Chapter 7

Conclusions and further work

This chapter draws conclusions from the discussions and results in Chapters 3,4,5 and 6 and addresses the aims given in Chapter 1 and finishes with suggestions for further research in the area of creative software development.

7.1 Conclusions

This thesis has investigated the suitability of EM as a framework for a new approach to SD, that has creative as well as analytical components. This investigation strongly suggests that EM does provide a suitable framework for a new creative approach to SD that combines concepts and principles from EM, PD and software development.

In Chapter 3 the disciplines of EM, PD and SD were compared. It was found that the activity of EM corresponded to conceptual design and that the activity of SD corresponded to conceptually static design in the sense of Pugh [Pug91]. This suggests that SD is associated with the advanced stages of EM when the modeller has committed to a set of artefacts that represent the subject. It was also found that, with respect to subjects, constraints, environments, artefacts, changes and knowledge, there were more similarities between EM and PD than between EM and SD. This suggests that there are some fundamental differences between EM and conventional SD.

Chapter 3 challenges the assumption that the software crisis will be resolved with the emergence of a software engineering discipline that takes an analytical approach to SD. In the chapter it was suggested that conventional SD is similar to conceptually static design. Pugh [Pug91] states that this kind of design tends to be engineering based suggesting that a software engineering discipline might well emerge from the existing approach to SD. However, Pugh [Pug91] argues that engineering based design can only be commercially successful if the product is conventional, such as in the design of the automobile. Since modern computer systems are typically of an innovative nature this would suggest that more is needed than just a software enginering discipline to solve the software crisis. Pugh [Pug91] argues that engineering has to be seen in a broader creative context if innovative products are to be successfully designed and produced. The primary aim of this thesis is to present EM as a framework for just such a creative context for software engineering.

In Chapter 4 the artefacts of EM, PD and SD were compared to determine similarities and differences between them. This was done by searching for the properties essential for creativity as identified by Finke *et al* [FWS92] in their investigation of creative cognition. It was found that the properties of EM and PD artefacts made them more suited to creative discovery than the artefacts of SD. It was also found that the artefacts of EM and PD had properties similar to the formal models of SD. This suggests that the artefacts of EM can have a mixture of properties to do with creativity and analysis.

In Chapter 5 the actions of EM, PD and SD were compared to determine similarities and differences between them. This was done by reconstructing activities in terms of generative and exploratory actions. It was found that the activities of EM and PD could be reconstructed in terms of actions of both kinds. This suggests that EM and PD are creative activities. Although the activities of SD could also be reconstructed in terms of generative actions they were far more constrained and there were no exploratory actions found in the activities of SD. This suggests that there is a restricted form of creativity when generating artefacts in SD making creative exploration of the resulting artefacts ineffective and unnecessary.

Chapters 3, 4 and 5 investigated how EM relates to an essentially creative discipline such as PD. This was facilitated by using Pugh's notion of total design [Pug91]. The results of the chapters indicate that EM is a suitable framework for creativity: it was found that EM corresponds to the conceptual design stage of conceptually dynamic design, an essentially creative activity, and that the artefacts and actions of EM share the same creative properties as those of PD. The comparison with PD brought attention to the danger of general concepts and principles of EM acquiring an analytical bias by emphasizing SD only. The analytical and creative significance of the concepts and principles of EM are made clear in Chapters 3, 4 and 5 by relating them to PD and SD.

Chapters 4 and 5 make use of an experimental approach to investigating creativity, in relation to the EM framework, that gives scientific integrity to the findings. Creative cognition [FWS92] characterizes the properties of structures and processes essential to creativity. These properties and processes were investigated by the psychologists Finke, Ward and Smith through experimentation. Chapters 4 and 5 give the results of experiments to find the properties and actions, characterized in creative cognition, within the artefacts and activities of EM, PD and SD.

Chapter 6 presents a paradigm for creative SD and assesses the likelihood of it being adopted by those currently involved in SD. The proposal for the new paradigm uses EM as its framework and has the potential of changing the way SD is thought about and performed. It is centred on the generalized notion of a *computer as a system* that is configured by the software developer. In this new paradigm *programming is configuring the system* and the *program is the system configuration.*

7.2 Further work

7.2.1 Software engineering

There are many conflicting views about software engineering. There is confusion over whether software engineering even exists as a discipline: it is clear from companies advertising for software engineers and universities teaching courses in software engineering that many in industry and academia believe that software engineering already exists but there are others who argue that an engineering discipline has yet to emerge from SD [Gib94]. There is also a lack of consensus over what is meant by software engineering [Ber92] with research papers giving alternative and sometimes conflicting definitions of the term.

Arguably the main cause of these conflicting views about software engineering is the problem of understanding engineering in terms of SD. The word engineering has always been associated with engineering in PD and established disciplines such as civil, industrial, electronic and mechanical engineering which are to do with the analysis of objects and systems. This makes it difficult to associate the word engineering with SD which is essentially to do with abstract descriptions of structure and function. This lack of understanding has lead to the term software engineering meaning different things to different people.

The creative approach to SD proposed in this thesis promises to provide a way of understanding SD from an engineering perspective. It achieves this in two ways: first, by relating SD and EM to total design thus providing a framework for understanding conventional engineering disciplines and their role in PD. Second, by broadening the notion of SD to include the development of objects and systems that are the subject of conventional engineering disciplines.

This suggests that software engineering is in the same relation to creative SD as engineering design is to total design in the sense of Pugh [Pug91]:

- software engineering is preceded by an essentially creative activity, such as EM, whereby ideas for objects and systems are generated, represented and evaluated;
- software engineering incorporates an activity whereby representations of ideas for objects and systems are analyzed to turn them into detailed component specifications;
- software engineering is an activity whereby existing objects and systems are analyzed to discover their constituent components and the interfacing between components in order to formulate detailed component specifications;
- software engineering is followed by an activity whereby components are produced and combined into finished products to be sold.

Within the context of SD as systems development component specifications are not restricted to abstract definitions of computer processes and data structures. The components specified could include integrated circuits, resistors, shafts, bearings, concrete beams and door frames.

Software engineering, within the context of creative software development, would have to deal with the analysis of programs, objects, systems and their representations. Conventional SD already provides a way of analyzing a statement of requirements for a desired system and transforming it into code. Computer science provides powerful techniques and tools for analyzing code and formal representations of structure, behaviour and functionality [OG75, Bar85, Pnu86, Hoo91, AO91, MP92a, Man74, Heh84, San88, Bac87, C90]. Traditional engineering disciplines provide powerful techniques and tools for analyzing objects and systems. But currently there seems to be no established framework for uniting the techniques of SD, computer science and traditional engineering disciplines. Further work could include investigating EM as a possible candidate for such a framework.

A key problem is finding an appropriate framework for analysis within the context of creative SD. Two approaches to analysis emerged during the research for this thesis both based on the notion of observation in EM: "A theory of observation is a plausible basis for a principled method of constructing a program model for a reactive system" [BR92]. These approaches involve

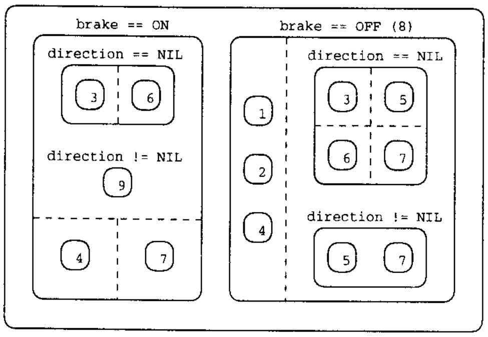
- representing the organization of observations of a phenomena described by a script as a single graph showing observations that cannot happen concurrently, observations that can happen at the same time and contexts for observation (see Example 7.1), and

- representing the sequence of observations of a phenomena described by a script as a single graph showing the order in which observations are be made (see Example 7.2).

Example 7.1. Organization of observations in EM. The guards of the nine MUL ADM actions

```
1 landButton{_F} == UP && _F == floor + 1 && brake == OFF -> brake = ON,
2 landButton{_F} == DOWN && _F == floor - 1 && brake == OFF -> brake = ON,
3 landButton{_F} != OFF && direction == NIL -> destination = _F,
4 floor == _F -> landButton{_F} = OFF
5 carButton{_G} == ON && _G == floor + direction && brake == OFF
                                                         -> brake = ON,
6 carButton{_G} == ON && direction == NIL -> destination = _G,
7 floor == _G -> carButton{_G} = OFF
8 brake == OFF -> floor = floor + direction,
9 brake == ON && direction != NIL -> brake = OFF
```

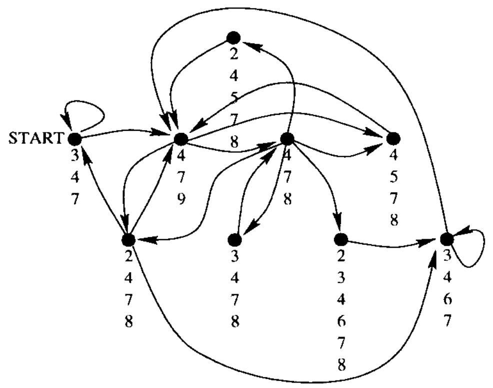
corresponding to observations are represented by the graph



showing observations that cannot happen concurrently in separate boxes, observations that can happen at the same time as boxes separated by a dashed line, and contexts for observation as boxes within boxes.

Other frameworks for analysis in EM are being investigated by Gehring [GYC+96]. Further work could include investigating the relation between formal methods and these approaches to analysis. The organization of observations represented in Figure 7.1 is similar to Harel's statecharts [Har88, Har87, HLN+88, Har92]. Statecharts and EM artefacts are contrasted by Cartwright and Beynon in [BC95]. Representing behaviour as sequences of states, as in Example 7.2, is standard in computer science especially in the subfield of temporal logic [Har88, Pnu86, MP92a, Hal87].

Example 7.2. Sequencing of observations in EM. Some of the sequences of observations described by the the actions of the MUL ADM given above are represented by the graph



showing the order in which observations are made during the interaction by two user entities over a period of ten clock cycles on five different occasions.

7.2.2 EM in context

As well as providing the framework for the proposed creative approach to SD EM also has a practical role to play. Within the context of creative SD it would be expected that EM would be used to generate, represent and evaluate alternative ideas for innovative system solutions. However, there is no preconceived strategy given by EM because it is inappropriate to put abstract constraints on situated activities. Further work could involve the investigation of contexts for EM that provide practical support for the generation, representation and evaluation of ideas for system solutions.

Finke *et al* [FWS92] made one of their goals to develop practical techniques for applying the principles of creative cognition in everyday situations. Pugh [Pug91] also makes suggestions for practical techniques to be used in conceptual design:

- brainstorming which is when a group attempts to find a solution to a problem by amassing all the ideas spontaneously contributed by its members;
- exploring new problems and functions suggested by a structure (function-follows-form) instead of the more conventional approach of generating struc-

tures with specific problems or functions in mind (form-follows-function);

- interpreting concepts which are unusual combinations of existing concepts as solutions to problems.

Further work could include finding ways of using such techniques in EM which would use LSD specifications, scripts, visualizations and animations with the aim of helping the modeller.

Pugh [Pug91] talks about creativity in the context of controlled convergence in relation to conceptual design. He argues that creativity should always be carried out in conceptual design within the context of a product design specification (PDS). He suggests an approach based on generation followed by evaluation in which solutions are generated by a designer with the PDS in mind and then these solutions are evaluated by a group using a decision matrix and criteria based on the PDS. Those solutions which pass evaluation continue to be refined and evaluated until the process converges on the best solution. Further work could include investigating a suitable method for controlled convergence for EM perhaps using the PDS and decision matrix as a starting point for research.

7.2.3 Distribution of EM tools

The concepts of EM are best communicated by demonstration using the EM tools. This is because their meanings depend on the experiences provided by models generated by EM tools running on computers. So, it is important to have the tools to fully appreciate EM. This presents a dilemma: people need EM tools to fully appreciate the modelling approach and yet they are unlikely to bother acquiring the tools unless they already have a an appreciation of EM. The way to break out of this cycle is to make EM tools freely available on the internet and to distribute them to people who might find them useful. This work is important if disciplines that depend on EM, such as creative SD, are to have any hope of adoption in the future. Extensive work in this area has already been done by Yung who most recently developed Tkeden [BSY95]. Currently work in this direction is being done by Cartwright who has ported tools to the PC platform and improved the tools to make them more appealing and easier to use [BC97]. Most recently a workshop has been organized for teachers with a view to introducing EM into the classroom.