9 Conclusion

In this concluding chapter, we shall summarise and reflect upon the key ideas that underlie the contents of each chapter. We shall also review the proposals for further research discussed in previous chapters.

In the Empirical Modelling chapter (chapter 2), we have explained that the philosophical foundation of Empirical Modelling (EM) is based on a commonsense way of understanding phenomena in terms of concurrent agents, observables and dependencies. Based on this philosophy, we have proposed the Definitive Modelling Framework (DMF) as a conceptual framework for supporting the cognitive processes behind our commonsense way of understanding phenomena. The DMF is characterised by its distinctive definition-based agent representation, which supports the use of interactive computer-based artefacts (namely Interactive Situation Models or ISMs) for personal and interpersonal modelling activities. We have also described the practical tool TkEden and the LSD notation as the main means of using the DMF in practice. We can find three levels of abstraction within EM research: the philosophy, framework and practice as represented by current tools and modelling building activities. The philosophy is reinforced by the framework, and the framework is realised by the practice. This raises the question: to what extent are there conceptual gaps between the philosophy and the framework and the framework and its current practical realisation? Where the relationship between the philosophy and the framework is concerned, we believe (from our experience) that the gap is already quite narrow. There is a direct correspondence between the concepts in the philosophy and the framework. This is no surprise because the DMF is the result of unification of ideas specially designed for the philosophy as it has evolved from about 20 years of EM research. The justification of this claim is beyond the scope of this thesis. Where the practical realisation of the framework is concerned, we believe that there is still much to be done to narrow the gap. The discussions in chapter 7 and chapter 8 are especially directed towards this concern. In general, chapter 2 supplies a consolidation on EM concepts which not only serves well as an ‘EM in a nutshell’ style of introduction to EM but also serves as a foundation on which in principle any

EM research can be based.

The System Development chapter (chapter 3) aims to give a 'bird's-eye' view of system development from the EM perspective. The chapter starts with a brief overview of some of the principal strands of research that relate to system development. In particular, system development has been informed by two more or less separate research strands. One approaches system development from 'thinking about systems'; the other from 'thinking about development activities'. We identify the EM perspective on system development with a broader and more comprehensive view to which both research strands are relevant. This view leads to the identification of three intrinsic properties of systems (complexity, predictability and unity) and three aspects of development activities (cognitive, collaborative and methodological). Where complexity is concerned, EM aims to manage complexity by modelling construals from the simplest to the deepest levels of understanding. Where predictability is concerned, EM offers an alternative approach, based on the 'what-if' principle characteristic of the spreadsheet, to the conventional prototyping and testing of system behaviour. It also promises to address issues of system reliability beyond the scope of mathematical analysis and prediction of expected failures. Where unity is concerned, EM presumes that there is no fixed specification for a system (i.e. no fixed system boundary). The system boundary can always be negotiated and can evolve according to the situation. Where the cognitive aspect of development activities is concerned, the EM model facilitates the evolutionary processes of knowledge creation. Where the collaborative aspect of development activities is concerned, the DMF embraces the idea of a hierarchy of human agents in which the negotiation of meaning is facilitated by sharing EM models. Where the methodological aspect of development activities is concerned, the thesis points out that both the standardisation of process (as e.g. in formal methodologies), and of representation (as e.g. in fixed architectural views), have limited application in real world system development. In EM (as is suggested by a growing body of literature), both process and representation should be allowed to evolve according to the current situation. The chapter then turns to a case study that compares EM and an OO modelling with UML. This chapter serves as an introduction to the EM perspective on system development with reference to current state-of-the-art system development research. It is arguably the most comprehensive view of system development based on EM principles so far. In particular, previous doctoral theses that relate to system development using EM only concentrate on the discussion of certain aspects of the topic that are most relevant to their specific research focus.

The Human Problem Solving chapter (chapter 4) discusses what system development using EM essentially entails – computer-supported human problem solving. We have characterised human problem solving using a computer as comprising searches in multiple problem spaces at two levels of abstraction: concrete and abstract. Traditional computer programming primarily supports problem solving either at the concrete level (e.g. spreadsheet paradigms) or the abstract level (e.g. conventional programming paradigms). We believe that EM gives more comprehensive support for problem solving, by assisting the solver in searching problem spaces at both levels of abstraction simultaneously. We argue that building an EM model (namely a Construal of the Problem Solving Situation or CPSS) is an activity well-suited for heuristic problem solving. This is illustrated with reference to a Crossnumber problem and a real-life timetabling problem. As mentioned in the System Development chapter, EM prescribes no method for system development. This chapter elaborates this idea by showing that in EM system development can be guided by situated problem solving heuristics instead of by rational predefined steps.

In the Before Systems: Conceptual Integrity chapter (chapter 5), we point out that conceptual integrity is an important concern before any system is conceived. We have discussed issues that are associated with obtaining conceptual integrity, and explained – and illustrated using the Railway model – how EM can help to address them. The key idea is that conceptual integrity is closely associated with sense-making from a sea of incoherent experiences. The principal concern in the EM perspective on system development is not modelling the functionality of a specific system but on developing a construal of the incoherent experience from which a coherent system conception may emerge. This chapter gives a comprehensive exposition of this idea.

The Beyond Systems: Ubiquitous Computing chapter (chapter 6) is mainly concerned with systems that can never be formalised by developers. The trend towards ubiquitous computing (ubicomp) motivates us to consider systems that can only be formulated when they are used in particular situations. This chapter includes a critical discussion of issues common to many ubicomp research agendas (namely automation, visibility, connectivity and adaptation), and highlights the importance of human involvement in each of these (namely human in the loop, user engagement, understanding and controlling, and user customisation). We propose a new conceptual framework (SICOD) based on EM principles for the ‘development’ of ubicomp systems. We introduce the concept of a ‘soft interface’ as defined by a simple EM model – an Interactive Control Model or ICM – that facilitates the management and customisation of agents (typically ubicomp devices) whose reliable behaviour has been identified. An ICM comprises an ISM that connects a set of Interactive Device Models (IDMs). We have used simple examples to illustrate how a ubicomp system can emerge from constructing a simple ICM. This chapter has explored the potential for applying EM in the modern trend toward everyday life computing. This chapter also discusses many challenges involved in realising the full potential of EM for ubicomp (section 6.4). These challenges motivate future research on this topic.

In the Evaluations and Prospects for EM Tools chapter (chapter 7), we investigate different techniques and tools that can be used to support EM modelling activities. The main concern of this chapter is to minimise the conceptual gap between the concepts in the DMF and the tools. We explore the scope for using different existing technologies for building ISMs, and attempt to construct a simple ISM by using Java, Excel, Forms/3 and TkEden. The general conclusion is that the concepts of OO in Java and first-order functional programming in Forms/3 detract from many essential characteristics of the DMF. Where supporting EM is concerned, TkEden and Excel are the best choices. TkEden is still preferable to Excel because of its support for a variety of special-purpose definitive notations and data types. However, we have identified three major conceptual limitations of TkEden. Firstly, there is no direct mechanism to group definitions and actions into agents. The modeller can only address this limitation partially and indirectly by adding comments to TkEden scripts. Secondly, definitions and actions cannot be updated concurrently. Thirdly, the hybrid concept of procedural variable and definitive variable is confusing. There is no corresponding notion of procedural variable in the DMF. Furthermore, we have identified eight major interface limitations that obstruct the realisation of the DMF. The conceptual and interface limitations of TkEden motivate the development of WING and EME. We discuss features of WING and EME that aim to address most of the limitations. However, due to lack of time, one major limitation of EM tools has remained unexplored – definitions and actions cannot be updated concurrently. This is a significant topic for further research on EM tools.

The Dependency Modelling Tool chapter (chapter 8) describes a new visual tool that aims to enhance the experience of performing modelling activities under the DMF. The motivation behind the Dependency Modelling Tool (DMT) is that there are structures in EM models (namely the dependency, locational and contextual structures) that can be visualised to facilitate model comprehension and reuse. The main characteristic of the DMT is that it allows the modeller to build an EM model as a directed acyclic graph in which observables are nodes, the dependency structure is visualised using directed edges, the locational structure is visualised through the two-dimensional arrangement of nodes, and the contextual structure is visualised using ‘abstractions’. Our experience of using the DMT suggests that visualisation and its direct manipulation mechanisms can greatly help the cognitive process of performing modelling activities, especially where model comprehension and debugging are concerned. DMT models have already been used for understanding many TkEden models created in the past. The development of the DMT has opened up many new research directions. Details are discussed in section 8.6.3. The most significant contribution of the tool is its potential for supporting the kind of end-user modelling that is essential for example for the application of EM in ubiquitous computing.

This thesis is the first comprehensive exploration of the potential of EM to support system development. It goes beyond the traditional conception and approach to system development, and promotes system development from interactive construction of computer-based artefacts. The idea of ‘emergence through use’ is very important in the modern context of systems where the distinction between developer and user is not so clear. With the practical tools developed during the preparation of the thesis, the gap between EM principles and implementations has been narrowed. This provides a solid foundation for future research on applying EM to meet the challenging demands of system development in different domains.