Chapter 7 – Summary and Conclusions

7.0 Overview of the Chapter

In this chapter, we review the thesis, discuss possible future work, discuss limitations of the research and draw conclusions.

7.1 Review of the thesis

In this section, we review the contents of the thesis and evaluate them with reference to EM and learning.

In Chapter 2, we considered the qualities of software based on spreadsheet principles where learning is concerned. We identified a connection between the support spreadsheets offer to learning and their characteristics as tools for exploratory modelling. We examined how exploratory modelling is related to Cantwell Smith's semantic relation β and argued that it has two key aspects: negotiation and elaboration. Negotiation of the semantic relation β is essentially concerned with understanding the nature of a concept. Elaboration of the semantic relation β is essentially concerned with understanding how a concept can be applied in its wider domain context. We evaluated computer-based modelling tools in respect of supporting the two key aspects of the semantic relation β. We argued that spreadsheets are well suited to negotiation in certain domains but are limited in respect of elaboration. We also considered research products related to the spreadsheet and found that these also had limitations where supporting the semantic relation β is concerned. We introduced the practical EM tool, TkEden, and argued that it offers general support for both negotiation and elaboration of the semantic relation β, and hence is a good tool for exploratory modelling.

In Chapter 3, we outlined the challenges of exploiting ‘computers for learning’. These difficult challenges can be attributed in part to the differing concerns of the educationalist and the computer specialist. In learning, typical computer use can be separated into two types of activity: building models and using models. We argued that satisfying the requirements of the educationalist and the computer specialist requires that:

- in model building there is a close connection between the model construction approach and domain learning.
- in model use there is easy and flexible adaptability of software in response to different learning situations.

We introduced an experiential framework for learning that can be used to describe different domain learning activities on a spectrum that ranges from the concrete, empirical and private to the abstract, theoretical and public. We argued that this framework can be viewed as being generally applicable to any learning situation. We introduced the key features of EM: the development of construals; the emphasis on state-as-experienced being prior to behaviour-as-abstracted; and the concepts of observables, dependency and agency. We argued that model construction in EM can support learning activities, and migration between learning activities, across the whole range of the EFL. This close connection between EM and the EFL suggests that domain learning and model construction can be intimately linked.

In Chapter 4, we examined the relationship between model construction and domain learning from an educational perspective. We took a broad constructionist perspective on learning embracing bricolage and situated learning. We argued that all computer-based model construction approaches involve active knowledge construction through building a public entity, and hence satisfy Papert’s basic definition of constructionism. By relating instructionist and constructionist theories to the EFL, we established that instructionism is typically concerned with abstract learning activities and constructionism is typically concerned with concrete learning activities. We examined how three domain learning techniques: concept mapping, conventional programming and EM, are related to our broad constructionist perspective on learning and the EFL. We concluded that:

- concept mapping is primarily useful in brainstorming activities and that knowledge gained using it is typically set aside when constructing models.
- conventional programming is not well oriented to the broad perspective on constructionism adopted in this thesis because it emphasises planning, abstraction and circumscription. These emphases align programming with abstract learning in the EFL, and this detracts from its usefulness as an approach to model construction that promotes domain learning.
- an EM approach to model construction supports our broad perspective on constructionism and learning activities across the EFL, enabling effective domain learning to proceed in tandem with model construction.

In Chapter 5, we considered the advantages to learners, teachers and software developers in using EM to construct learning environments that support many different types of learning objective. We described three different types of learning that can be scaffolded: comprehending a fixed referent; exploring possibilities and invention; and learning domain-specific languages. In each of these types of learning environment, an initial seed model was embellished by extending or refining the existing model. The case studies used to illustrate scaffolding of EM learning environments exhibited advantages for:

- learners, since the learning environments lend themselves to exploration of the model domain and referent.
- teachers, since environments are – in principle – customisable resources that can be utilised to take advantage of their particular teaching requirements.
- software developers, since the approach to model development means that families of models can be easily created in response to teacher demands and flexibly adapted or extended to suit different contexts or learner competencies.

In Chapter 6, we described EM case studies illustrating model building and model use in relation to the EFL. We recalled the two categories in the use of computers for learning identified in chapter 3: the construction of computer-based models by learners; and the use of existing models where a user does not make any changes to the model. We argued that a third category was particularly appropriate in EM: where a partially complete model can be used as it stands, or extended to fulfil some learning criterion. The three case studies described in the chapter: the Free Distributive Lattice model; the Heapsort model; and the Robotic Simulation Environment give practical illustration to our claim that the use of EM in learning can support a wide range of learning activities within the EFL.

7.2 Future work

7.2.1 Empirical Testing

In this thesis, the focus has been on establishing a solid conceptual foundation for future EM-based educational research. It is the author's opinion that future work needs to be targeted at practical developments for use in educational situations.

The quality of computational resources in an educational environment is assessed with reference to the learning objectives to which they are being put. Many hundred models have been developed in practical EM case studies as a result of student project work and academic research. The majority of these models have not been developed with education in mind. There is a definite need to target schools more effectively with regard to the development of EM models as learning resources. In my view, this targeting should involve a close relationship with a particular school and model development based on their educational requirements. This arrangement would have a two-way benefit. The school – who are not so interested in the underlying computer implementation – will receive specifically relevant educational software. The research group would have the opportunity for valuable empirical testing of EM models in practical educational situations.

7.2.2 Comparative studies

To date, there has been no evaluation of EM in education in comparison with other programming languages or software environments. There is a need for comparative studies to ascertain whether there are quantitative and/or qualitative improvements in learning that occur in practice through the adoption of an EM approach. An exemplar for this type of comparative study can be seen in the Playground Project [Pla03], a 3-year European research project that compared the Toontalk programming environment [Too03] and a Logo-based language called Imagine [KB00]. Imagine was used to create a simple game-programming environment called Pathways [GKN+01]. Children built their own games on top of these environments, and these games were used as the vehicle for discussing how they coped with ideas such as building rules, cause and effects and issues of object-orientation (cf. section 5.3.3). The primary concern in the Playground project was to compare and contrast the quality of the children's evolving understanding of both their game domain and their programs in the two environments. Comparative studies of EM with other educational programming languages and software is essential to identify whether, in practice, the suitability of EM for learning argued in this thesis can be realised. Further, in this thesis we have argued that there are advantages in respect of domain learning when using EM models rather than conventional programming. We have explored this claim insofar as it relates to theoretical and conceptual issues. Practical evidence in support of such a claim can only arise from realistically scaled comparative testing, and not through the anecdotal accounts described in this thesis (and referenced from [Her02, Won03, Wan03]).

7.2.3 Developing an Empirical Modelling environment for children

The current EM tools are suitable for programmers and not for end-users, whether adults or children. Our own experiments in trying to introduce our tools to 17-18 year old students showed that non-computing students could not grasp the fundamental ideas underlying programming languages, such as functions or parameter passing. We had evidence that students grasped the conceptual ideas of analysing a domain such as the jugs game (discussed in section 2.4.4) in terms of observables, dependency and agency, but they could not translate these into the necessary computer definitions. The creation of a simpler development environment or visual language, developed in accordance with EM principles, could potentially enable students to construct their own models.

There have been many attempts at giving what Papert termed ‘the power of programming’ to children [Pap80]. Most share his vision for the Logo language: of providing the children with a tool with which they can construct their own models or programs according to their desires. Programming languages aimed at children typically take inspiration from a particular programming paradigm, and translate the important features of each paradigm into an appropriate and simple form. For example, Logo was a language aimed at children that was built on the procedural programming paradigm [Pap80]. Agentsheets (as discussed in section 2.3.3) uses a visual programming language to specify rules to give agents in a simulation [Rep93]. Toontalk associates computational notions such as procedures and message passing with actions such as training robots and sending birds back to their nests [Too03].
Without significant empirical testing, it is difficult to imagine how the principles of EM can be expressed using metaphors with which children will be comfortable. In order for an EM language aimed at children to be successful, it is essential to remove the syntactic barriers presented by our current notations. Previous research by Wong [Won98, Won03] has been targeted at developing end-user EM languages. However, the DMT tool (see chapter 4 and [Won03]) and the WING environment [Won98] – although they are both visual – are still too complicated for children to use in developing their own models. Both still require users to interact with complex syntax.

At present, the construction of models could proceed with children and expert modellers collaboratively building models. The child would provide the ideas and the expert would translate them into definitions and program code. It remains for future work to ascertain whether this is an appropriate mode of working from a pedagogical perspective. Careful planning and evaluation would be required to establish the educational benefits to the child in such an arrangement.

7.3 Conclusions

In this final section of the thesis, we identify limitations of the research described in this thesis and put forward the final conclusions that have been reached.

7.3.1 Reservations about the research

Although the EM research group has been in existence since the early 1980's, there has been relatively little empirical testing with software users, and very little use of the tools for model construction outside the University of Warwick. Although many models developed by the EM group have been deployed in 'one-off' trials with students, or demonstrated to teachers, only two have been more extensively tested in educational contexts:

- The current SQL-EDDI environment (see section 5.4.3) has been used over a three-year period by approximately eight hundred computer science students as part of an undergraduate database course. An earlier version of the EDDI interpreter was used in the previous two years by approximately five hundred computer science students. With this level of exposure, and the associated feedback from students, we have evidence for the educational merits of the environment (see [BBR+03] for more details).
- The Clayton Tunnel accident scenario (see section 2.4.5) has been used on several occasions as the basis for University of Warwick open days for 14-year-old children. Each session has lasted for an hour and involved children playing the roles of the participants in the situation and taking

part in discussions on railway safety issues. The educational benefit of such sessions is hard to evaluate, but the quality of the post-mortem discussions indicated that the close engagement with the model stimulated a positive learning experience.

The models described in chapters 5 and 6 that are targeted at educational settings (see e.g. racing cars (section 5.2.1), the OXO models (section 5.3.2), the clown-and-maze environment (section 5.4.2), the heapsort model (section 6.3) and the RSE (section 6.4)) have not been tested in educational contexts. To fully evaluate the claims made in this thesis, these models should be tested in the environment in which they are intended to be deployed. Only after such an evaluation can we be sure of the validity of our conclusions.

Writing by itself is not the ideal way to disseminate EM. In this thesis, we have argued that model construction in EM is different in character from developing a conventional program. It is difficult for the reader who is unfamiliar with EM to form a considered view on this claim without any practical exposure to developing models. In our opinion, such exposure is vital to fully appreciate the distinctive character of EM in relation to conventional programming. Whilst discussions in previous papers (see e.g. the discussion of the clock model in Appendix D and [Bey01] and the discussion of the lift model in [Bey03]) can give an impression of the open-ended and flexible nature of model construction, it is only through personal and practical engagement (as Papert himself would advocate!) that the reader can fully appreciate the nature of EM model development.

7.3.2 Conclusions of the thesis

In this thesis, we have explored the potential of EM in the application of computers for learning. This exploration has addressed its merits in respect of the construction of models, and the use of models.

In respect of model building, we have argued that in EM, the process of model construction is intimately linked with furthering domain understanding. This intimate link exists because of the support EM model construction offers for a wide range of learning activities, as reflected in the EFL. In particular, EM supports the concrete learning activities in the EFL that are more closely associated with constructing spreadsheets than writing programs. Furthermore, the model construction approach in EM allows the modeller to move at will between learning activities across the EFL in response to their learning needs at the time.

In respect of model use, we have argued that EM learning environments are flexibly adaptable in response to different learning contexts. This flexibility stems in part from the openness of the model development approach. The use of cognitive layering – the layered development of microworlds in which future layers are not preconceived, and can be flexibly adapted - has advantages for all the stakeholders in educational software development. Our case studies have illustrated that EM is able to support the wide range of different learning activities represented in the EFL, and confirm its potential as an approach for addressing the agenda of 'computers for learning'.