CHAPTER ONE

Introduction

The core of this research is aimed at introducing an innovative approach (Empirical Modelling) to business process reengineering and the development of associated information systems. This first chapter starts with the aims and motivation for the research and this thesis. It includes the main fields and focus of this research, and gives the preliminary overview of the challenges and potential problems faced in these subjects and sometimes met by the author and other members in this group. At the end of this chapter, the outline of the thesis is presented.

1.1 Research Motivation and Aims

The motivation for this thesis stems from the author's master study in the London School of Economics and Political Science (LSE) during the year of 1995/1996 focusing on the security and management issues of information systems (Chen, 1996). At that time (five years after the first appearance of BPR in 1990), the concept of BPR was still quite new and attracted considerable attention amongst academics, business and industries. The keen interest in BPR can be attributed to the fact that lots of leading companies were anxious to stay close to their customers. BPR was originally developed by former MIT professor Michael Hammer both in his article (Hammer, 1990) and book (Hammer and Champy, 1993). To sum up, BPR is about completely rethinking processes without too much analysis of the existing practices. In BPR, the power of information technology (IT) is a central theme. It encourages the use of IT to radically redefine the business. Although BPR is regarded today as one of the prime managerial approaches for business survival and increasing business competitiveness, still about 50 to 70 percent of the BPR projects have failed to achieve the results at which they aimed. These failures can be partly explained by the poor attempts to gain a complete understanding of the business processes and the subsequent deployment of inappropriate IT systems to support them. One main issue of BPR is to investigate whether or how IT can generate new process designs or how it can contribute to the performance of business processes (that is, to explain the role of IT as an enabler or a supporter). This thesis focuses on the topics of BPR and system development together partly because BPR is tightly connected with IT. And the hardware and software of IT have the ability to support the operational and informational, and even managerial, activities of business processes.

Today, it is widely admitted that the successful deployment of information systems within an organisation depends heavily on a good understanding of the customers' needs and on the quality of the requirements analysis for the supporting systems. That is to say, if the system developed does not meet the actual needs of users, the users may fail to use or even discard that system because for them that system is regarded as an ineffective means of assistance or even an obstacle to the achievement of their goals. Thus during the projects of BPR and the process of developing supporting information systems for the newly designed business processes, the task of requirements analysis is one of the most challenging and the most error-prone activities. One reason for this is that it involves a communication-intensive activity, and usually the business management and personnel (in the rest of this chapter we will call all of them 'stakeholders') lack the clear understanding of what the desired system should do for them and tend to change their minds about the functionalities of that proposed system. Traditionally the analyst elicits requirements from stakeholders in informal ways, synthesises the information and then develops a representation of the system requirements (such as requirements specifications or models). These requirements models developed are typically based on the analyst's personal understanding of the requirements after he has synthesised the information. And the stakeholders are only involved during the requirements determination process[1]. This leads to one of the motivations of this research: to identify an appropriate channel between the stakeholders and analysts for their communication and negotiation, or the process of jointly developing models, during the requirements elicitation as well as the other stages of system development and BPR, which enables the participation, good understanding and speedy acceptance by both parties.

Now let us turn to the issues of software system development. As software systems become more and more complicated, it becomes more and more difficult for designers and programmers to cope with such complexity. Further, some large programs may need several years to develop, and the programmers often meet situations in which some significant parts of the system are still under development with none of their original developers. Normally the principal efforts of system development are to deal with the partitioning of the program and technical knowledge among individual designers. But with each partition, the members of the designing team may change several times during the development, and thus further partition the program.

One way to deal with the complexities of large system development is to abstract and circumscribe the details of programming. This characteristic of abstraction and circumscription is popular in object-oriented development methodologies. This raises one of the motivating issues of this research: the expressive power provided by system development notations, i.e. the richness of the concepts they can offer and in terms of which the customer's requirements can be expressed. Normally these notations reflect the domain of application for which the models has been designed. But what is essential is that the modelling concepts should not only support the modelling of requirements which are inherent to the software system, but also the modelling of the *environment* of that system (people, other devices, etc). To this end, suitable modelling concepts should include facilities for modelling the real-world situations. In the context of the development of information systems to support the business processes, the results of requirements should include the specifications not only of the software to be implemented, but also of the environment around this software, that is the business, which includes the interactions between human and automated agents. This is what many conventional methods fail to achieve, especially those

1. Galliers (1995) further describes this situation as: "the [business managers] are often happy in the mistaken belief that information technology can be left to the technologists, and many [system developers] are happier to have an information systems strategy and an information systems development that are more concerned with technological issues than with business imperatives – with as little as possible involvement from business executives". (p. 52)

focusing more on software development (design and implementation) rather than on requirements. Because they are aimed at, and limited to, the modelling of software artefacts, such as data-flows, database or control and communication processes, they do not offer enough expressive power to capture the real-world happenings.

This raises another issue: that of ‘preconception’ in system development. The task of defining data-flows or communication processes involves preconceiving the rules of event-condition-action through which the action triggering is specified. That is to say, at any moment, when an event occurs and if a condition on the current state is met, then the action should occur. For this, the system designers need to evaluate the set of events which occurred as well as the set of conditions currently prevailing in order to determine the set of actions. But actually no one can preconceive or predict all the causalities in the real world, especially when developing information systems for the use of business or modelling the business situations which consist of human beings. When dealing with the modelling of the real-world things such as human behaviour, we may need to consider the uncertainty which is associated with the occurrence of actions.

The approach taken in this thesis involves a more holistic view of the system development environment, which should also include the business environment in which the system is or will be running. We assume that no methodology provides a complete solution. Instead the focus of the research is mainly on ‘programming through modelling’ rather than on a system architecture which aims to develop new systems.

Exploring the existing environment in a new way will enable us to address the unforeseen complexity in the business domain. This will be critical to the success of BPR. As we can see, business management or BPR analysts do not have the adequate knowledge and understanding of the technology. Software developers may not understand the principles of management enough to know what they should communicate to the managers in order for them to use and manage the IT systems. This situation forms a conceptual gap which divides the world of system design from the world of BPR techniques, i.e. the gap between what the users want and what the software developers think the users want. The existence of this gap shows itself not only in the poor understanding of business processes by the IT experts, but also in the difficulties that BPR experts meet when they try systematically to uncover the business process from the existing software system. This gap leads to difficulty in transferring knowledge between the people who are designing the information systems required for a BPR implementation and their customers.

Mingers (1995) suggests that this situation may be made even more difficult when requirements and design are focusing on technical characteristics of the system. This makes the developers see the situation differently from how it is seen by the users. As he describes:

> Users are concerned with business tasks and objectives and formulate requirements in these terms. They are concerned not with the system itself, but with what it can achieve. This is a problem that the [system] analyst must overcome by trying to see the world as the user does, but conventional methodologies have little to offer in this respect. (p. 20)

From the perspective of participative BPR we are aiming to bridge this gap. During the modelling process (broadly including BPR modelling and system development), all the participants playing various roles (stakeholders, analysts, designers, etc) are expected to apply their appropriate knowledge to the modelling process by constructing artefacts to serve as inputs and providing feedback as well as to generate outputs. The relationships between different roles can be understood through the modelling process under the distributed environment, in order to identify the interactions and in turn enable the appropriate tasks for system development and BPR to be formulated. For example the first level of interactions are between system developers and business stakeholders (cf. Figure 1.1). The responsibility of system developers is to develop the supporting information systems for the business. The developers include software developers and hardware developers, and in the software development organisation the software developers may comprise project managers and software engineers. Then we can find the second level of interactions between the project manager and software engineers. For example, it is the responsibility of the software engineers to communicate the appropriate information about the state of application development to the project managers. This enables the manager to fulfil his responsibility to the clients, which may be business management or personnel. Similarly the business stakeholders can be divided into customers, management and personnel, each of them have a specific set of experiences and beliefs about how the business process is organised and running. For example in BPR, if their values and attitudes are not deeply considered in the redesign of new processes, then their old attitudes or behaviours may conflict with the new process. The distributed tools of EM provide an environment for participative process modelling in which stakeholders, analysts and modellers are represented and included in the process. We can conclude that the increased user involvement in the modelling process will raise the level of participants' contributions in the development of systems. User participation enables the creation of better models than that solely created by the analyst. To this end, the purpose of this research is to investigate how the continuous interaction with and the involvement of users throughout the modelling process of BPR and system development that will increase the probability of implementation success.

The primary purpose of this thesis is to investigate the suitability of Empirical Modelling as the framework for a new approach to software development and business process reengineering. Relevant

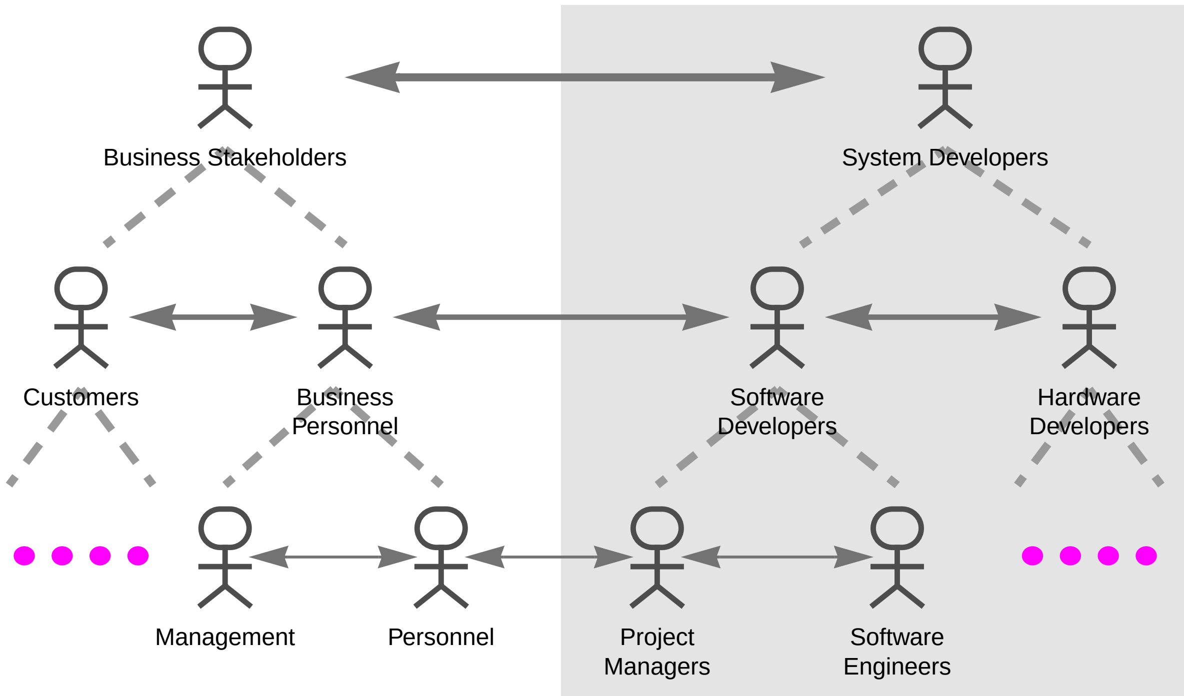

Figure 1.1 Various Roles of Participants in the Process of BPR

researches into EM in system development have been done by two previous PhD students in this group (e.g. Ness, 1997 and Sun, 1999). However we should note that, as has been discussed in these previous research works, EM does not involve the definition of well-structured steps or procedures that follow formalised rules and algorithms, and should not be viewed as a technical process like conventional system development. The emphasis of the research is on the need to improve the modeller's understanding and shape his experience through experiments and observations, i.e. regarding the software system to be developed as a computer-based model, and furthering the development of this software system through modelling. Through the interaction with the computer-based model, the modeller can conduct the modelling activities in a situated manner just as human agents solve problems in the real world. This situated characteristic of the modelling process enables the modeller to design and learn (through the feedback from his interaction with the model), and enrich his experience. In this way the modeller can also adapt the system (the model) to its evolving environment. When adopting EM as an approach to BPR, we need also to consider the interactions between participants. Thus this thesis tries to clarify the characteristics of EM from the perspective of BPR, in particular in relation to user participation in the modelling process of BPR.

1.2 Research Background

The aim and motivation of this research is to extend EM to serve as the process for BPR. The aim is to integrate the EM practice of system development with BPR, through the framework for user participation and the EM tools for the support of interactions among participants for exploring, experiencing and communicating their insights and knowledge. In this section a number of important issues most related to this research are identified.

1.2.1 Business Process Reengineering

Business Process Reengineering (BPR) is concerned with redesigning existing operations in such a way as to exploit new technologies and to serve customers better. It involves radical changes to business processes to achieve dramatic improvements in quality, cost, speed, etc. Today this concept has become popular both in industries and in business management schools. But the term BPR is sometimes used in different senses: it may represent either the cases of minor process improvements or the radical changes in both business processes and organisational structure. Conventionally the process of BPR includes a description of the business activities and the deliverables involved. These deliverables are usually represented in the form of business models which focus on the organisation's structure and dynamics, together with the well-defined and preconceived process for the development of the new business models.

The following summarises some characteristics of BPR that give a preliminary view of how EM is appropriate as an approach to BPR:

- **Process Orientation** (from structure to process): The process orientation of BPR changes the perspective from *structural* relationships between two hierarchical levels of an organisation to the *interaction* processes between individuals and between departments. That is, the principal concern of BPR is to facilitate radical change at process level rather than through organisational functions[2]. One motivation for this is that people in business have started to think "how to develop *new* products or services *effectively* rather than to produce *old* products or services *efficiently*". This explains why the framework presented in this thesis is aimed at the creative development of innovative systems rather than at providing the structured steps and methodical transformation of models between different phases that are characteristic of a conventional development lifecycle. As we shall discuss in chapter 4, EM modelling aims to achieve *effectiveness* – measured by the overall outcome of such processes rather than the speed or immediate results during the process – for the whole development.
- **Customer Orientation**: BPR is radically customer-oriented because the original purpose of BPR is to get closer to customers. That is, the newly redesigned processes should not only sup-

2. The dramatic development of IT has enabled many completely new modes of operation in both business and industry. As a result, many companies are moving toward combination not division of human labour. Thus the previously divided tasks are today being re-unified into coherent business processes.

port the organisation's objectives, but must also satisfy the customers' requirements. But the human factors inside the business should also be taken into account. This broadly includes human, organisational, cultural and political issues. It has been determined that one of the main reasons for BPR failure is the neglect of the human element – most BPR approaches take much account of the scale of change but fail to consider such change through people. As Peltu et al. (1996) comment, "failure to give priority to human factors at a time of radical change can break an existing social contract within an organisation". Thus it is essential that the relevant human agents should be integrated into and involved in the process of redesign.

- **Holistic View of Process** instead of Piecemeal Engineering: Because of its process-oriented characteristics, BPR takes a holistic view of the network of cross-functional processes within an organisation. Such a holistic view can overcome the problem of piecemeal engineering of isolated parts (such as one department) of a business which often results in solutions which are sub-optimal and only appropriate for that part[3]. Such problems closely relate to the issues of systems thinking and reductionism which will be further discussed in chapter 2. As BPR, whether interpreted as continuous improvement or radical innovation, aims at redesigning processes to reduce cost and time and to increase customers' satisfaction and organisational flexibility, a holistic and deep understanding of the business is needed to identify the significant issues. In the thesis, we propose an approach – similar in spirit to the systems approach to be described in chapter 2 – that takes a holistic view of the many different influences on BPR. This is crucial in enabling us to define the relationships amongst IT, human agents and the organisation, and to analyse the nature of these entities and their impact upon an organisation, because these can only be assessed in terms of the total impact.

3. Crowe et al. (1996) point out that most existing computer-based support for reengineering was developed from the bottom upwards, as determined by the traditional product life cycle, and applied piecemeal to closed problem areas as they arise.

1.2.2 System Development

Two activities are sometimes distinguished within system development (or software development): requirements engineering (RE) and software engineering. Normally software engineering refers to the whole lifecycle of software development, and requirements engineering is one central activity in that life-cycle (Figure 1.2). Some researchers separate requirements engineering from software engineering as they argue some problems and questions addressed in requirements engineering cannot be tackled as software engineering[4]. Requirements engineering starts from 'goals' expressed by customers and then elaborates a requirements specification (either in the form of documents or models) in which the system to be developed is defined precisely. This requirements specification[5] is then the starting point of the activity of software engineering, which will develop a software system to meet the requirements. Normally there are the following tasks involved in requirements engineering:

- **Elicitation** The analyst collects the information about the customers goals and problem. This can be done through informal (non-technical) ways such as interviews, discussions, observations or study of relevant documents about the organisation.

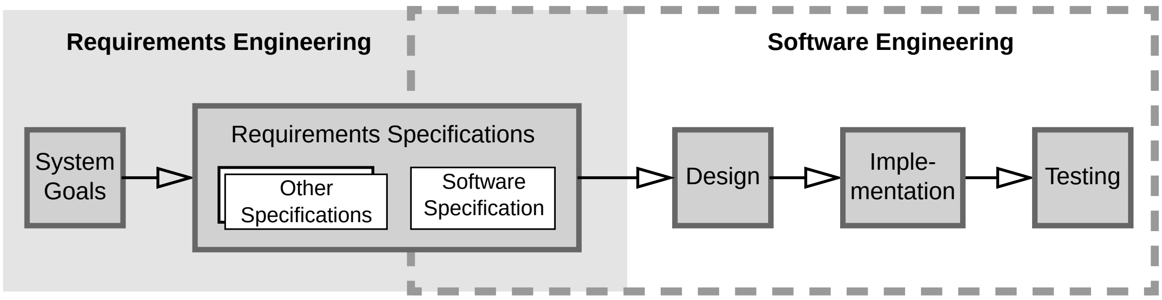

Figure 1.2 Requirements Engineering and Software Engineering

4. This is not the main issue discussed in this thesis. Relevant information can be found in many RE books, *Requirement Engineering* journal, IEEE conferences of *International Symposia on RE* and *International Conferences on RE*, IFIP Working Group (IFIP 2.9), etc.
5. Or precisely, software specification. Because the requirements specification also contains other non-software specifications, for example hardware specifications. Thus in Figure 1.2 the boundary of software engineering only includes the component of software specification in the requirements specifications.

- **Modelling** After collecting the relevant information about the customers' requirements, the analyst will construct models as the representation of the problem domain. Then the analyst studies the models he has built with a view to detecting problems such as inconsistency or the problem of integrating new models with the rest of requirements.
- **Validation** If the analyst detects any problems, he needs to communicate these problems to customers and try to solve them. If no problems are detected, the analyst will formalise the descriptions of requirements and show them to customers in an appropriate way.

Thus the tasks in requirements engineering comprise the interaction between the customers and the analyst in order to progressively clarify the requirements (i.e. in elicitation and validation). But it is obvious that the customers are not explicitly involved in the modelling task. This may lead to discrepancies between the formally described specifications of requirements, and the imprecise, incomplete and even inconsistent goals and wishes of customers in their minds. Further the transition from requirements to design may be more difficult for the end users to follow due to the changes in representation. Also through Figure 1.2, we can find that the customers' goals cannot be directly mapped into the tasks in software engineering (i.e. design, implementation and testing) because customers are not involved in these processes and thus their requirements were abstracted during the modelling in requirements engineering[6]. Various forms of formal modelling methodologies have been developed to support the analysis and verification of the software specifications. Different styles of specification have been proposed (functional, object-oriented or textual like use cases), but none of them can really offer what the customers can use to express their requirements.

Another potential problem is so-called 'requirements creep', which refers to the difference between the requirements specification developed after the requirements procedure and the requirements at the time when the actual product is built (cf. requirement engineering through 'theory-building' in (Loomes and Jones, 1998)). There are many reasons why the problem of requirements creep occurs: the stake-

6. Another aspect of this problem is that we can view the traditional approaches for requirements as relying on a 'snapshot' of a specific state (or at a specific time). The state of the requirements specification only mirrors the real-world state at that time as well as containing other information related to the past.

holders change their minds about the functionality of the proposed system, or (as frequently happen) new or modifying requirements emerge during the processes of design, implementation and testing[7]. It is commonly agreed that this problem is a significant source of excessive cost and time in system development. In today's system development, there is much iteration (e.g. in the Spiral Model) in the refinement process due to the misunderstanding or misinterpretation of the requirements.

The third problem is that the requirements engineering focuses on the system functionality which measures what should be provided by the proposed system in terms of functionalities and services. But the analyst cannot broadly check the adequacy of that system against the constraints within an organisation which are related to business rules or to the way the allocation of the work is organised. That is, the requirements specifications only specifies 'what' the system will do at a high level of abstraction and does not explain the reasons behind that choice of system.

To sum up, better tools and open-ended modelling artefacts are needed to support the communication and negotiation, and the requirements refinement process. The development of an appropriate modelling processes to support the whole lifecycle of software development and maintenance is one of the key issues in this research. It is the aim of this thesis to bridge the gaps mentioned through the concept of 'participative process modelling' which will be detailed in chapter 6.

1.3 Thesis Outline

This thesis draws extensively on the concepts and principles of Empirical Modelling in its approach to system development and BPR. The thesis is divided into eight chapters, of which this introductory chapter gives a background to the thesis. The remainder of this thesis is organised as follows.

7. Davis and Hsia (1994) also give the similar reasons to this problem: (1) Needs are often impossible to realise until after a system is built. (2) Even when requirements are stated up front, it is likely that they will change immediately after deployment. (3) The time between the requirements phase and product delivery is usually too long to pinpoint how specific requirements engineering techniques contributed to a product's success or failure.

Chapter 2 discusses the general concepts of system development. It starts from the concept of 'system' and the systems approach which have affected the thinking in various subjects including the development of computer systems. Then how a computer system is developed and what problems the designers always face when developing the system are discussed. The origins and key concepts of object-oriented (OO) methods are also described in this chapter. This includes the difference in analysis and design between OO and conventionally structured methods, the claims and problems of OO methods, and the influence of OO on system development. At the end of the chapter, the focus is on the concept of circumscription and the details of the link between circumscribing and programming. This includes the discussion of two paradigms: closed-world and open development, and an account of evolutionary design. The discussion of the problems associated with circumscription foreshadows the advantages of developing systems using the EM paradigm, i.e. through the construction of computer-based artefacts to support the open development. This will be further discussed in chapters 5 and 6.

Chapter 3 presents a background of BPR. It provides a survey of the related work in the field of BPR and discusses the problems met during these BPR projects. It is commonly agreed that BPR is important but also problematic for business. Thus it is necessary to identify the factors affecting the successes and failures of BPR, before commencing the reengineering project or proposing an approach to reengineering the processes. The problems facing BPR discussed here are divided into two main categories: the role of IT in BPR and those associated with human factors affecting the success of BPR. The discussion of the role of IT concludes that the development of appropriate IT systems is crucial for the success of BPR work; whereas the discussion of human factors will give strong support to our proposed approach to BPR, which emphasises the significance of user participation in the process of BPR. Furthermore, as the failures of BPR projects are partly due to the use of inappropriate applications or the poor design of supporting computer systems, this chapter investigates the relationship between system development and BPR. This includes the relationship between IT[8] and the organisation, and the deficiencies of conventional system development for BPR.

8. The systems described in the chapter include the existing systems, the future (proposed) systems and the legacy systems which increasingly raise problems in today's business.

Chapter 4 introduces Empirical Modelling as a human-centred and situated computer-based approach to modelling which is being developed at the University of Warwick. It reviews and illustrates the basic concepts and principles of Empirical Modelling which form the basis of this thesis. In addition to introducing the essential concepts, this chapter also describe the construction of EM models – the interactive situation models (ISMs), and the notations and software tools developed in this group to support EM for open-ended development. Finally the modelling process of EM, and its key features, are discussed. The fundamental comparison between the EM process and the phase-based processes of other modelling methods is also given in the last section. On the basis of the discussion in this chapter, the following chapters will consider the use of EM as an approach to BPR. By further comparing EM with object-oriented methodologies, the advantages and limitations of using EM can be made clear. This comparison will be discussed in chapter 5.

Chapter 5 compares the EM approach with object-orientation and with Jacobson's use case approach. The aim of these comparisons is to give insight into how EM differs from other development methodologies in character. However, it is still necessary and important to find an appropriate framework for comparison because these activities are quite different in nature, e.g. modelling vs programming or system development. Firstly in this chapter a summary of the use-case related works, such as Jacobson's OOSE and Object Advantage (Jacobson et al., 1992 and 1995), is presented. This also includes the discussion of the potential difficulties when using use cases in the development of software systems. Then a comparison is made based on two concerns: the first, the different natures of their development processes, involving the construction of the models; and the second, the different natures of the artefacts they employ, viz. the LSD specification, visualisation and animation in EM, and the structure of the artefacts in OO methods (e.g. models built using Unified Modelling Language (UML)). Even though these two concerns are compared separately, they are in fact closely inter-related: the processes and activities of EM and OO are essentially sequences of modellers' situated actions performed on artefacts. That is to say, the character of the processes and other aspects of activities are determined by the nature of artefacts (Ness, 1997).

Chapter 6 brings together the previous two chapters and is the core of the thesis. It considers how Empirical Modelling can be used as an approach to system development and BPR. For developing useful systems, the EM approach is based on the alternative views of the notions of computer, program and programming that are outlined in (Ness, 1997): the *computer* is viewed as *artefact* rather than just an electronic device; *programming* is viewed as *configuring systems* rather than just as software development; and a *program* is viewed as a *system configuration* rather than just the stored program. These concepts and those in chapter 4 together highlight the importance of user participation in the modelling process. The concepts of *participative process modelling* and *participative BPR* introduced in this chapter focus on the activities with which relevant users as well as modellers (both are participants) are involved when employing the approach. This chapter mainly presents the heuristics which guide the modellers through the process of EM. Examples from the case studies in the next chapter are provided to illustrate such heuristics.

Chapter 7 presents examples for the case studies. Two examples are used to illustrate the features of the EM approach as it applies in two different fields: the digital watch example for system development; and the warehouse management system for BPR. Though the examples described here are quite simple, they are significant to illustrate the principles of EM application suggested in chapter 6, and their investigation raises a wide range of challenging issues.

This thesis ends in chapter 8 by drawing conclusions from the discussions and results in previous chapters. It brings together the major findings and summarises the contributions of the thesis. Then it continues with the discussion of the limitations of the present research and proposes suggestions for future research work in the areas of system development and business process reengineering.