the domain of information strategy and planning but it also provides detailed specifications of any problem domain where detailed requirements specifications are needed before designing a computer based system. Further descriptions of MEASUR methods are found in Chapter 4. Here an examination of the major philosophical assumptions will be given.

Stamper (1992) proposed a new paradigm for MEASUR: *the information field*. As opposed to information flow, which is the basis for most of the conventional information systems approaches, this information field paradigm enables to us understand information from a new perspective and therefore to develop information systems more properly. A physical analogy can be used to illustrate this information field paradigm (Huang 1998). Just imagine how a space vehicle in the sky is under the interacting influences of many different internal and external fields: fields of gravitation, electro-magnetic forces as well as clouds of gas, and internal, elastic tensions. There is no point trying to explain the vehicle's behaviour in terms of energy, momentum and materials exchanged between its components – we need a macro model. The information field paradigm is the macro model. It helps us to obtain a macro perspective before one works on details of flows where that is possible.

MEASUR rejects the position taken by many practitioners in this field in that they tend to consider information systems as devices for representing and interacting with some objective reality. It takes a different stance. MEASUR is based on the assumption that the world is constructed socially and subjectively. It recognises that in and out of a business system, there are many actors, or human agents. The owners, managers, staff, suppliers, clients, professional groups, local communities, and so on, are all governed by the forces in the information fields and therefore behave accordingly. These forces are related to their interests, assigned functions, tasks, objectives, personal values and organisational goals. These forces may be present in the forms of formal and informal rules, beliefs, cultural habits and conventions, which can be called *norms*.

This subjective, organisational view of information systems and the information field paradigm lead to the MEASUR approach to information systems work. In the development of an information system, particularly in requirements analysis and representation, the scope of attention should be the whole organisation instead of only the part of the business operations that is going to be automated by the technical systems. The foci of analysis should be actors (i.e. agents) and their behaviour which are governed by social, cultural, institutional, economic and other kind of norms. Therefore, an effective way of representing users' requirement is to describe the agents and their intended patterns of behaviour in terms of social norms.

1.5 About this book

One of the motivations of this book is to examine the relevant issues in information systems development. It will then introduce a semiotic approach with a set of methods initially developed by the MEASUR research team. This team has now grown to a community with colleagues in many countries and the MEASUR methods have been applied in many research and industrial cases. But little systematic account can be found.

This book aims to fill this gap amongst many MEASUR paper publications, technical papers, research reports and project documentation.

The structure of the book is in two parts. Part one (Chapters 2–8) focuses on discussion of semiotics, the semiotic framework for information systems, and the semiotic methods for information systems analysis. Part two of the book (Chapters 9–13) puts emphasis on the application of the methods. It describes how the methods are used in systems analysis, design and implementation of a computer information system.

Chapter 2 is about principles of semiotics and the relevance of semiotics to organisations and computing. Chapter 3 examines different philosophical paradigms and then introduces a semiotic framework for information systems. The semiotic framework represents a radical subjective view of the social world, organisations and information systems. It serves as philosophical and methodological guidance for the development and formulation of semiotic methods to be discussed later. In Chapter 4, an overview of the semiotic methods for information systems, MEASUR, is presented. MEASUR comprises a set of methods covering all stages and aspects of planning, developing and maintaining an information system. Although the rest of the book focuses on the tasks of systems analysis and design, with a detailed account of the methods of *Semantic Analysis* and *Norm Analysis*, the chapter also discusses the use of other MEASUR methods.

From Chapter 5 to Chapter 8, aspects in three layers of the semiotic framework are followed: *semantic*, *pragmatic* and *social* aspects. The reason for such a careful investigation of these aspects is that they are more relevant than another three (*physical, empirical* and *syntactical*) aspects as far as information systems analysis and design are concerned. For example, physical and empirical issues can be important in designing and implementing a computer network, and syntactical issues can be highly relevant in programming and implementing a database. The tasks of systems analysis and design are primarily concerned with capturing requirements, studying business knowledge and representing knowledge at semantic, pragmatic and social levels. Before the semiotic methods are discussed, Chapter 5 explores the basic notions of knowledge representation and information analysis, and presents basic considerations of those methods. When one is equipped with fundamentals of knowledge representation and systems analysis, Chapter 6 begins to tackle the issues at the semantic level. The method of Semantic Analysis, which serves as a core of other methods discussed later, is described here. Chapter 7 tackles the next layer of the semiotic framework, pragmatics and communication. It also examines other approaches before looking at Norm Analysis. Finally in this part, we come to the social layer in Chapter 8, which any technical system should ultimately serve. It is argued that an effective modelling method should enable one to analyse all these issues and it demonstrates that these semiotic methods are capable of this.

The second part of the book (Chapters 9–13) deals with the use of methods in systems development. Chapter 9 displays the performance of analysis to capture semantic and pragmatic information in information model. Chapter 10 shows how semantic and temporal information can be stored in databases, and processed and used with the help of a semantic temporal database language, LEGOL. Chapter 11 exhibits the application of the semantic approach to the entire development process of information systems. In addition, hybrid approaches with input from the semiotic methods are also discussed. Chapters 12 and 13 contain two case studies to illustrate the methods and their applications in systems development projects.

The book also contains two appendices. Appendix A gives a full account of the semantic temporal databases, which helps one to understand and appreciate better the power of such databases. Appendix B presents the CRIS case and illustrates the use of LEGOL.