CHAPTER SIX

Empirical Modelling for Participative BPR

As business environments continue to change rapidly, it becomes important that the developers of information systems should have a good understanding of the business environment which the systems are designed to support. The importance of linking information technology to business goals and objectives is addressed by the concept of BPR. In response to competitive pressures and changing conditions, many companies are rethinking the way they are doing business. For this, information systems are expected to provide 'innovative' solutions to business problems, rather than merely implementing the usual or known solutions (i.e. simply automating the established processes).

Many methodologies have been proposed for the BPR analysis[1]. However most of them do not address the implementation issues of the development of information systems and the combination of this with the implementation of BPR in detail. That is, the work of business modelling and supporting system development are normally implemented separately by different teams and at different times. As, in BPR, the organisation is regarded as a complex business system, we need an effective method to analyse the dynamic behaviour of the organisation and to evaluate the alternative choices of solutions to problems. Also, as computer systems and applications are currently viewed as transforming the tasks and practices in business processes, we need an artefact for representing, analysing and planning how

1. The Business Processes Resource Centre (BPRC) at the University of Warwick has provided the access to BPR research and the enabling technology providers for BPR analysis and implementation (the hardware, software, or consultancy). The collection of methodologies and tools for BPR analysis can be found under 'Best Practice' at http://bprc.warwick.ac.uk (20 December 2001).

the users' activities and experiences are influenced by the introduction of computer systems or the changes in business processes. In this chapter, we introduce the concept of participative process modelling and participative BPR, because we view system development and BPR as cooperative activities which involve different groups of people – system designers, business experts, end-users or customers – with different competencies, viewpoints and requirements. As no one group can fully understand the practices and meanings of all the others, what we need is to bring their experiences together with the aim of creating a framework for a modelling process which involves all these groups. In this chapter, we will describe a framework, based on the principles and tools of EM, that aims to give a comprehensive – or at any rate representative – view of the real-world situations and systems so as to allow all participants to experience what it is, or would be like, to work within such situations and therefore draw on their tacit and non-explicit knowledge and experience. We propose the application of EM to provide a practical way of implementing participative BPR. This is presented here, and illustrated in chapter 7, as a major part of the original contribution of this thesis.

6.1 Introduction

The role of information technology as the enabler for organisational rethinking has been emphasised in much BPR literature. However when coming to the actual implementation of the BPR project, the implementation issues of IT are usually ignored or addressed by just picking an off-the-shelf application and changing the business processes to fit it. Furthermore, these approaches only attend to the automation of business processes after the BPR implementation; they do not mention the actual procedure of the reengineering work. In chapter 3 we have pointed out that businesses and business processes are such complex systems that the developers and users require appropriate models to understand the behaviour of such systems whether in order to design new systems (or processes) or to improve the existing ones. Most of the BPR methodologies so far have their own systematic phase-based techniques to model business processes. Such 'scientific' techniques may be efficient and necessary if fully automated processes are going to be modelled or reengineered. But it is important that we should take account of the fact that businesses or organisations are not just systems like technical systems; instead such systems include humans participating within them. Thus any attempt at radical modification of the current structure and operations of a business should consider the human factors and enable the participation of all involved if we want to make sure the results are satisfactory and acceptable. This is necessary if we are to avoid a common problem – that newly designed structures or processes may appear to be suitable for business performance ‘theoretically’ but not be applicable in practice due to some factors which were not considered or expected during modelling work.

In defining the requirements for BPR and its support systems, it is essential to have a broad understanding of the organisational environment in order to make appropriate decisions about what changes to make or which parts to retain. There is a problem in requirements analysis, namely, that it is difficult to mitigate the bias of individual participants towards either the current or the future context. It is well-recognised that people have difficulty in envisioning the impact of a proposed system in the future. There are also biases when dealing with uncertainties and risks, especially when multiple agents are involved and multiple changes are happening simultaneously.

In this chapter, we will investigate the potential of computer models as a suitable technique for business modelling. We start by introducing of the concepts of participative process modelling and participative BPR. After the presentation of our framework for cultivating requirements, we will assess its potential as a medium for participation in BPR.

6.2 Participative Process Modelling and Participative BPR

BPR calls for a radical change that involves focusing on the analysis of business processes rather than the functions of a organisation, and reengineering these processes in order to maximise the performance. But the major issue in the BPR exercise is that the scope of the factors concerned is exceedingly broad, and many of these factors cannot be captured in the abstract process described in conventional mathematical models. For example, there are practical considerations which may affect the execution of the process such as the capabilities of the personnel or the environmental conditions and the equipment used. Further, in order to describe and model processes, we must have knowledge not only of their static structure, but also of their dynamic behaviour. Thus good approaches to BPR should focus on the analysis of processes and help to identify the problematic situations and new business opportunities. Besides, as we argued in chapter 3, BPR has more of an organisational focus than a technical one. The effort of BPR is directed at changing people's thinking and thus has to take into account the expectations and viewpoints of people in the organisation, which potentially involves conflict. The concept of participative BPR is thus proposed to answer the high rate of failure of BPR due to the adoption of modelling procedures that do not involve the participation of organisation members. As we cannot reengineer processes which have not been engineered before, we need adequate models for analysing the operations of proposed processes if we want the redesigned processes to perform successfully. We have given a preliminary introduction to our BPR requirements models in section 3.4. We will give further details of how EM has potential for BPR later in this chapter.

6.2.1 Participative Process Modelling

Participative process modelling can be interpreted in two different ways: *modelling of participative processes* and *participative modelling of processes* (Chen et al., 2000b). The term 'process' used here broadly refers to a generic sequence of state changes which follow a reliable pattern[2]. This broad definition is similar to that of Warboys et al. (1999) as they describe in their work that:

> In its loosest sense *process* relates to flux in the real world, to observable progressive changes of state of the world. ... A process is structured change, i.e. there is a pattern of events which an observer may recognise across different actual examples (or occurrences) of the process, or which may be made manifest, or implemented, in many different occurrences. (p. 32)

When we think about how processes are conceived in the business area (i.e. the business processes), there are two kinds of actions involved: the manual actions by human being and the automated actions

2. The processes of changing the state may or may not be necessary to achieve a specific purpose. For example, many natural processes exist and always have existed. But the business processes exist for a specific purpose to change the state of the real world to suit human needs.

by systems. These can be described in terms of two kinds of agency. The first is associated with the *internal* agents which enact the process and which are responsible for the state change. Their interaction can be referred to as a 'participative process'. The interactions of the internal agents may be governed by strict rules and reflect different degrees of autonomy. The second is associated with the external agents whose observation and comprehension of the interactions within a process rely on the integration of many agent viewpoints. Their interaction can be referred to 'participative modelling'. The external agents are typically responsible for designing and managing the process, but may also act simply as observers. As discussed elsewhere in this thesis (see sections 5.3 and 6.4.2), both kinds of agency can be supported in EM (cf. Beynon (1997)).

By way of illustration (cf. Figure 5.7), when a use case is a means of describing a system functionality or a business process, the objects in that use case are internal agents whereas the actors, other systems or other stakeholders, etc. are the external agents for that process. The distinction between internal and external agents can help to refine our thinking about the issue of BPR. The basic tenet of BPR is to gain competitive advantages by rethinking business processes and the use of IT for the redesigned processes. Such rethinking can be made from two directions. External rethinking is made from the external observers (generally from the customers' viewpoint) of the business system and its products or services. The aim is to make the business more responsive and effective. Internal rethinking is made from the viewpoints of component agents of the process. The aim is to find the barriers in that process or any difficulty these members may meet, and thus look for new ways to make the process more efficient. The work of BPR thus should combine both kinds of rethinking: to provide the personalised empowered environment for individuals and to develop the supporting technology for the automation of business processes and the collaborative environment.

But in most BPR contexts, as mentioned in chapter 3, the creation or modification of computer systems to support the newly designed business processes is not suitable or even has a negative impact on the business. This is because the complex issues of human factors are not taken sufficiently into consideration during the BPR work. This is why Warboys (1994) says that, "the general BPR approach could be said to suffer from a tendency to treat organisations as machines rather than ecosystems".

Such rethinking is mainly based on the external viewpoints and the redesigned processes are framed in terms of preconceived interactions to serve particular purposes. So the models are designed for describing 'what' the business process looks like, but cannot express 'why' the process has a certain form, and the motivation, intents, etc. behind the activities. This reflects the potential problem of circumscription in conventional modelling methods which has been described in chapter 2. That is to say, it may be necessary to pre-define personal responsibilities which constrain the individual and define the means of individual cooperation with others to ensure the business system can function appropriately. So we can imagine – as in the age of information technology a large amount of work is performed by individuals at computer terminals – that the constraints, or even the rules, are defined by the capabilities of the information systems which provide the means of interaction among individuals within the organisation. But ignoring such human factors when modelling business processes may be dangerous. Most business processes involve many participants with complex relationships among them, and the results of changing the current process or employing new IT systems are subject to the complexities from these relationships. The human factors associated with these relationships and the insights of individuals should be considered because each of them is concerned about opportunities and seeks to protect or further their interest in the attempt at redesigning the process. Understanding the situations and the usage of systems requires that we understand factors other than the frequency of use for specific features or the patterns of use expected of some users. Without considering or understanding these factors, it will be difficult to integrate existing systems (the problem of legacy systems) or to design new systems to meet the changing needs. When considering the relationship between system development and BPR, Warboys points out the need to consider the following:

- The business is managed as a set of business processes;
- These business processes are modelled in software;
- The IT strategy should be integrated with the business strategy;
- Designing the software is just part of designing the business.

Although the design of the software system does radically impact the design of the business, Warboys further points out two important points ignored by conventional BPR approaches:

- The software system is still a software system with all the old problems of versioning, integration, complexity, etc;
- The rules governing good architectural practices in software are not the same as the rules governing good business structures.

These problems may be even more tricky for the development of business applications, especially when such programs are parallel and asynchronous. Another prime difficulty of designing software systems for business is to achieve a balance between providing flexible and adaptive systems and maintaining and promoting group working (Warboys et al., 1999). This is important for two reasons: because designing support systems is not simply to assemble all the tools and applications and provide the users with an executable access; because there exist dependencies between the group members and each member in the group can achieve more through the cooperative actions than he could do alone. Yu and Mylopoulos (1994) describe the human beings in organisations as social actors who are *intentional* – have motivation, wants and beliefs, and *strategic* – they evaluate their relationships to each other in terms of opportunities and vulnerabilities. Thus when modelling business processes, we should not only describe the entities and the activities (to be automated) within processes, but also the concepts about how, or why, the actors are performing such activities, for example, their goals, abilities or commitments. What we need is a flexible modelling approach that can not only describe the context that the individual members need in order to handle the tasks as freely as possible, but also informs other members in the organisation about how each individual output will affect them. It is important when BPR is seeking new ways for the operation of a organisation, that the goals of each member's interactions are understood by those who participate in them. Without considering such factors, the efforts and results of BPR may simply become using computers to automate existing or outdated processes and will neither increase efficiency nor realise the true potential offered by IT.

6.2.2 Participative BPR

The importance of stakeholders' active participation in the process of BPR has been described earlier in this section and in section 3.4. The purpose of BPR is to employ IT systems to achieve radical change in the business processes and to obtain maximum improvement and performance, and the main potential advantage of our EM approach over conventional methods is in helping the management to find an optimal solution to the many variants of the business problem. As showing in Figures 6.4 and 6.6, conventional approaches involve an abstraction phase to analyse and abstract the problem domain and preconceive new solutions (or systems) in terms of abstract representations such as objects. Under this closed-world paradigm, the end-users are not involved in this abstraction phase, and any further changes to the final product due of their being unsatisfactory will result in another costly development lifecycle. Timetabling can be taken as an illustrative example. Computer-based support for timetabling may be directed at replacing manual processes by algorithms which aim to perform a labour-saving task and reduce the human effort as far as possible. However in the manual timetabling process, human judgements play an essential role which is usually influenced by various environmental factors explicitly or implicitly. Human timetablers explore different possible temporary solutions based on resources available and make qualitative judgements. The judgements they make usually stem from their tacit knowledge which may emerge during or after the construction of the timetable. This tacit knowledge is the kind of knowledge which is used by individuals and is difficult if not impossible to transfer or represent. It is argued by Baets (1998) that if we try to transfer tacit knowledge, then it will become explicit and hence lose its distinctive character. He further concludes that transferring tacit knowledge can only be done via a process of joint learning. That is to say, if people live together through the same experiences, they can learn from each others without making their mutual knowledge explicit. Thus the aim of the EM approach is to provide an environment for participants to communicate and 'visualise' their thinking and knowledge through their interaction with the computer model to achieve the purpose of 'organisational learning' (Baets, 1998). Baets's argument also highlights the potential problem of conventional methods in preconceiving manual processes and explicitly representing human knowledge about timetabling during the abstraction phase. The result is that the human timetabler has to conceive all the possibilities of the timetabling activity during that phase. Through this development the tacit knowledge of human timetabler cannot normally be captured or can only be captured in a limited way.

Some of the considerations that apply to timetabling also apply to BPR. For BPR, any person in a business process has his own perception of that process, and knowledge (include tacit knowledge) about business processes that is closely related to these perceptions. Thus it is important that BPR should take account of this kind of knowledge, and this knowledge can only be considered in the real-world context rather than in the abstract. In this connection, Stowell (1995b) observes that “the incorporation of computing power is more likely to be successful if undertaken by a person with knowledge of the problem than by a person with knowledge of the system”. We also find that the trend in information system development has been to develop systems with languages and interfaces increasingly closer to the forms of human communication and mental representation. For example, by the mid-1990s as Crowe et al. (1996) observe, many facilities for implementation were at the application level and the level of tools and operations were meaningful to end-users. This can explain why today many businesses and industries are increasingly appreciating the importance of involving users in the development process.

We propose the concept of participative BPR which means that people (especially the end-users) participate in the process of BPR in such a way that their individual actions can form an input into the BPR process and also lead to an organisational interaction (and learning) with the environment. All these participants can also get feedback, both from their interactions with other members and from the global responses of the environment, by which they can learn and update their beliefs about the relationships of cause and effect. Because business processes and the process of their changes (during reengineering) are dynamic, it is not sufficient to describe and model these using the abstract (context-free) notations. What we need is a flexible framework to model the continuously changing situations and allow people with different perceptions in the business process to have access to their own environment in the modelling process. The characteristic of our participative concept is that we are aiming to enable the modeller to capture partial solutions, and to explore ‘nearby’ solutions, and opportunistic uses of them, in an experimental manner. This contrasts with the representation of partial solutions in a conventional program which merely traces a path in the construction of a solution from the point of defining the problem to the point of proposed solution in the problem domain. In particular, we are focusing on *situation* rather than *process*, as a situation relates to immediate experience whereas a process is concerned with sequences of situations following a particular pattern.

The aim of this research is to adopt an alternative perspective on computer use which is primarily concerned with constructing computer-based models to support the task of BPR. For this the central issue is the relation between actions which are circumscribed and situated actions where, as mentioned in Chen et al. (2000b), “we have to address unprecedented problems, engage in interactions of a serendipitous and creative nature, and isolate the cues for process interaction from a world where sensation can be overwhelming and action unconstrained”. This issue plays a centre role in participative BPR: we have to consider the subjective nature of an individual participant’s perception as well as their capability. The main consideration is to disclose the tacit knowledge of participants which plays an essential role in the successful execution of business processes.

In the warehouse case study to be discussed in the next chapter, the external objective description of the processes can be represented by the interactions of actors with filling and distributing paper forms. One advantage illustrated in the warehouse case study is that the participants are human, thus we can hope to understand the perceptions and responses of the participants to some degree. The form processing, which abstractly represents the real world activity of the warehouse, involves the physical movement of items, the observation of their location, and the monitoring of their status. Such activity is essentially situated, and is associated with the complex perception of individual participants of many aspects of the current state. The internal insight into the warehouse operation requires an account of each participant’s activity which is based on their direct personal experience of the environment. What we need is a kind of computer-based modelling of processes for understanding and reengineering via participative modelling. In which the aim is to develop a suitable computer-based representation for the state-as-experienced for each internal and external participant.

6.3 The SPORE Framework

SPORE (Situated Process of Requirements Engineering) is a human-centred framework which was initially proposed by Sun (Sun, 1999; Sun et al., 1999). It is problem-oriented because the requirements in this framework are viewed as the solutions to the problems identified in the application domain. The requirements are developed in an open-ended and situated manner. It is open-ended because such requirements cannot be completely specified in advance; and it is situated because the context presented in SPORE is closely connected to the referent in the real-world domain. Within this framework, people participating in the requirements process can cultivate requirements through collaborative interaction with each other and aim to solve the identified problems rather than search for requirements from the 'jungle' of user's needs (Sun et al., 1999).

The SPORE framework for building situated models for the process of requirements engineering is depicted in Figure 6.1. Three kinds of inputs of SPORE are:

- *Key problems* of the domain which are identified by the participants in seeking to address the functional and non-functional requirements of the proposed system. Such identification of problems can occur at any time during the requirements process and is never regarded as completed.
- *Relevant contexts*, such as the organisation's goals and policy, as well as the relationships between participants, which act as motives and constraints for the participants in creating the outputs.
- *Available resources*, such as documents, technology and past experiences of participants, which are used by participants to facilitate the creation of SPORE outputs.

There are also four kinds of outputs from SPORE. The main one is *provisional solutions* to the identified problems which are developed by participants on the basis of the available resources and the current contexts. The other three outputs, which include *new contexts*, *new resources* and *new problems*, combine with their earlier versions and in turn form the new inputs to the model for creating the next outputs. That is, all these contexts, resources and problems are modifiable and extensible in SPORE. Thus participants can develop requirements in a situated manner to respond to the rapid change in the contexts, resources as well as the problems themselves. This framework addresses the concern raised in chapter 2 that requirements are changing all the time and can never be regarded as complete.

6.3.1 The SPORE Framework for Cultivating Requirements

There is no specific activity nor fixed ordering of activities in the SPORE framework, since the problems to be identified during the requirements process are various. For example, some may be difficult to solve given the specific context and the resources available; some are interdependent and need to be solved concurrently. In describing the drawbacks of the closed-world paradigm in system development, we have already concluded that no rule or algorithm can be applied to take all the various factors into account in advance. Different problems need solutions by different methods depending on different situations. That is why the SPORE framework enables the participants to take the current context and available resources into consideration to cope with the diverse and unexpected issues raised during the requirements process.

As mentioned earlier, the main activity of participants in the SPORE framework is to cultivate requirements by interacting with each other as well as with the referent and the external environment to

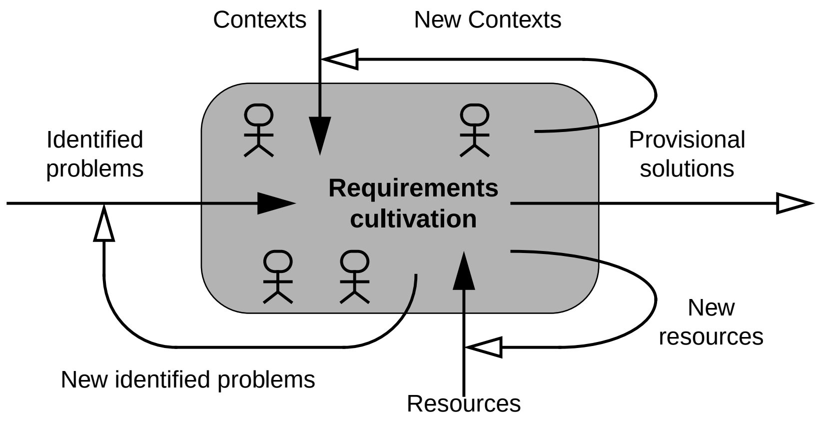

Figure 6.1 The SPORE Framework

develop the requirements (i.e. the solutions to identified problems). Sun et al. (1999) say the term 'cultivation' is used to express the idea that, "requirements (like plants) should grow gradually rather than be conjectured from their initially fragmentary, chaotic and rapidly changing states". Extending the metaphor, Sun et al. make the complementary assumption that the requirements are pre-existent but hidden in the sources just like the grown plants in a huge jungle. So an additional purpose of the SPORE model is to elicit the appropriate requirements just like searching for the right plants from the jungle. The elicitation of appropriate requirements is a difficult and challenging task because there exist various and unexpected factors and elements during the process of requirements, just like searching the right plants from the jungle with a mixture of numerous kinds of elements.

In SPORE the requirements are cultivated through the collaborative interaction among the participants. The focus of the process is neither on the problem domain nor the solution domain, but on the interaction by which the problems are solved on the basis of the current context and the resources available. The participants, who can be designers or end users, are working together to solve problems. In this way the solutions are not just in an individual's mind but distributed among all participants. In addition, the solutions are developed in an iterative and incremental manner through the participants' interaction with each other. Thus the requirements are cultivated through various activities with different purposes. This can also rectify the bias of the insight of an individual participant, which is always limited by his context, expertise, and the resources available at the specific time.

6.3.2 Applying EM to SPORE

The models that feature in a requirements specification in conventional software development are usually too abstract for it to be possible to get a detailed understanding of the model by simply reading it. This situation may be even worse if the application domain is a complex system. Therefore having a way for the participants to experience and visualise the behaviours defined by the model is essential for system development. The SPORE framework involves constructing computer-based artefacts to be used to explore and integrate the insights of individual participants in an interactive manner. The artefacts created are ISMs and are based on the principles and tools of EM. That is to say, within the iterative and incremental framework of SPORE, the ISMs are built in a structured development cycle, and new elements are added into the whole model during each phase (cf. subsection 5.3.1). The insight of the individual participants can be reflected through the consistency between the construal in their mind, the context of the models, and the situation in the external environment. Progress can be achieved through the continuous feedback and evaluation gained by the participants.

In EM the knowledge of participants is constructed in an experimental and not a declarative manner. That is, the insight of the participant is expressed by the coherence between what he expects in his mind and his experiments with the ISMs and the external referents. His insight can also be extended through 'what-if' experiments. Any introduction of new definitions, or redefinition of existing definitions, will evoke changes of state in the models and these in turn will affect the state of his mind through the visual interface. These results of experiments not only change the participant's individual insight but also are stored in his memory, which forms an important resource for the participants to take situated actions. This experimental interaction of EM is particularly powerful in SPORE because the participants can interact with each other as well as with the computer model. Figure 6.2 illustrates how the participant takes situated actions to explore his individual insight by interacting with the computer model as well as the external environment, based on the current context and resources available. Through the distributed EM tools the interaction of an individual participant (i.e. adding or redefining definitions) can be propagated to other artefacts and thus affect the views and insights of other participants. In this way the participants can collaboratively interact with each other through their artefacts.

The distributed version of EM tool – **`dtkeden`** – which was developed by Sun (1999) provides a computer-based distributed modelling environment in which the interaction between modellers is computer-mediated. As the networked computer-based models act as a communication medium for the modellers' interaction, modellers do not need to look at each other or use verbal language for interaction (through this may still be desirable where it is impossible). The visualisation of these models represent the construals of modellers. The 'communication' is achieved through changing the computer models by modellers, and such change is passed through the network to affect other computer models and thus 'tell' other modellers (the listeners) what they are thinking. Four interaction modes have been implemented in **`dtkeden`**:

- The *broadcast* mode: This mode provides a broadcasting function by which any message sent to the server will be propagated to all other clients. In this mode the server acts in the role of message-transferring centre[3]. This supplies an open environment for multi-agent modelling, such as is needed for a multi-user game.
- The *private* mode: In this mode, each client has a private communication channel to the server. In contrast to the broadcast mode, the message sent by a client will not be propagated to the other clients. Because it is possible for more than one private channel to exist in parallel, this private mode is suitable for environments such as computer-supported classroom teaching where many-to-one communication is required.
- The *interference* mode: This mode allows the superagent (in the server model) to directly interfere with the interactions between agents (the clients), or between each client and the server. That is, each request sent from clients is displayed on the server's input window. The superagent can then have discretion over how the suspended request is processed, e.g. to perform 'what-if' experiments.

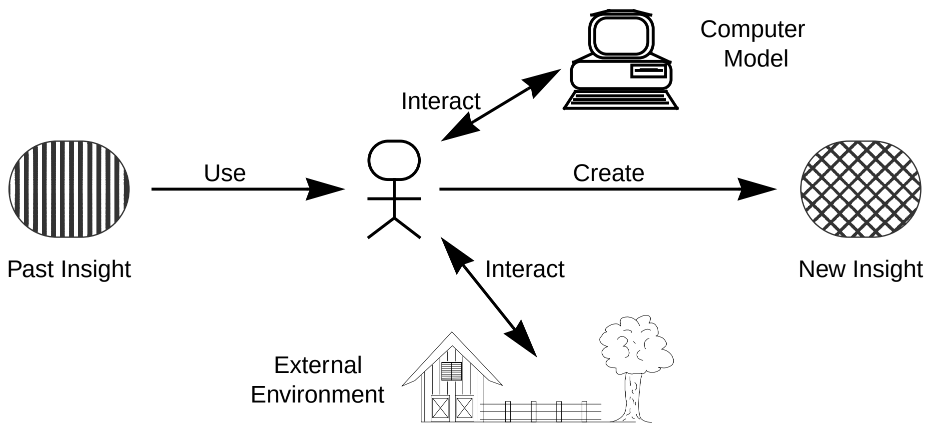

Figure 6.2 The Experimental Interaction of a Participant (adapted from Sun et al., 1999)

3. That is, each message sent from a client will firstly come to the server, and will be automatically broadcast to all other clients and consequently change the visualisation of the server's and clients' computer models.

- The *normal* mode (default mode): In this mode, the interaction between clients is mediated by the computer with reference to specified privileges of modellers to access observables and change the definitions. Any request will be accepted by the server, subject to the current privileges of that agent[4]. This mode is good for hierarchical modelling environments where the relationships between agents are known. The LSD notation supported in **dtkeden** can be used to set up the privileges.

This distributed EM tool supplies the framework for the collaborative environment in which the shared understanding of the key problems and their solutions (i.e. the requirements) can be established (cf. Figure 6.3). The requirements are cultivated through the interaction between participants and the exploration and integration of their individual insights. The greater the consistency between the interactions of the participants in the distributed ISMs, the better their mutual understanding and the more appropriate are the requirements that have been developed. Furthermore, the participants can continually refine their interaction in order to achieve greater coherence and consistency.

The four modes of interaction provided by **dtkeden** address the needs of modellers to participate in a cooperative modelling environment under different situations, or cultivate requirements for different applications. For example, with the broadcast mode of interaction, the modelling environment is very similar to that of an electronic group meeting without video and audio facilities. All the participants share all messages from each member and typically interact with each other in an iterative manner to reach consensus. According to Sun (1999), this mode can be used to develop a system such as a multi-user game which requires a shared environment for supporting the interaction between its users. The private mode is suitable for a many-to-one modelling environment such as the classroom project in which a teacher (at the server) can monitor the learning process of each student. The interference mode enables the server as a superagent to do ‘what-if’ experiment to explore different scenarios, to present significant situations to clients and result conflicts between their viewpoints. For example, where two

4. The access privileges of modellers are given so as to reflect the management hierarchy and social relationships between modellers (Sun, 1999). For example in a system design process, users are not allowed to change the observables associated with implementation; even designers from different groups may need to be restricted to access different observables.

designers propose different values for the same observable, the superagent can arbitrate and suggest or impose a compromise value. The interference mode is particularly useful in modelling the openness of real-world phenomena; the introduction of an unexpected context by a superagent can enrich the understanding of all the modellers involved in the process. It is especially important in representing the perspectives of external observers (in what we have described as participative process modelling in section 6.2) for the process of system development and BPR.

One of the most important benefits of interacting with the ISM, as discussed above, is that it makes the individual insights and the shared understanding visible and communicable. This overcomes the disadvantage of invisibility and incommunicability of shared understanding based on conventionally test-based models. This experimental interaction can also keep the requirements synchronised with the shared understanding among participants, which evolves faster than textual specifications. That is, the way in which the evolution of computer models and the individual insights are synchronised allows the

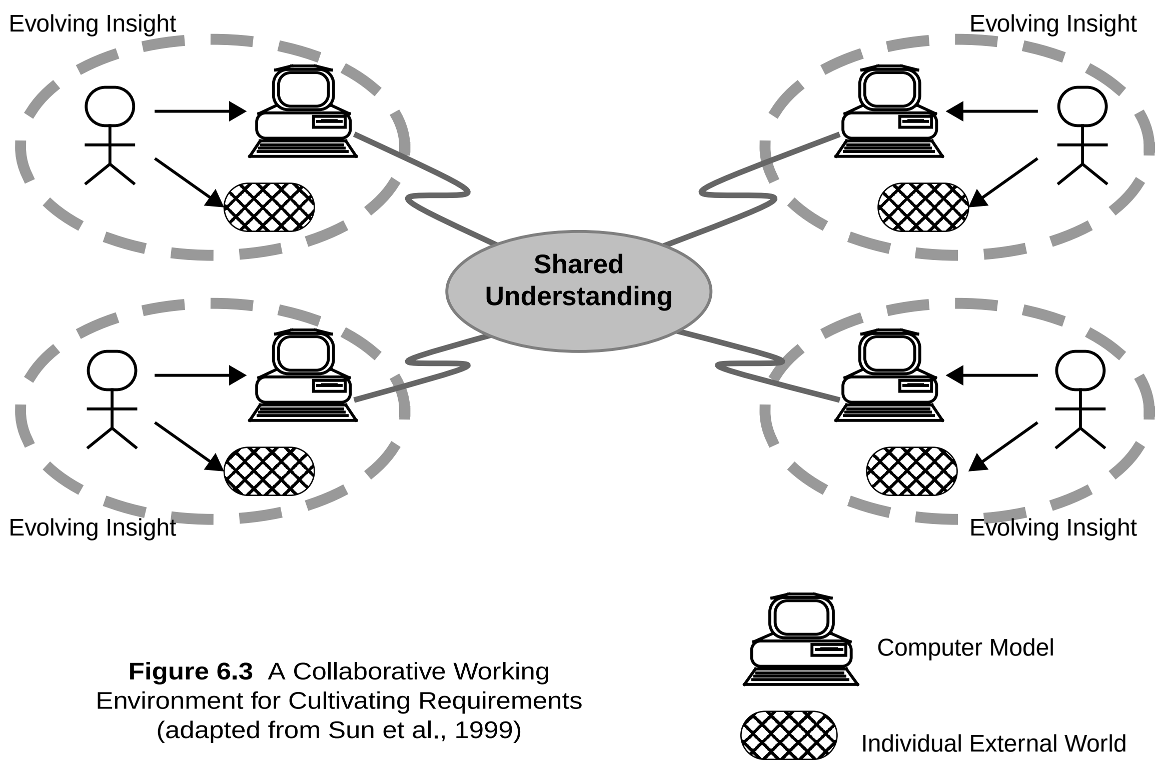

Figure 6.3 A Collaborative Working Environment for Cultivating Requirements (adapted from Sun et al., 1999)

participants to ‘see’ other participants’ viewpoints and to ‘communicate’ with them by interacting with their own artefact (Sun et al., 1999).

6.4 Applying EM to System Development and BPR

The conventional approach to business process modelling is first to make an in-depth analysis of the objectives of that organisation and business needs, then to construct the specification of the requirements, and subsequently to build models based on this specification. Under this paradigm, the modellers have to preconceive all the normal situations as well as the potential error situations or exceptions. This activity is guided only by discussion with the management and sometimes with the personnel who have experience in the practical execution of processes. This is insufficient for business process modelling, because any minor change in the business context which is considered insignificant during the construction of abstractly mathematical models may become the crucial factor for the success or failure of the real execution of newly designed processes. In conventional approaches to process modelling – whether models are abstract (static models) or have the character of simulations (dynamic models) – their construction still relies mainly on the knowledge and experience of the existing pattern of state changes in the real world system. This problematic feature of such approaches is illustrated in Figure 6.4, taken from Fernandes et al. (2000). Conventional BPR approaches involve an abstraction phase in which the business domain is analysed and the new business processes are preconceived. At a later stage these abstractly defined processes are translated into software specifications and protocols for users and actors for further implementation. The real-world concerns (left-hand side) appear in two places: in the initial stage of defining requirements for business, and in the final stage of validation of the built system.

Conventional approaches enable us to model the normal routine of business processes and some exceptional situations to some degree. However, we cannot preconceive in advance all the circumstances which may arise or the hidden factors which may be essential for the successful BPR. The context or the system which can be modelled in the traditional paradigm is the application of the modeller’s pre-existing knowledge about the business process to be modelled. Normally, conventional modelling techniques focus on the activities which can be captured in systematic processes. But, as Evans et al. (2001) point out, such techniques cannot take account of the human observation and assumptions about the context which are tacit in normal successful execution. They cannot also be integrated with the human intervention which is necessary to deal with unforeseen or unpredicted situations. The potential problem in conventional approaches is that the users are not able to give concrete expression of their ideas – in the technical notations with which they are unfamiliar – without being able to intervene and negotiate in the development process. Thus there may be implicit 'politics' which are against the interests of users, and inhibit the capturing of user's expertise in the domain.

In the SPORE approach, the models are intended to deal with the knowledge which is discovered by the participants of the process through observation and interaction in the real-world environment. Our approach focuses on modelling the states as well as the potential agency in the environment where the process to be modelled is enacted. This differs from conventional approaches in which the modelling describes the preconceived patterns of state changes and transitions which define the process in an abstract and implicit manner. No matter how we choose observables to describe the process, there

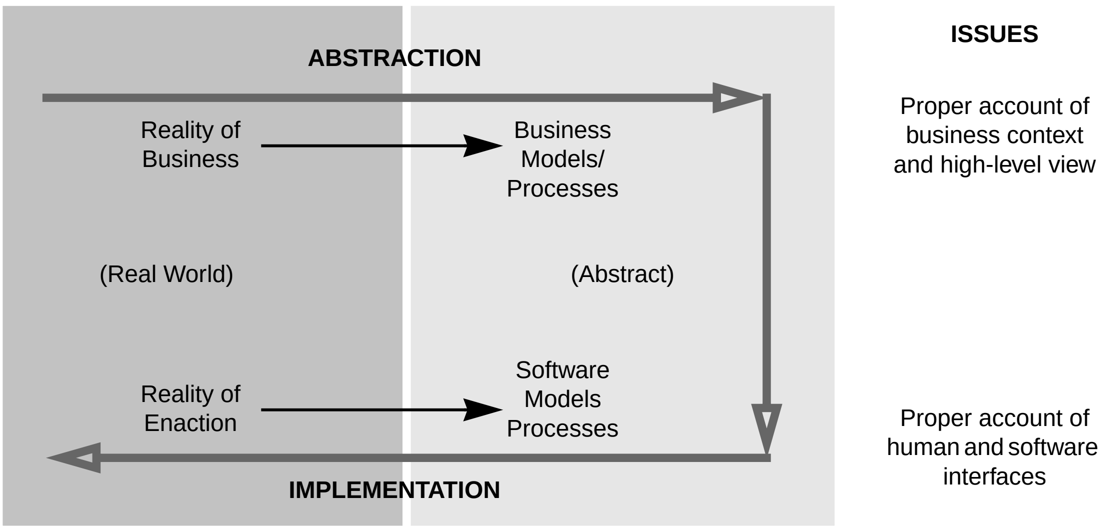

Figure 6.4 The Challenge of Realistic System Engineering (adapted from Fernandes et al., 2000)

will always be observables which will not yet be taken into account. In EM, the states of the model are associated with ‘state-as-experienced’, and only represent implicitly the particular patterns of state change enacted by the modeller. Through the construction of the computer model within EM the modeller can trace the process, and take account of any environment factors which are considered to be relevant, in an open-ended manner. That is to say, the definitive script cannot include all the observables associated with the states to which it refers, but the modeller can interact with the model at any stage and this “allows the choice of observables to be opportunistic rather than constrained by prior commitment” (Evans et al., 2001). This relates to the issue of circumscription which we have discussed in chapter 2.

Following the discussion in Evans at al., we can illustrate the difference between the EM approach and other conventional approaches in process modelling with reference to the context for rework[5] in the manufacturing process. Figure 6.5 shows the three main activities involved in a manufacturing process: the standard work within the normal scope of the process, the routine rework where the nature of the rework problem is familiar, and the exceptional rework where unprecedented or ill-understood problems are encountered. Conventional approaches to process modelling entail viewing the activities from the inner to outer, i.e. normal operation of the process is considered first, then the routine rework, and then the exceptional rework. Such interpretation is based on the presumption that the modeller is familiar with, or has experience within, the process, but this cannot apply to all the cases. In particular, this does not apply when – as in EM – we wish to interpret processes in terms of actual experience, or fashion new processes from raw experience. For the observer who has no experience in the manufacturing environment, it is difficult to distinguish what observations are significant in the process and what are associated with rework. Thus from the EM perspective we interpret Figure 6.5 by viewing the activities from outermost inwards. That is, we identify the process through observation and experience in practice which involves a long process of familiarisation and the observation of the instances of the process in operation (Evans et al., 2001). The identification of the process and the construction of the model is

5. ‘Rework’ refers to the standard processes for recording and/or remedying flaws in manufacturing products, where these have been anticipated.

made by the modeller step by step in an interactive manner which resembles the process of observation and experiment in an unfamiliar and ill-defined environment. This is a natural way of modelling as our conception and evolution of the process are subject to the identification of reliable patterns of interaction from the experience of participants. This evolution and the growing familiarity from participants' experience have the characteristic of ongoing understanding and reengineering activities which is closely associated with the agenda of BPR. In the next chapter we will show how to apply EM principles and tools to a warehouse management system, and how the computer-based modelling can support the execution of processes and the development of a useful system.

6.4.1 Understanding System Environment with LSD

When we redesign business processes and in turn redesign computer systems to meet the new requirements, we usually need to have a broad understanding of the organisational environment in order to make decisions about what changes to make and which parts/components should remain. The reason is that when we define the requirements for the business or its support system, the main work is to investigate the future vision and the impact of the changes. We have described the concept of systems thinking in chapter 2 and emphasised its focus on 'emergent properties' (i.e the ways in which the whole is greater than the sum of the parts). This means that the implementation of system development

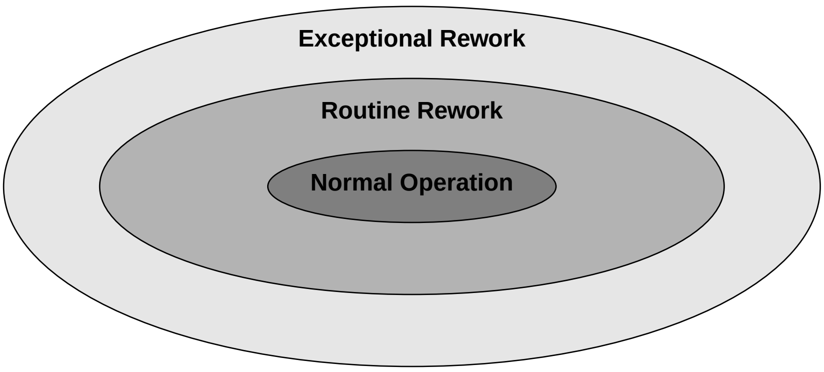

Figure 6.5 Normal Operation, Routine Rework and Exceptional Rework (adapted from Evans et al., 2001)

or business modelling should also refer to the higher level (e.g. on purposes or the future vision) when focusing on relationships at any level. Section 2.1 also mentions the difference between systems thinking and other approaches (e.g. reductionism and holism) that may indicate why conventional methods for system development have failed to meet the actual needs for a business and its users, and why we need the systems approach to analyse the impact of the system upon the business (cf. section 3.3).

The move towards to the future vision must consider the context which is shaped by the past. As Checkland and Scholes (1990) remark of SSM: “it was found useful to think of an intervention in a problem situation as itself being problematical”. This means that for modelling we need to understand the structure of the environment in which we are involved, and our first concern should be to ascertain what our intervention is intended to achieve. Davenport (1993) puts forward similar arguments in respect to BPR. For example, in the framework he proposes for BPR implementation, one step is the understanding of the way existing processes are performed. A view of the existing processes is necessary in order for problems and pitfalls to be identified and eliminated. Moreover, especially in the case of complex and specialised business processes, the reengineering team might need to get familiar with the nature of the current work before designing the new processes. Modelling and understanding existing processes will provide a basis for comparing the impact of new processes on specific metrics.

As Jackson (1995) describes, the pre-existing components of a composite system are the elements of the environment into which a new machine or system is to be introduced. In this respect, Jackson defines two kinds of properties to a composite system:

- properties that the environment intrinsically possesses, in spite of the machine to be introduced (the so-called *indicative* properties); and
- properties that we hope the environment will satisfy after the introduction of the machine (the so-called *optative* properties).

As depicted in Figure 6.6, a model for system development should accordingly have both kinds of properties: the indicative ones, which refer to properties already in the environment, and the optative ones, which refer to properties of the environment that the proposed system will include. In Figure 6.4, we see that there are two kinds of 'world' in system development: the real world (the left-hand side) and abstract world (the right-hand side). The real world includes the real-life context (the upper block) for our daily work and the business activities, and the technical world (the world 'inside' the computer systems) which comprises both the hardware and software. The two kinds of properties, the indicative and the optative, feature in the real world (i.e. the business domain) in which the computer systems are used. The abstract world is mainly shaped by the conceptual analysis and design. It is abstract because many of the complex technical details are 'hidden' in order to enable the designers to control the development more easily and to search for solutions at the logical level. For instance, because of the presumption that as much as possible of the human process will be automated (i.e. become computerised), the workflows (by means of information flows) between human agents are replaced by data flows within the computer-based system.

Conventional approaches to system development and BPR mainly focus on the activities in the abstract world and thus various techniques and tools are developed for the tasks and procedures in this

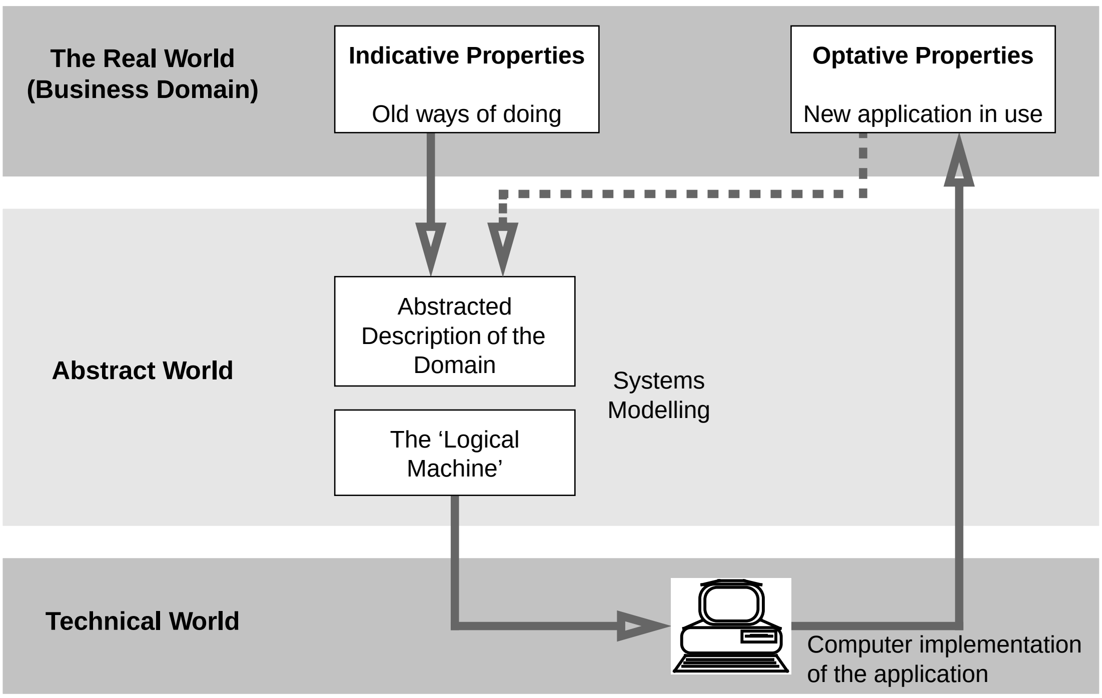

Figure 6.6 The Real World and Abstract World in System Development

world (cf. Figure 6.4). Generally, these approaches can be regarded as product-oriented because they abstract concepts from the characteristics of the systems and consider the usage to be fixed, thus allowing the requirements to be predescribed. But as Figure 6.6 indicates, the business processes in the real world will always be influenced by the use of computer systems and thus change the situations. It is difficult to explicitly define our situations, or the business processes, in a preconceived way.

Many commentators highlight the difficulty on making the transition from the real to abstract world. Crowe et al. (1996) criticise the replacement of human workflows by computer dataflows in the process of automation as too reductionist[6], but also caution against too close an identification of human and machine perspectives that might stem from more holistic view. In this context, the observation of Stowell (1995b):

> It seems clear that the method by which we develop technology-based information systems, and the manner in which we use them, will have a significant effect upon the way the systems behave and also upon the actions of those who access the data held. (p. 121)

and Winograd (1995):

> The overall environment of computer use is a constant co-evolution in which new tools lead to new practices and ways of doing business, which in turn creates problems and possibilities for technical innovation. (p. 73)

are also relevant.

In EM, we prefer to view the development of the computer system and its software associated with human learning and communication that should take place in an evolving world with changing needs. In this view, the business processes should impact on both system development and their use. The focus should be on a better understanding of users' needs – in the real world business domain – and such needs can only be revealed from the business domain in which the systems are going to be used. The

6. And too holistic when with respect to the customer and the organisational environment (treating them as things).

aim of EM is not to provide another efficient system development method by prescribing some technique (i.e. in the abstract world). Rather it is to make the understanding of the business domain (the problem area), open the area for concern for human activities, and the elaboration of the business solutions proceed in parallel with the development of models. For the success of BPR, we need to understand the way the business works. EM not only assists business comprehension but meets the need identified by Stowell (1995b) for analysis and design methods take account of subjective as well as objective viewpoints:

> Explanation is sought from the perspective of the observed rather than the observer, and social reality is considered to be a function of an individual's self-consciousness plus assumptions, together with the shared meanings and beliefs between individuals and groups. (p. 125)

In the rest of this section we will give details of how such business-comprehension activities can be carried out using an EM approach.

The LSD account provides the framework for describing the system environment in terms of agents and their interactions. Agents perform the state-changing actions with different privileges which are constrained by protocols. In LSD, agents are modelled as depending on each other through the observables to which they can response, and which they can redefine. The identification and classification of the agency and observables reflect the modeller's observation of both the model and its referent. The LSD account records the observables which indicate the external view of a modeller about how the agents interact with, and depend on, each other, based on his personal construal of the subjects. It assists the modeller to gain a deeper understanding about the system environment and, through the animation by ADM or EDEN, helps to explore other alternative and sometimes unpredicted patterns of state-change.

LSD encourages the understanding of business situations by focusing on the observation and interaction of agents, which differs from the usual understanding based on activities or entity flows. Modelling the business processes in terms of dependencies and agency within the LSD account can provide a higher level of description that does not presume fully preconceived activity, by which open-ended situations (such as exceptions) can be accommodated more easily. LSD can be regarded as a conceptual model and the semantics used in LSD (such as states, handles, oracles, derivates and protocols) are abstractly representing the concepts of the agent's ability and commitment in the situation. Unlike conventional approaches to business modelling, LSD has the potential advantage of managing large amounts of knowledge and dealing with large scale real-world situations in an open-ended fashion. For this purpose, the LSD account, as well as the EM models, are usually considered as of their essence incomplete. Any new agents or observables which are relevant can be included in the model at any stage.

From Process Redesign to System Development

As explained above, the LSD account frames an understanding of the system environment in a business domain. The next concern is to specify the systems requirements in order to develop the system to support the business. The transaction from business modelling to specifying system requirements is typically not straightforward. That is, we need to go back and forth between business models and system requirements because new issues discovered in one model will require us to go back and look at the other. Throughout the requirements process, the analyst needs to go back and forth between system requirements and organisational requirements in order to deal with the impact of one on the other. Thus the main issue here is how to combine the processes of these two modelling efforts. In EM, the conception of the interactions of internal agents, as represented in the evolving LSD account, can give a good understanding of the system environment. (This is the modelling of a 'participative process' in the sense of subsection 6.2.1). In parallel with this, the system requirements can be cultivated through interaction with the computer model and the associated exploration and integration of individual insights of the participants. (This is the 'participative modelling' of the process as carried out by the external agents in the sense of subsection 6.2.1). We now elaborate on how the EM approach can help BPR and system development by exploiting these two kinds or levels of modelling.

For understanding the system environment (the first level of modelling), we need a model which can capture situations in such a way that the actions and interactions of agents are not circumscribed. This means that they may violate the constraints which are pre-imposed in a conventional approach, so that the state-changes in the model are similar to those in the real business environment, and cannot be totally predicted. For successful BPR, we shall need to analyse such exceptional situations and the behaviours and insights of individual participants. The LSD model is mainly descriptive. It helps the modeller to understand the existing or proposed situations, as well as the conditions, in an organisation and can be seen as serving a strategic function. The LSD account can provide a strategic view of the reconfiguration of relationships associated with adding new dependencies (which typically means introducing new systems or processes, or new arrangements in the work environment in an organisation) in EDEN. Such changes may further alter the scenarios (paths) of business processes which the modeller or management deems to be possible, or impossible. The discovery of such new opportunities is an essential factor for successful BPR. The LSD account is viewed as incomplete because new strategic factors can be added and analysed as required.

The cultivation of the requirements of support systems (the second level of modelling), exploits the distributed variant of EDEN. Agents have states and actions, and their interactions are represented in terms of what they can observe and what they can change (i.e. the ‘oracle’ and ‘handle’ observables). The ISMs offer a higher level view than conventional approaches for requirements analysis and specification through the use of concepts of agency and observation. We have compared our EM approach with object-oriented approaches to system development in section 5.3. We will describe the use of EM in this respect in more detail in the next section.

We use LSD to evaluate the organisational environment and make more permanent revisions; whereas the analysis of support system requirements can be done by temporary redefinitions made directly to the ISM. As the organisational requirements change, they need to be reflected in the requirements of its system. The elaboration of the system requirements will further reveal more issues of the business which need to be addressed. We have described three categories of system boundary (i.e. hard, semi-hard and soft) in subsection 2.1.2 and we can conclude that the boundary of information system development for business should be categorised as the ‘soft’ system. This means the process of business modelling is iterative, as both the organisational requirements and system requirements affect each other and accordingly need to be refined. Thus we need a unified approach to examine the issues of both the organisation and system development rather than different frameworks to be used independently.

6.4.2 EM for System Development

In today’s business environment, information systems are becoming more interconnected to each other and are increasingly involved in complex business processes. That is, the systems used in an organisation (which are broadly viewed as information systems) need to cooperate with human beings to achieve the organisational goals. These systems engage with the business processes that involve customers, staff, etc, which together form the organisational configuration to achieve goals or services. So the information system development occurs in the context of legacy systems and business processes, which involves the issues of systems comprehension and BPR. For this, we can anticipate that the focus of system development will shift towards the understanding of organisational environment and needs, and conventionally focusing on the technical design and implementation in the development of single system will not be sufficient for the needs of today’s business. In determining the requirements of information systems, it is necessary to understand the organisational environment so that the proposed systems (and the existing systems) can work well together with human beings. What we need is an open-ended and flexible approach to modelling the organisational environment and the behaviour of the actors and the support system. In EM, the agents, and how they relate to each other, are characterised in terms of observations. Through the construction of the computer model, the understanding and analysis of agency can be made concrete and amenable.

LSD and ADM in System Development

As discussed earlier, there is a need to have contextual knowledge in system development and BPR. In EM, contextual models operating at two levels are proposed for this purpose: the LSD account is served to represent the organisational environment at an abstract level; whereas animation in the ADM supplies a more concrete model for representing such the information relating to the interactions captured in the LSD account and the interactions of the modeller. Both models enable the modeller to express his contextual knowledge about the real-world situations and thus support the understanding of interactions within the situations.

In the EM approach, the LSD account describes the contextual information about the organisational environment through observation and agency, which represents the modeller's viewpoint and interpretation of the domain. That is, the contextual knowledge of the modeller is represented by defining the actions and relationships between agents by means of observables and protocols. The LSD specifications can be animated through the use of the ADM, which represents the organisational environment on the lower and concrete level. By animating and interacting with the ADM, the modeller can use 'what-if' experiment in order to understand how a certain interaction, or a set of interactions, are performed.

Requirements Elicitation and Validation

As mentioned in subsection 6.4.1, a model should have both the indicative properties referring to properties of the existing environment and the optative properties referring to properties of the future environment in which the proposed system is to operate. We have described the developing procedure characteristic of EM in a previous chapter (cf. Figure 5.3). In the procedure, the elicitation activity (i.e. the arrow of new input) identifies the indicative properties of the existing system (the ISM) and its environment from observation of the real-world situations. It provides the support for producing the conceptual model (the modeller's construal) and generating goals from the interactions and experience of the observed system and model. Through this procedure, the interrelations between the abstract requirements (and goals) and the artefact of real-world referent can be established, and these can be used for the elicitation of system requirements.

Since the conceptual models of participants are too complicated and difficult to represent in notations which all participants can understand, we propose the cooperative validation of the conceptual models within the SPORE framework. This is carried out by participants with different roles who can elaborate different behaviours through ISMs in an interactive experiment. During the experiment, different patterns of behaviour can be explored.

In EM, the construction of ISMs is closely linked to comprehension. Thus the interactions in the real-world domain and the development and validation of ISMs are interdependent. During the validation process (i.e. the arrow of test and experiments), the indicative factors identified during the elicitation activity will be extended and adapted to fulfil the additional optative properties. This validation is mainly achieved by correspondence checks. Once a reliable correspondence between the states of ISMs and those observed in the referent is established, the interaction with ISMs can serve as a representation of the understanding of real-world domain. Thus, through the correspondence checks, the modellers will gain insights into both the existing reality (the indicative properties) and the new requirements or new goals (the optative properties) which in turn form the new inputs for the next phase of system development. With reference to the SPORE framework depicted in Figure 6.1, we find that the 'inputs' to the ISMs depicted in Figure 5.3 (i.e. the arrow of new input (the elicitation activity) and the arrow of test and experiments (the validation process)), correspond to the inputs of the SPORE framework (the key problems, relevant contexts and available resources). Similarly, the 'outputs' of ISMs in Figure 5.3 correspond to the outputs that provide the provisional solutions in the SPORE framework which form the inputs in the cooperative validation of the conceptual models. We regard the development procedure of EM as a continuously evolving process, so that the model is rarely completed. For example, during the elicitation activity the modeller may want to experiment on parts of the model to check and improve its accuracy. Also, during the validation activity, the modeller may need to revisit the situations considered in the elicitation process so as to have a better understanding of the environment. The open-ended characteristic of ISMs provides a way of integrating the elicitation and validation. Prior to defining and establishing the reliable patterns of state change, the interactions with ISMs, similar to activities in our everyday life, have an experimental character and are the primary means to improve our understanding of the domain. Thus the correspondence between the particular patterns of agency, dependency and observation embodied in ISMs and our expectations of state changes in the referent can be established in a natural way.

The development procedure of our proposed approach to building a new system originates from a ‘seed-ISM’ based on considering the current system. The reasons (and advantages) are that the new system, whether we propose to automate or replace the existing processes, will to some degree provide the functionality of the old system. Also, the modeller already has knowledge of problems in the existing system and thus can avoid making the same mistakes again. This strategy for initiating BPR activities does not rule out the possibility that the ISM will be enriched at some stage by knowledge from sources independent of the current system. This will certainly be necessary if the BPR is to involve a radical change.

There are various potential advantages of using the EM approach for requirements elicitation and validation. For example, the artefact provides the concrete explanation for the conceptual model of the modellers. That is, during the procedure the modeller will make a decision about what action to perform and see its impact. Also it allows the modeller to check how accurate and consistent are the correspondences between the computer model and its referent, and between the computer model and his construal. Furthermore, he can check and discuss his construal with other participants and at the same time animate the model. Whilst the development of the conceptual model can draw on ‘direct access’ to real-world examples, it can additionally allow exploration of different variants which may not be predictable or preconceived. Correspondence checks can be used to validate the accuracy and completeness of the model, and lead to a focused elicitation of missing knowledge which in turn form the new inputs to the next development ‘phase’. In this manner the requirements of the proposed system can be driven by several elicitation and validation cycles for evolutionary improvement.

When talking about BPR, there are many disciplines which are considered to be relevant to the issue of BPR. Among these, there are four main fields which we think to be most relevant to BPR: human-computer interaction (HCI), process modelling, requirements engineering, and decision making. For HCI, the ISMs can be used for interface construction; for the development of processes from the

ISMs, an observation-oriented analysis and an associated simulation of behaviour can be provided during the construction of the model; for requirements engineering, the ISMs serve as a prototype which help in understanding the current problems and visualising the reality of the future system; for decision making the ISMs can be used to explore a set of alternative solutions for a problem and their consequences.

The Construction of ISMs as Scenario-Based Design

The construction of ISMs in EM is similar to the paradigm of scenario-based design for human-computer interaction (HCI). The ‘scenarios’, as described by Carroll (1995), are not just the traditional activity logs of human factors, but refer broadly to the artefacts which are meaningful and discussible by users that “are couched at the level at which people understand and experience their own behaviour”. The researchers of scenario-based design[7] argue that computer systems should be viewed as agents that transform user tasks and their supporting social practices. As Caroll asserts:

> When we design systems and applications, we are, most essentially, designing scenarios of interaction. (p. v)

Thus scenarios (especially the user-interaction scenarios) are the appropriate medium for representing and analysing how the computer system might impact on the users’ activities and experiences. Galliers (1995) has described the key features of the scenario-based approach and proposed a way to build up scenarios[8]. The scenarios can be represented in various forms, such as textual narrative (as for example, in use cases), video mockups or computer simulations. On this basis, we can regard the construction of ISMs as (amongst other things) a kind of scenario-based design, since ISMs can serve as a working design representation of the users’ experience with, and reaction to, the system functionality.

7. These include: (1) the ESPRIT research project CREWS (Cooperative Requirements Engineering With Scenarios). Further details can be found at http://sunsite.informatik.rwth-aachen.de/CREWS/ (20 December 2001). (2) The various projects published in *IEEE Transactions on Software Engineering – Special Issue on Scenarios Management*, Vol. 24, No. 12, December 1998. (3) Professor Bob Galliers who was in Warwick Business School and is now in the Department of Information Systems in LSE (cf. Galliers (1991; 1992; 1993; 1995)).
8. He suggests that scenarios can be built up by reviewing (1) those key elements that are expected to remain *constant* during the planning period; (2) those trends that appear likely to continue; and (3) those *issues* over which there is some debate.

Within EM, what the users can do with the existing or new systems, as well as the consequences of their structure, can be analysed prior to the detailed designs of the system functions and features which enable that use. Interacting with the computer models enables the designers to reflect the concrete circumstances and experiences of users throughout the design process.

The ISMs can support the needs of scenario-based design by making it possible to suspend commitment and by supporting concrete progress (Carroll, 1995). The analysis of tasks can be done vividly by the modeller with reference to direct experience of the ISMs, and the form and roles of the system components can be justified by showing what they are used for in a experimental way. The ISMs can also be used to demonstrate design alternatives and expose the analysis behind a particular choice of design. The ISMs are regarded as incomplete and the analysis of the modellers exploits the principles of 'what-if' reasoning. Furthermore, ISMs are an exceptionally promising medium for participative design. They allow the designers and users to communicate in a common medium. As mentioned in the previous section, users may have difficulty in describing their needs in conventional modelling approaches, due to the language barrier associated with the descriptive notations used in functional specifications.

Figure 6.7 illustrates how the development procedure of EM depicted in Figure 5.3 is influenced by the observation of scenarios both in the model and its referent and can be regarded as a scenario-based design. The requirements side in Figure 6.7, represents the observation and experiments of situations (scenarios) of both the model and the referent that can help to identify issues, or criteria, which are then validated against further observations. This corresponds to the arrow of 'new inputs' in Figure 5.3.

The specification of requirements generates the needs, design problems, resources available, etc. for the design stage. Once a prototype with its associated user interface (a 'provisional solution' in Figure 5.3) has been built, the work of evaluation can proceed to validate the design and requirements. The evaluation is done by observation and experiment with the computer model whose focus is on the internal (structural) view of the system. This corresponds to the 'test and experiments' arrow in Figure

5.3. In participative process modelling, as the participants' understanding of the situation increases, their perception of previous activity directed at requirements, design or evaluation may change, and necessitate a return or review of past decisions. Thus the process in Figure 5.3 is sometimes never-ending.

Process Modelling with ISMs

Since the principal concern of BPR is to facilitate radical change, it targets business processes rather than organisational functions alone. Thus the concepts of process and process modelling are the main issues in the area of BPR. However process modelling by itself is not sufficient to define the whole domain of business. Most research on process modelling has concentrated on how to represent processes and the relationships between them by means of structural architectures such as flow diagrams or activity diagrams. But no particular notation for process description is suitable for the needs of all applications.

We have emphasised the process view of systems thinking which regards the organisation as a process (cf. section 3.3), and have also described the contribution that this systems approach to the understanding of how such systems can be developed earlier in this section. In section 2.3 we introduced the concept of E-type software/systems (i.e. system embedded in the real world). Lehman's research shows that even if the requirements of a software system are developed and preconceived by

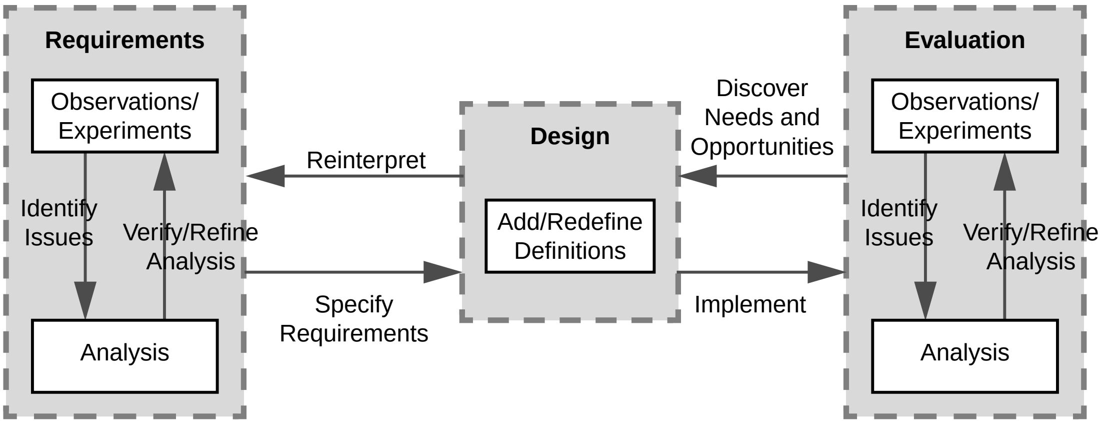

Figure 6.7 The EM Development Procedure as a Scenario-Based Design

carefully exploring the views and needs of the users, the installation of the system will still change the world in which the system and the users operate. As such changes are unpredictable, the users will find their perceptions of the world also change in the new context. Thus the system comes to embody a set of requirements which the user did not identify. This is why the operation domain of the E-type system is unbounded, and its acceptability is determined by its ‘consequences’ (cf. Table 2.3).

The philosophy of the concepts of participative process modelling and participative BPR that we introduced in section 6.2 is similar to the process-view of SSM and the E-type paradigm. As far as proof-of-concept is concerned, the processes represented in ISMs have been thoroughly validated through experiments and long-established practice. What we are representing in our models are the situated activities rather than abstraction of the process involving ‘coordinatising’ all possible situations with reference to selected aspects of state. Where such abstraction is used, the model can only represent the modeller’s understanding at a particular time plus an abstract representation of part of a possible future (cf. Figure 6.6 ‘The description of the domain’). Warboys et al. (1999) suggest that four things are necessary for a model to exist: (1) the part of reality that is the subject of the model; (2) the model itself; (3) the relationship between them; (4) an observer, user, or creator of the model. The EM modelling process described in section 4.3 and the illustration in Figure 4.2 have given the general idea of how our modelling emphasises all these four things and integrates them within the modelling process in a natural way. The key idea is that ISMs serve as an *active model* in the sense defined by Warboys et al., that is, as a model that it is more closely linked to its real-world referent than the conventional passive model[9]. The use of ISMs as active process models in the context of Figure 6.4 is intended to establish a connection between the business domain and the implemented business processes. The development of ISMs also serves as a means of simulating behaviour. This feature stems from the participation of human agents in interaction with ISMs. Further discussion of the construction of ISMs as a process of simulation modelling is the theme of the next section.

9. Warboys et al. (1999) classify three kinds of models for the different uses: (1) passive models; (2) passive models that are dynamic; and (3) active models. They describe the idea of active model is that “it is constructed in a modelling medium which allows the modelling relationship to be maintained even though elements of the subject may change. ... The active model and its subject are synchronised so that the model reflects the current state of its subject”. (p. 43)

The ISM as a Rapid Prototype

In system development, it is difficult to capture the right, or appropriate, requirements due to the different viewpoints of the many interested individuals and the long period of the development life cycle. For example, in the use case model, the development process may generate large sets or versions of use cases. Each use case is associated with a collection of possible scenarios involving the actors and the system, and maintaining this large set of complex scenarios poses a problem. What is more, such text-based descriptions rely on the designer's imagination to fill in the details of the system design and thus only allow the design to be evaluated at an abstract-level. As Crowe et al. (1996) point out, it is because of this that software engineers have turned to prototyping to enable users to experience and experiment with incomplete implementations of a software system before the final system is developed.

In addressing the issue of viewpoint on requirements in EM, the ISMs are the tools for understanding and communication. In the application of EM to BPR, the role of ISMs is similar to that of rapid prototypes in engineering design: they are constructed to inform the design of a complex system and can reduce the uncertainty about organisational or technical requirements[10]. For example, the visual display of ISMs can be the 'mock-ups' that exist on papers or computer screens. Such mock-ups can be used for heuristic evaluation or some testing. Like other business process models, the construction of ISMs does not need to be complete but still helps to give an overall picture to the modeller. The characteristics of an ISM are in some respects different from those of an engineering prototype, which restricts the users to performing actions that are prespecified. The ISMs resemble the artefacts that provide specific information associated with a representative experimental context relating to one or more aspects of a complex system design (cf. the experimental rig that might be used to investigate the effect of temperature on a material to be used in an engineering structure). By constructing ISMs and interacting with them, the modeller focuses on critical issues, and the differences between the states of the model and the states of the referent, without requiring a complete description of the current system or the new sys-

10. Generally the prototype in engineering design works by reducing the complexity of the real systems, either by eliminating functionality or by eliminating the coverage of some parts of the interface (Johnson et al., 1995). In an ISM, the effect of eliminating functionality is achieved (somewhat paradoxically) by enhancing the model through the addition of a restricted interface.

tem. Through this interaction, some aspects of the proposed system which are considered important by designers or users can be highlighted from a subjective viewpoint. This can speed up the interaction between designers and customers in requirements elicitation and advance the system implementation and validation. This endorses Winograd's (1995) view of prototypes as vehicles for communication "both in the designer's interactions with the materials and the user's interaction with the designer". The significance of ISMs is that the interactions made by the modeller are intrinsic to the domain and may only be loosely related to the functionality of the whole system. We can say that ISMs provide an intermediate artefact between declarative and operational types of specification for system development. In applying EM to BPR, the ISMs which are developed in the early stages of understanding business processes serve mainly to capture the open-ended localised interactions with real-world situations, rather than the circumscribed behaviour framed by human agency or agency automated by support systems which are associated with traditional processes or objects. This is consistent with Winograd's view of the organisational context for software systems:

> structures and practices of business are not taken as a fixed environment to be analysed and adjusted to, but as a domain of potential change and new design. (p. 73)

The advantages of interacting with ISMs for supporting design decisions and cultivating requirements can be summarised as the following:

- They can help the evaluation of system requirements in the presence of incompleteness and inconsistency.
- The construction process as prototyping is in the sense of participative modelling in which the user can actively participate in the process of defining the requirements as well as in the evolution of the design. For this reason the system is developed based on the aspects of the requirements which are understood and agreed with the users.
- ISMs as rapid prototypes can help the modeller to estimate the potential effects or costs of the specific requirements, such as unintended side effects of proposed solutions.

- The framework of SPORE can help to define the roles and responsibilities of participants in the process. This can help the project leader in making design decisions (e.g. about the division of work), and help the modeller in validating functions from a user perspective.
- The open-ended and interactive experiments with ISMs can enable the requirements for a proposed adaptation to be evaluated at the early stages. Through various kinds of interactions and analysis, some dynamic aspects of the system requirements can also be explored.

ISMs for Decision Making Support

In the BPR, it is essential that the directions for reorganisation are defined by both the management and designers. To identify these directions, the management has to make decisions about changes which are purposeful but also novel (the tenet of BPR) under uncertain conditions. Such decisions may result in high impact outcomes for the organisation. Thus the models for BPR should also provide support for the management to make decisions. As has been explored in parallel research on EM, the construction of ISMs has some characteristics in common with the development of decision support systems (DSSs) (see e.g. Beynon et al. (2002), Beynon et al. (2000b), Rasmequan and Russ (2000), Rasmequan et al. (2000)). The links between ISMs and DSSs, as they relate to BPR, will be further explored in this section.

As we described in chapter 3, although BPR and DSS have much common with each other, there exists a significant difference between BPR and DSS in their scope of analysis. That is, BPR focuses on the whole organisation (embracing the business processes) whereas DSS focuses on individual decisions (within a business process). With reference to our characterisations of participative process modelling in section 6.2, we find that DSS focuses on the perspectives of the internal agents whereas BPR focuses on those of the external agents. Both perspectives can be taken into account when employing the distributed EM tools. Under the *normal* mode of interaction, modellers can interact with each other in the roles of the internal agents. Under the *interference* mode, they can act as external agents in a context where conflict resolution and the exploration of different situations is orchestrated by an external observer acting in the role of superagent at the server. The EM artefact is in a narrow sense representing the current situations but also can simulate the future situations of an organisation. In a broader sense, it embodies assumptions about processes and actions, the models and procedures for determining the elements of requirements, and the modeller’s decisions, situations and interpretations.

Experimental interactions with ISMs offer the modeller some insights into future situations which help in defining and implementing stages of the system development and BPR. This can give the modeller or the management a clearer idea about the directions for exploration and development, and clarify the actions which need to be taken in order to get closer to desired future states. Open-ended interactions with ISMs provide a flexibility for the decision maker to choose plans. They also enable the modeller to alter the decision factors quickly and assess their impacts. Whether the change of state in an ISM is consistent with, or confounds, the modeller’s expectation, it offers insight that is important for making decisions. Interactions that are consistent with expectations can provide the basis for a step-by-step approach to achieve a desired result. Those that confound expectations can identify the indicators which need to be monitored, and inform the modification of plans that may be needed to correct unexpected behaviour. Recognising the unexpected changes of state which require further assessment of the work is the most important ingredient in this decision making. Another by-product of constructing ISMs is protection against errors of judgement.

As the business environment and the technology is constantly changing, no-one can forecast the future and preconceive the most appropriate solutions for business problems. Thus continuous reviews and corrections are integrated into the EM modelling process in order to build models corresponding to the real-world situations. The models are reviewed and assessed through observations and interactions to determine whether the current plan is appropriate or needs to be modified. The main issue is that the analysis, revision and modification can be made in an efficient and responsive manner within EM. To sum up, the potential advantages of the EM modelling for system development are:

- In the perspective of analysis: to uncover some hidden requirements; to envision future system usage; to make requirements behaviour more concrete; to enrich the contextual information which helps to uncover risks and other organisational (global) problems; and to help communication between participants.

- In the perspective of design: to illustrate the trade-offs between proposed solutions; to validate the design solutions; to understand and resolve the conflicting requirements defined in earlier stages.

6.4.3 Using SPORE for Participative BPR

As described earlier, there is usually some conflict of interest between different levels of hierarchy within an organisation, or even within one level. For example, within a level there are multiple viewpoints between departments or between different user groups. There may also be conflicts between the organisational goals (profit, policy, etc) and the purposes of users (flexibility, convenience, etc). Thus any framework for business modelling should make these different viewpoints and relationships explicit. In the previous section, we have emphasised the importance of gaining a shared understanding of problems in order for participants to negotiate and reach consensus about solutions. Most organisations and most methodologies of system development are also aiming for such understanding. In chapter 5, we have described the problems and disadvantages of conventional models based on natural language and diagrams, which themselves form a boundary restricting the communication between participants and inhibit shared understanding among these participants. Sometimes the models need to be read or understood by non-computer scientists. Especially for BPR, the purpose is to ‘enlighten’ management; so the models need to be accessible to management and to express the process in a helpful way. With the help of EM we propose computer-based models as a new means for making the shared understanding and knowledge visible and communicable. They support the communication amongst people in the organisation that provides the basis for the design process. Under the SPORE framework this leads to the development of techniques which allow information arising from users and the organisation to be used in modelling. The participants (as agents), their activities (the agencies) and other environmental factors form the inputs into SPORE in formulating the requirements. This results in the shared understanding of existing situations through the collection and communication of information about the participants’ insights and tasks. The scenarios shown in ISMs metaphorically represent how these tasks are to be carried out without any further commitment to the detailed implementation of the artefact. Through the development of the artefact, the framework also provides the representation of future situations which are expected to result from the further detailed design of the artefact. Thus we see that our modelling is expressed in terms of agents, agencies and observations, and makes use of some techniques such as role playing, observation and direct participation.

Our research work in EM and the SPORE framework proposes a computer-based modelling medium for the participative work of constructing the artefacts which can faithfully and flexibly represent referents in the real world. It enables the modeller to ensure that people can directly understand how the artefacts resulting from the design can be used. In their account of the merits of rapid prototyping, Johnson et al. (1995) mention their design philosophy in the development of computer systems:

- Good systems design and business modelling should be done through having increased involvement, communication and participation of users and designers in the modelling process.
- The increased involvement and communication are occurring through the greater understanding of users and the greater knowledge of usage of the artefact.

Our approach to participative BPR not only endorses these principles, but also provides a practical technique for the involvement and communication of users participating in the modelling process. Through the process of artefact construction, the systems can be developed in a open-ended and interactive evolution, from the original conception of that systems, with immediate feedback. The feedback may be at various levels of abstraction, for example, via direct experiential knowledge, or conventional mathematical representations.

The Shifting Focus of Participative BPR

The goal of our framework is to facilitate the direct involvement of users in the work of BPR and the design of systems which will influence their working lives. That is to say, people who are affected by decisions should have a voice or opinions in the decision making. The participative process modelling should take place during the entire lifecycle of the business modelling and system development. The various activities in the development procedure, for example from analysis through design to assessment, involves a shift in the conceptual focus. For example, in the analysis activities, the focus is on the existing business environment and the user's working process. This includes the activities of understanding the structure of the business by means of understanding the system environment. This analysis is done through the description of the LSD account. Next, the focus of the design work is concerned with how specific tasks can be carried out under some specific conditions, for example with the technology or resources available. The flexible interactions of participants with ISMs can help the users to keep their focus on the real-world situations and be aware of the constraints from different insights of other participants or from the implementation environment. Finally, the focus of the validation activity is to determine whether a particular design artefact is meeting the user's needs. Normally this testing is done by the developers. But within the participative framework of EM the assessment can be re-situated in the user's working context and enable the users to be involved in this activity. The shift in focus identified above are not interpreted as associated with a sharply defined phases in the modelling activity. In particular, the validation is not a separate activity but instead is the central feature of our process that pervades the iterative participatory design, and is reflected in the interaction of the designers or users with the models at every stage. The computer-based models make it easy for participants to experiment with them and use the validation activity as an opportunity for redesign.

The movement of attention during the participative modelling process is broadly from the business process to the domain of the support systems to be developed. That is to say, the focus is shifting from being process-oriented to being application-oriented. Conventional approaches are not well-suited for the integration of the implementation of business modelling and the implementation of system development. In particular, traditional business models which are process-oriented cannot be easily translated to the domain for software development, because this involves a shift from the concrete to the abstract, whilst the focus of the software development often narrows to a concern for how specific tasks can be done under specific conditions. Another issue is that the abstract notations of system development cannot be well understood by the business stakeholders or other non-computer professionals (cf. Figure

6.6). In the light of the above discussion, the way in which the use of ISMs allows the focus in BPR to shift throughout the development process is one of the most significant potential benefits of EM.

The Characteristics of EM in BPR

From our previous discussion, we infer that EM has great potential for system development and BPR. Through the modelling process within the EM framework, we can have a better understanding of the behaviour of the existing business environment and system, and more clearly identify the tasks, processes, and problematic elements. We can also find, and experiment with, the alternatives more easily. The following summarises the characteristics of experimental interactions with ISMs, as these apply to business process modelling:

- As described in chapter 4, EM is an unified activity rather than the traditional development lifecycle. It can potentially address BPR from the perspectives of both the internal and external agents, and empower and allow participants to “consider the meaning of the task at hand rather than becoming embroiled in peculiarities of its implementation” (Crowe et al., 1996).
- Interacting with ISMs within the SPORE framework allows participants to experiment with any entity of the business system. This means that the behaviour of both computer systems and human components can be identified and thus incorporated in a natural way. Thus the various enablers of BPR, such as information systems or human resource management strategies, can be investigated in essentially the same experimental manner.
- Under the ‘open development’ paradigm, participants can set up problematic scenarios, realise them visually and correct them easily and inexpensively using a ‘what-if’ strategy. Furthermore, the computer-based character of ISMs renders the changes and updates to the existing system apparent and makes the model maintainable and reusable.
- EM helps the participants to communicate their ideas and assess the impact of proposed changes/alternatives immediately. Participants can be guided towards a shared understanding and consensus for decision-making through the continuous evaluation, communication and checks for consistency in the distributed ISMs.

- The visibility and communicability of the interaction within the SPORE framework increases the participants' understanding of roles and relationships amongst others, as well as the effects of individual activities. This lets the participants gain feedback from the results of experiments which is used not only as a basis for comparing the alternative solutions but also as a means of evaluating their validity (through investigating the results of corresponding experiments with the real world referent).
- EM allows the participants to obtain a 'global' view of the effects of 'local' changes made by individual artefacts. This assists the identification of implicit dependencies between parts of the business system.

We have mentioned in the previous chapter that the modellers have no specific role when interacting with ISMs. A family of ISMs built in the SPORE framework can be regarded, according to the different patterns of interaction, as (1) a *requirement*: when we interact in the roles of particular users; or (2) a *system*: when we interact in the role of key components (or agents in EM) and thus explore the internal structure of the model; or (3) a *business process*: when we interact in the roles of workers, markets, suppliers, etc (Chen et al., 2000a). Such flexibility of interpretation according to the style of interaction allows the SPORE framework to be applied to BPR.

We can also identify higher-level functions that a family of ISMs can offer when considering the integration of the IT system with a business system: (1) If we want to integrate the IT strategy with the business strategy, we need a business process model to support. In this case the ISMs can serve as the representation of the business process. (2) If we want to develop software applications to support the business processes (either the existing ones or the new ones proposed from the BPR work), the ISMs in this case can serve as the media for the integration of the business system with the IT system. (3) If we wish to provide a cooperative environment within an organisation, the ISMs can be the means of representing the coordination among the relevant people.

In chapter 3, we proposed that the reengineering work should start by modelling the existing business processes. Under the SPORE framework, this activity can be carried out by simulating the processes by getting participants familiar with the existing process to interact with the computer model and communicate with each other. In the sections on participative BPR (sections 3.4 and 6.2), we have stressed the importance of the key problem with much conventional BPR: the difficulties – but necessity – of involving all relevant people in the reengineering process. We argue that EM is appropriate for participative BPR as it is a human-centred approach and has tools to support the distributed working environment with various patterns of communication and interaction. The aim of EM modelling is not to provide the final answers to problems, but rather to help the participants to make inferences about the different aspects of the system within their individual viewpoints. That is to say, during the construction of ISMs, and through interaction with them, the identified problems and their solutions can be structured, whilst the participants also gain some visual and quantitative information as feedback which is essential for decision-making. The benefits of applying EM to the cultivation of requirements in a participative and distributed manner, as described in section 6.3, can also apply to other BPR applications. Within EM some features of business process modelling, such as making dynamic behaviour explicit or being able to communicate and analyse such behaviour, can be addressed.

The result of experiments with ISMs (the what-if modelling) giving feedback to participants is critical to the success of BPR, as the proposed solutions or alternative business processes can be validated through comparison with the real world system. This validation of the model can indicate the errors of the proposed processes and thus new solutions can be developed based on the feedback. On the other hand, if the proposed processes are validated and accepted by the management, the BPR project can proceed by actual implementation in the organisation. The aim of our framework is to enable participants to gain shared understanding of the problems and to identify solutions which are acceptable to both the management and the end-users and that, at the same time, will not contradict the existing organisational structure nor be subject to the restrictions of the existing systems.

It is clear from the foregoing paragraphs that the openness and flexibility of EM can be exploited to allow participants to develop new versions of models in response to new requirements or changes in the system environment. In an industrial or commercial setting such new versions correspond to the ‘frozen’ stages, or systems, as described in subsection 5.3.1. For medium or large scale applications, it is possible that parts of the model, such as frequently used agent actions, be translated into a conventional language for the sake of speed and efficiency. Then if the whole model needed revision after such a process of optimisation, the developer would revert to the 'pure' EM version while revision to a new version took place. The optimisation could then be re-applied when appropriate. In a large scale application open to many participants creating new versions there is potentially a major problem of version control and management. This could be addressed to some extent by using the version control tools available on a given platform (e.g. SCCS[11] on UNIX) but this is an area for further work.

6.5 Concluding Remarks

Many processes in science and engineering are precisely prescribed by theories and equations, but for business processes it is difficult to consider the process and its associated real-world factors (including human factors) in an abstract and unified way. That is, the correspondence between the states of the process and the states in the real world is hard to represent precisely. Furthermore, business processes cannot be fully prescribed and it is impossible to give a full account of the roles of human beings within them. In contrast to problem solving in science and engineering, we cannot (and never) understand the human activities in enough detail to be able to list the attributes for the proposed system/process and apply modelling approaches which only consider the results of experiments and techniques in a predetermined way. Our best line of attack is to develop the rich and flexible concepts and models to directly involve the potential users and incorporate their activities into the analysis and design of the new system or business. In this context, subject to the modeller having an adequate construal and sufficient understanding of the situation, EM has the advantage of allowing the states of the real world to be modelled in an open-ended fashion so that any new factors considered relevant can be taken into account. This kind of experimental interaction with the computer model seems to be the appropriate technique for the purpose of business process modelling. It can also be used for experimentation purposes and to help decision making during the modelling procedure. By the EM approach singular conditions can be

11.Source code control system.

explicitly modelled and human intervention, which is essential when modelling the scenarios in business, is possible throughout the modelling process.

The role of ISMs in business modelling and system development is as an 'active model' which is similar to the construction of a spreadsheet. The construction of ISMs is guided by the semantic relation between the ISM and its external referent. Referring to Figure 6.4, the understanding of the business domain and the elaboration of the business solutions proceed in parallel with the development of an ISM. It is an essential and distinctive feature of EM that the shaping of the semantic relation between the model and its referent does not shift the focus from reality to abstraction, but rather is established through experimental interactions as the computer model is interpreted as situated in the world. For BPR, the ISM can encompass both the necessary knowledge of business behaviours and provide the prototype of human and automatic agencies in the process.

As the configuration of information systems and human beings in organisations are increasingly becoming both cooperative and distributed, it becomes more important to analyse and model the complex interlinked relationships within business systems. The development of distributed ISMs in the SPORE framework (Sun, 1999) offers an approach that is suited to this purpose.

Through open-ended experimental interaction with the computer model, the adoption of the EM approach to process modelling and reengineering can connect both the business process modelling and system development. Although it may be argued that EM does not handle all parts of the BPR exercise, the techniques we describe in this chapter are aimed at integrating the implementation of the supporting system with the BPR implementation and participative modelling during BPR. Furthermore, there is a potential for integrating EM with other tools used to carry out the modelling and analysis process of BPR. For example, as described in chapter 5, EM can be used as an alternative approach for creating use case specifications that may help to address the problem of managing many variants of use cases expressed in a text-based form.

Where the conception and application of BPR is concerned, we do not regard EM as a methodology or so-called 'scientific method' because its primary concern is for exploratory activity rather than systematic problem-solving procedures. That is, EM is not aimed at developing the final answers/products or detailed specific guidance for designers. Like SSM, EM is concerned with the exploration of different people's systems of meaning but not with describing the 'objective' reality (Mingers, 1995). We are developing a rich and flexible modelling approach and concept to directly incorporate the descriptions of the modellers, as well as potential users, by involving them in the design process. This gives priority to generating a rich understanding of the relevant situation before exploring potential improvement. As Mingers suggests, it also involves considering notional activities which may, or may not, already exist, and may lead to BPR before the information system design is considered. EM principles are based on the presumption that there are always experimental factors in the real-world system of which the modeller is unaware or that they cannot predict. Such factors may play a critical role in the success of system development or business modelling and this can explain why conventional methodologies sometimes fail to achieve their preconceived goals. This is especially true for BPR as we need to take account of other human factors such as knowledge, perceptions and skills. With EM, the construction of computer models can lead to a way to disclose hidden conditions and investigate the matters of agency and observation such as what knowledge or perceptions are associated with a specific role.