Chapter 8

Conclusion

8.1 Research Summary

The research presented in this thesis grew out of the author's previous experience of developing MISs (management information systems), and was initially motivated by his interest in seeking an amethodical approach for SSD. More particularly, the author was attracted by the potential of EM to serve as an open development model for developing an ill-defined, volatile software system (see Section 3.3). As explained in Chapter 1, an amethodical approach of SSD must take both technical but also social processes into account. Since technical support for social interaction in EM is not well developed, the research emphasis shifted to clarifying the distributed perspective on EM and establishing the framework for distributed Empirical Modelling (DEM). The proposed framework and the supporting tool dtkeden have fulfilled the initial objective of the research.

The thesis has examined the following fundamental issues for SSD detailed in Chapter 1: *essential character*, *real-world context*, *human factor*, *social factor* and *computer support*. The main findings of the research may be summarised as follows:

- **Essential character**

  SSD is a human activity but needs technical support in order to enhance the viability of given social goals.

Although an approach based purely on a technical perspective cannot adequately solve the social problems arising from SSD, an approach based purely on a social perspective runs the risk of withdrawing into contemplation and reflection. As recommended by J. A. Goguen in [Gog97], a clear need to reconcile the technical and social issues of information system development (i.e. SSD in the terminology of this thesis) is emerging. Even though the so-called socio-technical approaches have taken this reconciliation into account, they remain deeply rooted in an engineering discipline that can only accommodate social issues to a limited degree.

Instead of regarding SSD as a technical process with social behaviour to enhance the viability of given technical goals, this thesis regards SSD as a social process, but with technical practices to enhance the viability of given social goals. In the light of this fundamental stance, the framework for DEM is intended to address interpersonal interaction, the most primary but difficult social issue in SSD, through providing appropriate technical support (that is, the situated modelling in a collaborative work environment proposed in Chapter 4). By adopting the approach proposed in this thesis – which may be referred to as a 'techno-social' approach to denote its concern for social processes – SSD can move towards realism (from a social process perspective) and practicality (from a technical process perspective).

- **Real-world context**

SSD is highly associated with its context, which must be considered in a situated manner.

The context of SSD is the real-world environment in which the software system is developed and operated. Obviously, it is situated (cf. J.A. Goguen's description of the qualities of situatedness given in Chapter 1) and in most cases cannot be specified in advance. Without taking context, in particular the social context, sufficiently into account, a purely technical view of SSD leads to practical difficulties (such as user resistance and managerial conflict) and has been seen as a distortion in the information technology community [AF95]. Moreover, few models take the context in the operational domain into account. The user who is using the developed system is typically not allowed to change the system in response to his/her evolving requirements. However, faced with a rapidly changing and radically competitive real world, the user often needs to adapt the system in time in order to cope with diverse situations. It is too late and expensive for the user to achieve this need simply through traditional system maintenance. The concepts of design-in-use [SH96, Fis93] and end-user computing [Nar93, DL91] have pioneered this new trend of taking the user's context in the operational domain into account in SSD.

The clear advantage of DEM is that it makes possible to take account of the context not only in the construction stage but also in the operational stage of a software system (see Sections 3.3 and 4.4). The knowledge captured by developers and users from the context in both stages can be structurally coupled with the existent knowledge embedded in computer-based models by means of definitive programming, and in turn used as the base of further interaction. In this way, the context of SSD can be considered in sufficient detail by both developers and users, as well as in a situated manner.

- **Human factor**

  Human agents are the most important dimension in SSD. The enaction of a process model should not hint at any particular roles of human agents in guiding the process of SSD.

It becomes apparent that even the use of the most advanced technologies cannot ensure the success of SSD [Gib94, RHHR98], so the influence of human beings in SSD, in particular in the success of SSD, should not be overlooked [Bro87, DS97, Leh98a]. Although human agents have negative characteristics (such as being error-prone and resistant to change) that are obstacles to the success of SSD, human agents also have positive characteristics (such as intelligence and ability to collaborate) that are very important for successful SSD and cannot be replaced by rigid algorithms and preconceived mechanisms. However, most models for SSD only focus on eliminating the negative characteristics rather than promoting the positive characteristics. They formulate the behaviour of human beings as a set of sequential activity patterns. Like the mechanism of an assembly line, these patterns are well-structured and well-defined as befits an engineering discipline. When enacting these models, human agents are thus obliged to develop software systems by following these rigid activity patterns.

Such rigidity is liable to be untenable in the real world, because the behaviour of human beings is oriented much more toward situated than postulated actions [Suc87]. There is also no reason to prohibit SSD from enjoying the positive characteristics of human agents simply because there are also negative characteristics. In fact, the so-called software crisis, which was often thought to be the result of these negative characteristics, does not result from failing to follow these well-defined activity patterns [cf. Fit96, Gib94]. Instead, it is often because these advanced technologies provide little support for making the best use of the human agents' positive characteristics. As a result, in spite of the use of this kind of model, SSD has yet to dispel the software crisis [Gib94].

Hence, the enaction of a process model should not hint at any particular roles of human agents in guiding the process of SSD. Characterised by the features of a situated activity, DEM allows the developer and the user to interact with the system that is being developed or used in an open-ended manner (See Section 2.2 and 4.2). No rigid activity patterns that limit the best use of the human agents' positive characteristics is given within DEM. The evolution of the system is utterly guided by human agents rather than by rigid algorithms and preconceived mechanisms. More significantly, by means of computer support, the human-centred process for SSD that is invoked by DEM can greatly promote the human agents' positive characteristics with the least risk of resurrecting the problems of an *ad hoc* approach, and at the same time help to reduce the influence of their negative characteristics.

- **Social factor**

Interpersonal interaction is the most critical activity in SSD. A model for SSD should be able to provide effective support for interpersonal interaction to a large extent.

Interpersonal interaction that integrates both cognitive and social processes is the most critical activity in SSD [Pot93, Bro87, RHHR98]. Because of their failure to support effective interpersonal interaction, many projects have been cancelled or abandoned [ITC98, STM95, VPC98]. By drawing on an engineering discipline, traditional models for SSD pay little attention to this crucial activity in SSD. It is presumed that interpersonal interaction can be attained through well-defined, well-structured representational media. However, in practice, obstacles of interpersonal interaction arise very often and pose the main obstacles to the success of SSD [Bos89, Sal87, STM95, VF87, VPC98].

Within the framework for DEM, interpersonal interaction between modellers for shaping the agency of agents in SSD is supported in two levels. Being external observers, modellers can shape the agency in the context of their task roles (such as developers and users (cf. [Son93])) in a concurrent environment. On the other hand, being internal observers, modellers can act as agents to carry out the interaction between agents through pretend play as proposed in this thesis. This being-participant-observer approach serves to shape the agency of agents within the system in their (agents') customary context rather than the modellers' context (see Sections 4.2 and 4.3). By using networked computer-based models to support modelling in these two levels, DEM can provide modellers with computer-mediated interaction that is complementary to traditional interpersonal interaction. In this way, interpersonal interaction for SSD can be supported to an even larger extent.

- **Computer support**

  Computer-based support plays an enabling role in facilitating human agency and promoting interpersonal interaction. With this support in a distributed environment, human agents can explore, expand, experience and communicate shared knowledge in an open-ended, interactive, and situated fashion.

It is evident that hitherto the computer has been widely used to improve human interaction. However, the typical use of a computer as an application tool, in particular for knowledge representation, restricts its advantages. For example, most CASE (computer-aided software engineering) tools use the computer for documentation, automation and code generation. In this sense, the computer is at best simply a powerful word processor that helps its users to organise documents in certain forms and to translate documents from one form (text or diagram) to another form (program code). However, the computer can be best used as an open-ended artefact for facilitating knowledge construction by situated modelling (see Section 3.2) [Cro94, FP88]. In addition, with the aid of network communication, the computer has powerful potential to enhance interpersonal interaction (see Section 4.2) in ways that serve 'to be equipment for language' as highlighted in [WF86, p.79].

The enaction of DEM with computer support is a situated activity. Within DEM, modellers need to create computer-based models to support their situated modelling for SSD (see Chapter 2 and 4). This enables each modeller to explore, expand and experience individual knowledge by interacting with his/her own computer-based model. At the same time, modellers can also interact with each other through their networked computer-based models for exploring, expanding, experiencing and communicating shared knowledge. In this way, human agency and interpersonal interaction can be effectively enhanced and improved in an open-ended, interactive and situated fashion.

The above preliminary findings supply a promising basis for the application of DEM to SSD. A radically new objective in this application is to encourage the use of an open development model (ODM) for a software system throughout its development and use (see Section 3.3 and 4.3). With the openness and situatedness of an ODM, not only the developer but also the user can guide the evolution of the system in response to their rapidly changing needs. In this way, SSD becomes a human-centred activity with autonomy rather than a preconceived mechanism followed by human agents (cf. [Flo95, Flo87]). Moreover, in order to avoid taking the risk of resurrecting the problem of an *ad hoc* approach, it is definitely necessary to provide the computer support for exploiting the positive characteristics and neutralising the negative characteristics of human agents. In this manner, the formal technical issues and informal social issues of SSD can be to a large extent reconciled (cf. [Gog97, Gog96]).

Recognising the importance of the above fundamental issues for SSD in the real world, DEM as an ODM regards a software system as a computer-based model, and enables the developer and the user to guide the evolution of the system with computer support for modelling (see Chapter 2-5). Although the justification of applying DEM to SSD is not clearly given in this thesis, several case studies, such as a hotel booking system (see Section 2.4), railway accident animation and other examples (see Chapter 6 and Section 7.4), have illustrated promising potential for this application. Limitations of research time have prevented the author from investigating many issues in SSD, but the thesis has addressed one of the most difficult parts of SSD (that is, requirements development (see Chapter 7)) and includes practical case studies for SSD (see Chapter 6). Further work on applying DEM to other issues of SSD is on-going in the research group of Empirical Modelling (in Warwick University).

Requirements development is a labour-intensive task and is intertwined with SSD in a symbiotic manner. Its process is largely driven by human interaction. However, like the process of SSD, this process has often been examined from the perspective of an engineering discipline. It is not surprising that the technically-oriented models proposed to prescribe this human-centred interactive process face almost the same problems that arise in many software process models based on step-by-step algorithms. Following the view of SSD as a social process of human interaction, this thesis reengineers the process of requirements development in terms of problem-solving so as to highlight its situated, context-dependent character. The proposed SPORE framework describes this problem-solving process, in which human beings, on the basis of their current context and resources, interact with each other in order to solve problems as they are identified. DEM is applied to SPORE in order to create a collaborative work environment for participants taking part in the process of requirements development. Within this environment, individual insights into requirements and participants' shared understanding of requirements become visible and communicable through the use of networked computer-based models. In this way, the unsatisfactory nature of communication by traditional methods, such as documentation and conversation, can easily be improved, and as a result the communication obstacles within, among and between participants identified in [VF87] can be reduced.

Moreover, the case studies included in the present research (see Chapter 6) show that the tool dtkeden does exhibit DEM and provides a practical exercise environment for *knowing* DEM, in particular for novices. Since this tool is simply an academic product and is not technically elegant, it is inevitable that there will further improvements in its functionality (Section 8.3.3 describes some particular improvements). However, despite the need to improve its functionality for realistic practical use, dtkeden is already a system that effectively supports situated modelling in a collaborative working environment.

8.2 Research Limitations

Undoubtedly, there are several research limitations which need to be acknowledged. In part, these result from the deliberately restricted scope of the research. In part, they are due to the inevitable restrictions of time and resources.

- This research, that aims to establish DEM for supporting the social process of SSD, does not provide ways of addressing most of the social issues associated with this process. Some account has been taken of human interaction and learning, but this thesis does not deal with other important social issues, such as job satisfaction, user resistance and worker democracy. This is because they can only be adequately handled by non-technical methods which are beyond the scope of this thesis. However, the open-ended distributed modelling framework (DEM) and environment (dtkeden) do provide, to some extent, the technical support for such non-technical methods. For example, in order to improve users' job satisfaction, non-technical methods such as negotiation and brainstorming can be supported by 'what if' experiments in the proposed DEM framework and environment.

- Human interaction in the framework proposed for DEM is simplified to four primary modes when it is implemented in dtkeden. Obviously, human interaction is much too complicated to be reduced to these four modes. As D. Sonnenwald has shown in her analysis of both communication roles and task roles [Son93, Son96], the complexity of a participant's communication network is far beyond what has been presented in dtkeden. A possible extension of the existing structure is suggested in Section 8.3.2.
- From a technical viewpoint, it is assumed that dependencies between observables can be formalised by using definitive notations and represented by a computer-based model. However, many dependencies in the real world are very difficult to describe in this format. Even in this context, DEM can allow modellers to undertake social interaction with each other to advance the process of SSD without imposing rigid activity patterns.
- This research has not involved a detailed comparison of DEM with other methods. Each method has its advantages and disadvantages, and the choice of a method must depend on the project itself [AF95]. The main objective of this research, that is, clarifying and enhancing the distributed perspective on EM for supporting a social process of SSD, has been served by developing the proposed framework for DEM. However, further comparison of using DEM and other methods for SSD will help to classify the situations and domains in which DEM can effectively be applied to develop software systems.

8.3 Further Work

This thesis has proposed a framework for DEM to enhance the interaction between a group of people, and a tool to support this framework. The framework has been conveniently applied to requirements engineering in order to facilitate the cultivation of requirements in terms of a shared understanding between participants. The tool dtkeden has also been successfully used to develop several case studies. There remains the potential to

- apply DEM to other issues of SSD (cf. [BCRS98, BCSW99]);
- apply DEM to new subjects which rely heavily on effective interpersonal interaction;
- develop new case-studies and applications with the tool dtkeden;
- improve the functionality of dtkeden;
- evaluate the impact of computer-mediated interpersonal interaction

In the following subsections some suggestions are given for further work based on the research proposed in this thesis.

8.3.1 Possible Applications of DEM

The framework for DEM proposed in this thesis promises to provide a distributed environment of human interaction for facilitating mutual knowledge exploration, extension and communication through networked computer-based models. A main application of DEM to SSD has been achieved in requirements development (see Chapter 7). Other applications that have been initiated include human-computer interaction

[BRSW98] and program comprehension [BS98]. In addition to applying DEM to SSD, this framework is potentially applicable to the following research topics:

- Computer-Supported Collaborative Work (CSCW)

CSCW is a new field concerned with the research and development of software systems to support 'group working'. It usually takes face-to-face communication between developers as its natural form [LG97]. In CSCW applications, a common criticism is that there is a lack of user involvement, which causes severe problems [Kyn91, GKM93]. In CSCW, user involvement requires techniques that enable users to understand the possibilities for computer support and to envision work with a proposed system. Traditional requirements specification is not suited for this purpose, since most users are unable to bridge the gap between description and their professional knowledge and skills. In this case, by using the framework proposed for DEM and the tool dtkeden, further work situations for users can be envisioned through computer-based models, and thus users are allowed to gain hands-on experience.

- Group Decision Support System (GDSS)

Group discussion has become a very prevailing trend for making decisions in an organisation. It can facilitate not only the establishment of group consensus but can also inspire new thinking about innovation, thus enhancing the quality of a decision. Traditional decision support systems are intended to support a decision-maker by providing several alternatives. These alternatives are usually generated by decision models that are built in advance. However, decision-making in an organisation is deeply influenced by its context, in which a lot of people are involved. Accordingly, human interaction may become very critical for making decisions in ways that are more democratic, effective and creative but involve less conflict [WDP88]. The framework for DEM is able to provide such a social forum for GDSS in which participants (who are usually representative of a small group of people) are allowed to interact with each other in their specific contexts. Through the networked computer-based models, sensitivity analysis distributed across different people can easily be achieved. As a result, the influence of each alternative can be clarified, and more significantly, new alternatives can be created to help an organisation survive in today's increasingly competitive business world.

8.3.2 An Further Extension to DEM

The framework proposed for DEM presents a form of human interaction in an organisation. Due to the limits of research time, the interaction between modellers is driven only by their task. Therefore, in the context of modellers, the interaction presented in the framework for DEM is concerned with each modeller's role in performing a particular task, such as the role of a developer, a designer or a user. In the same way, in the context of the agents whom modellers pretend to play, only the task-oriented roles are taken into account, such as those of a train driver, a signalman, a button, or a signal. The interaction between participants arising from their communication (or interaction, in the author's terminology) roles proposed in [Son93] is simplified in this thesis; that is, only their main (or single) role is considered. Figure 5-7 illustrates this simplified idea.

Conceptually, the extension of Figure 5-7 by introducing role-oriented communication can be readily achieved, even though its structure may become extraordinarily complex. When communication roles can be classified and assigned to participants, participants can be divided into several small-scale communication groups (which are not the same as the task-oriented groups shown in Figure 5-7). According to his/her communication roles, each participant can belong to one or more groups and may change his/her roles in a group. Obviously a multi-layered architecture for supporting the complex interaction between participants can be anticipated.

However, the technical support for this architecture needs further research. First, the multi-layered communication environment has to be established. This complex environment may need a more sophisticated configuration for network communication than the star type proposed in Section 5-1. Then, role change in a group must be considered. Security control and richer interaction modes may also require more attention. In addition, if a participant may interact with more than one group at the same time, consideration must also be given to a multi-windows interaction model for the participant and to concurrent interaction across different network configurations.

8.3.3 The Improvement of dtkeden

As indicated in Chapter 3, dtkeden (as well as tkeden) still has some technical limitations which need to be overcome. For example, powerful data manipulation tools and friendly user-interface design tools would be very useful improvements. The former may be obtained by connecting dtkeden with a commercial database tool, and the latter then could be achieved by extending Scout and Donald to become icon/window-based event-driven tools. By characterising Scout with event-driven features, the author has tried to implement some VB-like event-driven functions, such as *click*, *change*, *setText* and *getText*, into Scout windows. The results are promising. These built-in functions allow a modeller to take less account of the internal mechanism of the computer and, more significantly, to define the dependency between a component's state and a built-in action or procedure. For example, by using a *change* event, a modeller can easily define the dependency between the change to the content in a textbox and a particular action or procedure, for example, data checking. This is very helpful for user interface design. Unfortunately, due to limited research time, this extension has not yet been completed.

In addition, an ambitious objective is to rewrite dtkeden in other advanced programming languages and techniques. In fact, attempts to do this are already under way

[Car98]. It is to be expected that these new tools, which are still being researched, may have the potential to provide yet more support for EM and DEM.

8.3.4 Evaluation of Computer-mediated Interpersonal Interaction

Although the author believes that the interpersonal interaction can be significantly improved through the additional use of computer-based models that complement traditional face-to-face interaction, further evaluation would be helpful. This evaluation might be carried out by studying the performance of two groups of people who respectively use dtkeden and tkeden to develop a complicated software system. The interpersonal interaction in the latter case is achieved through traditional face-to-face communication, but it is accomplished in the former case by using both computer-mediated interaction and traditional communication.